

# KD11-K

## KD11-K MICRODIAGNOSTIC MD-11-DQKUB-B

EP-DQKUB-B-DL-A

AUG 1977

COPYRIGHT © 1977

**digital**

FICHE 1 OF 3

MADE IN USA

The microfiche card displays a grid of 144 frames, arranged in 12 rows and 12 columns. Each frame contains a small, high-contrast image, likely a diagnostic test result or a component layout for the KD11-K microdiagnostic system. The images are too small to read clearly but appear to be diagnostic test results or component layouts.

# KD11-K

KD11-K MICRODIAGNOSTIC  
MD-11-DQKUB-B

EP-DQKUB-B-DL-A  
COPYRIGHT © 1977

AUG 1977  
**digital**  
MADE IN USA

FICHE 2 OF 3

This microfiche card contains a grid of 14 columns and 24 rows of tiny, illegible data frames. Each frame appears to be a small table or data record, but the text within them is too small to be read. The frames are arranged in a regular grid pattern across the entire surface of the card.

# KD11-K

KD11-K MICRODIAGNOSTIC  
MD-11-DQKUB-B

EP-DQKUB-B-DL-A  
COPYRIGHT © 1977  
FICHE 3 OF 3

AUG 1977  
**digital**  
MADE IN USA

The microfiche card displays a grid of 120 frames, arranged in 10 rows and 12 columns. Each frame contains a small, high-contrast image of a document page. The pages appear to be technical or diagnostic in nature, with some containing text and others containing diagrams or tables. The frames are arranged in a regular grid pattern, typical of a microfiche card. The overall appearance is that of a standard microfiche card used for data storage and retrieval.

11-11

B01

EOF1DXQLKBSEQ  
POP10 411

00010000

770720

POP10 411

HOR1DQKUBBSEQ

00010000

770720

I D E N T I F I C A T I O N

PRODUCT CODE:           MAINDEC-11-DQKUB-B-D  
                          VERSION /101/  
PRODUCT NAME:            KD11-K Microdiagnostic  
MAINTAINER:              Diagnostic Engineering  
AUTHOR:                  Don North  
DATE CREATED:            18-Jan-1977  
LAST REVISION:          22-Jun-1977

COPYRIGHT (C) 1977; Digital Equipment Corporation  
146 Main Street  
Maynard, Massachusetts, USA  
01754 617-897-5111

This software is furnished to the purchaser under a license for use on a single computer system, and can be copied (with inclusion of DIGITAL's copyright notice) only for use in such system, except as may otherwise be provided in writing by DIGITAL.

The information in this document is subject to change without notice, and should not be construed as a commitment by DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION assumes no responsibility for any errors that may appear in this document.

DIGITAL assumes no responsibility for the use or reliability of its software on equipment not supplied by DIGITAL.

- TABLE OF CONTENTS -

1.0	ABSTRACT
2.0	REQUIREMENTS
2.1	Hardware
2.2	Documentation / Listings
3.0	LOADING PROCEDURE
4.0	STARTING PROCEDURE
4.1	Operator's Console
4.2	DCS Switches
4.3	Console "INIT"
5.0	OPERATING PROCEDURE
5.1	DCS Indications While Executing
5.2	DCS Execution Time
5.3	DCS 'End of Pass' and 'Error' Indications
6.0	RESTRICTIONS
6.1	Hardware
6.2	Software
7.0	TEST DESCRIPTION
7.1	Test Structure
7.2	DCS Microcode Conventions
7.3	What is/is-not tested
7.4	Global Test Order
8.0	ERROR HANDLING
8.1	What is an 'ERROR' ?
8.2	'FAULT DIRECTORY' Format and Use
8.3	'SCOPE Loop' Facility
9.0	REVISION HISTORY
9.1	Revision Number
9.2	Revisions to DCS Code
10.0	DCS VERIFICATION / SELF TEST
10.1	Requirements
10.2	Method / Algorithm
10.3	Procedure and Indications
11.0	MISCELLANEOUS
11.1	ACT/APT/XXDP
11.2	Macro Instruction Interface

## 1.0 ABSTRACT

The 'DCS' (Diagnostic Control Store) Module is a diagnostic tool specifically designed for the PDP-11/60 (KD11-K) Central Processor. Functioning as an alternate 2048 word control store, microcoded tests are executed to detect and isolate errors within the internals of the processor control and datapath hardware. Error indication information is provided by the DCS module; coupled with an indexed FAULT DIRECTORY, errors are resolved to the module level, when possible. Additional information is also provided resolving the error to a specific functional logic block. Significant benefits of this micro diagnostic approach are seen to be:

- Memory/I-O Device/UNIBUS independence
- Direct hardware microcontrol and visibility
- Extremely fast load and execution times
- Excellent coverage and resolution

## 2.0 REQUIREMENTS

A preliminary note:

Throughout this DCS User's Guide, the two terms "FAULT" and "ERROR" have been used interchangeably. They both are to indicate "malfunctioning logic elements" (eg, busted IC's) and open/shorted ETCH runs, etc., in the unit under test. No distinction between the two terms is implied.

### 2.1 Hardware

To use the DCS, the following hardware is required:

1. DCS (Diagnostic Control Store) Module M7871 (KU116-BB)
2. PDP-11/60 (KD11-K) Central Processor
3. DL11-W Line Clock (required)
4. First 4k memory bank (required minimum)

## 2.2 Documentation / Listings

The available documentation for the DCS module user comprises the following items:

- DCS User's Guide (this document)
- ~~FAULT~~ DIRECTORY Listing; for module replacement information
- DCS Microcode Listing; when IC level debug is necessary
- DCS Maintenance Manual, Print Set; for detailed information on DCS hardware operation
- KD11-K Processor Maintenance Manual, Print Set; for IC level debug, base machine hardware specific information

Specific MAINDEC component part numbers for the DCS documentation are as follows:

- MD-11-DOKUB-\*-D;  
User's Guide, FAULT DIRECTORY (PAPER, 80 pages)
- MD-11-DOKUB-\*-LA;  
DCS Microcode Listing (PAPER, 450 pages)
- MD-11-DOKUB-\*-FA;  
User's Guide, FAULT DIRECTORY, DCS Microcode Listing (FICHE, 4 cards)

## 3.0 LOADING PROCEDURE

The DCS occupies slot 1 in the KD11-K processor backplane; thus if an ECS or UCS option is present, it must be removed. To load the DCS, the following sequence should be employed:

1. Power down the CPU
2. Remove the ECS/UCS option from processor slot #1, if present
3. Insert the DCS module into slot #1. Use caution while inserting the DCS module, as a slightly bowed board may require some gentle maneuvering to seat it in place.
4. Orient the DCS 'RUN/STOP' switch to 'STOP', and the 'NORMAL/VERIFY' switch to 'NORMAL'
5. Now power up the CPU

#### 4.0 STARTING PROCEDURE

DCS execution can be initiated by two distinct methods:

##### 4.1 Startup via the Operator's Console (KY11-P)

With the KD11-K CPU in "CONSOLE" ("HALTcd") mode, simultaneously depress the KY11-P operator's console "CNTRL/DIAG" keys to start the DCS. (If no DCS/ECS/UCS happens to be present, there should be no effect). Section 5.1 (below) interprets the display on the operator's console while the DCS is executing, and after it stops. If this method fails to start the DCS, proceed to the next paragraph.

##### 4.2 Startup via the DCS Switches

If it is desired for some reason to bypass use of the KY11-P operator's console to initiate the DCS, an alternative means is provided. This method would be used, for example, in the event that starting from the operator's console was not possible (using "CNTRL/DIAG") due to some hardware malfunction. The procedure is:

1. On the DCS module, set the 'NORMAL/VERIFY' switch to 'NORMAL'
2. Now set the 'RUN/STOP' switch (on DCS module also) to the 'RUN' position, or flip it from 'RUN'-'>'STOP'-'>'RUN'

The DCS should now assume control of the KD11-K CPU, irregardless of the previous state of the CPU ("CONSOLE", "RUN", or whatever).

#### NOTE

If the DCS 'RUN/STOP' switch was already in the 'RUN' position, then it is expected that the operator's console keys have no effect - the DCS is already enabled to execute, and is controlling the CPU. The DCS microcode does not monitor the console keypad for operator input.

If neither of the above methods produce a 'RUNNING' indication of the DCS (as per Section 5.1), then proceed to the next paragraph.



### 4.3 Using Console "INIT"

Set the DCS switches as detailed in Section 4.2 (ie, 'NORMAL' and 'RUN'). Now generate a processor "INIT" signal by simultaneously depressing the "HALT" and "START" keys on the operator's console. This should now initialize the CPU logic and restart the DCS, producing a 'RUNNING' indication (see Section 5.1).

If no 'RUNNING' indication is now present, then a problem exists in the power supply, system clocks, DCS module, or ???.

## 5.0 OPERATING PROCEDURE

### 5.1 DCS Indications While Executing

#### 5.1.1 On the KD11-K Operator's Console

While the DCS is executing, the operator's console display should be approximately as follows:

"RUN"	-	ON Continuously
"PROC"	-	BLINKing
"USER"	-	BLINKing
"CONSOLE"	-	OFF Continuously
"BATTERY"	-	<indeterminate>

The 6 digit octal display should read:

( 0 0 0 0 0 0 ) - In DCS pass 1;

(.0.0.0.0.0.0.) - In DCS passes 2-64.

If any of the above conditions are not met, then either:

1. The console hardware is inoperable, or
2. The DCS is hung in an 'error/scope loop'

[Not necessarily in that order]

5.1.2 On the DCS Module

On the DCS module, the two important indicators to watch at this time are the 'ERROR' and 'EOP' LEDs:

-----LEDS-----		State	Comment
'ERROR'	'EOP'		
OFF	OFF	Probable ERROR	DCS and/or processor HUNG
OFF	ON	* EOP *	Successful EOP
ON	OFF	* ERROR *	Genuine ERROR
ON	ON	Probable ERROR	DCS 'VERIFY MODE' indication

Note the 'Probable ---' notation. Either 'EOP' or 'ERROR' on the DCS should be lit: both OFF or both ON indicate non-standard conditions that require further investigation. Neither 'ERROR' nor 'EOP' indicates that the DCS and/or processor are in a HUNG condition (eg, clocks suppressed); see section 8 (Error Handling). Both 'ERROR' and 'EOP' is the standard DCS 'VERIFY MODE' indication. Make sure the DCS 'NORMAL/VERIFY' switch is set to 'NORMAL' (if this is intended), and then restart the DCS. See section 10 (Verify Mode) for further information.

5.2 DCS Execution Time

The execution time of the DCS will vary depending upon its mode of initiation.

5.2.1 "CNTRL/DIAG" Start From Operator's Console

When started via "CNTRL/DIAG" from the operator's console, and assuming no errors, DCS will execute with console display as detailed above for approximately 6 seconds (64 passes at 100 milliseconds/pass, about 350,000 microcycles/pass). After this time has elapsed, control should return to the KD11-K microcode, as if the CPU had just been powered up with the slide switch set to "HALT". Note that the machine will end up in console mode irregardless of the actual slide switch settings (RUN/BOOT/HALT). Section 5.3 interprets the console and DCS displays after this delay time, for successful 'END-OF-PASS' and 'ERROR' indications.

5.2.2 Start From DCS Switches

When started via setting 'RUN/STOP' to 'RUN' on the DCS module, the DCS will execute continuously, not returning to the KD11-K microcode until the 'RUN/STOP' switch is reset to the 'STOP' position. Throughout this time, the execution indications will be as previously detailed. When the switch is returned to 'STOP', the DCS may execute for a maximum of

approximately 6 more seconds; and then proceed as described directly above, entering "CONSOLE" mode.

### 5.3 DCS 'END-OF-PASS' and 'ERROR' Indications

#### 5.3.1 On The DCS Module

Successful 'END-OF-PASS' and 'ERROR' conditions detected by the DCS are indicated most directly on the DCS module by the two LEDs labeled 'EOP' and 'ERROR'. 'EOP' is turned on at the end of each successful pass thru the DCS code - assuming 'ERROR' has not yet been turned on. In an error-free running situation, 'EOP' will be on; 'ERROR' off. Note that 'EOP' comes on at the end of the first pass (assuming no errors), and thus will appear to be on continuously.

In the event the DCS detects an error, the 'ERROR' LED will be turned on, and the 'EOP' LED turned off. The 'ERROR' LED will be latched in such a way that it cannot be turned off if the error disappears (i.e., a 'flaky' timing error), thus retaining the indication of an error. See section 8 (Error Handling) for a full treatment of the various error conditions: Detection, Indication, Scope-looping.

#### 5.3.2 On the Operator's Console

Successful 'END-OF-PASS' and 'ERROR' conditions are indicated on the operator's console as follows:

1. A successful 64. passes and return to KD11-K processor microcode control, by a

(.1.2.3.3.2.1.)

in the console display, and the KD11-K processor "HALTed" in "CONSOLE" mode.

2. An error in pass 1 will usually (but not always) be indicated by a

( 0 0 0 0 0 0 )

in the console display.

3. An error in pass 2 thru -- will usually (but not always) be indicated by a

(.0.0.0.0.0.0.)

in the console display.

Note the qualifications in the previous statements:

For a successful 'END-OF-PASS' and exit back to KD11-K microcode control, only one display is valid:

(.1.2.3.3.2.1.)

For any other display except "(.1.2.3.3.2.1.)", check the DCS module 'EOP' and 'ERROR' LEDs for the most reliable indication of the result of execution.

See Section 8 for a full explanation of error processing.

## 6.0 RESTRICTIONS

### 6.1 Hardware

#### 6.1.1 Cache and Memory Management (KT)

The DCS executes with both Cache and KT disabled:

1. Cache, by setting both the "Force Miss" bits
2. KT, by clearing the "Enable" bit

The DCS checks the most basic path from the UNIBUS to/from internal data paths. Further macro diagnostic programs are available for Cache (MD-11-DOKKA-\*) and Memory Management (MD-11-DOKTA-\*) fault diagnosis.

#### 6.1.2 MOS Memory Battery Backup

The MOS memory battery backup (if present) must either be "Good" or else disabled. Otherwise, the micro diagnostic code WILL NOT execute without detecting an error.

Indications are: ERCD=5621/TNUA=7400, in TEST620C.

### 6.2 Software

#### 6.2.1 Return To Console / DCS End-of-pass Processing

Console "HALT" mode is the ONLY exit provided from DCS back to base machine microcontrol. This action is due to:

- 1) The DCS, upon detecting an error, locks itself (and the processor) up in such a way that manual intervention by the operator is required for return to base machine microcontrol.

- 2) The DCS completely alters the internal microstate of the processor, destroying its previous contents, and leaving "garbage" in its place. Thus the full base machine power-up "INIT" sequence is generated by DCS to "clean-up" the processor prior to returning control.

## 7.0 TEST DESCRIPTION

### 7.1 Test Structure

#### 7.1.1 Philosophy

The testing philosophy used in the design of the DCS microcode centers around two major points:

1. Start with as minimal a **HARDCORE** as possible.
2. Use only **TESTED** and **VERIFIED** logic elements to diagnose **UNTESTED** elements; add **TESTED** elements to the arsenal of logic available for further testing.

This method of test organization and construction presents the best approach for building a high resolution and high coverage diagnostic program. Section 7.4 (below) summarizes the actual testing order present in the DCS microcode, as designed using the above philosophy. Section 8.1 presents more specific details on the actual method of **ERROR** detection employed by the DCS microcode and hardware.

#### 7.1.2 Mechanics

The manner in which DCS tests are setup is depicted in the following diagram:

```
  \ /
  (setup ENUA/DCS-CNTR as test requires)
  .
  |
  (perform the actual test function)
  -may be inline microcode
  -may call subroutine
  .
  |
  (jump to BUT area, perform BUT(test) into TARGET)
  .
  |
  (enter TARGET area from BUT, ENUA::TNUA compare)
  -the DCS-CNTR was loaded above to enable compare here
  -the ENUA was setup at correct TARGET point
  -exit via BUTA(RETURN)
  .
  \ /
  (...next test now begins...)
```

7.2 DCS Microcode Conventions

7.2.1 Microdiagnostic Code Listing Organization

The format of the DCS microcode listing is presented in the following table, along with a brief description of each section (when necessary):

pages	contents
1-5	IDENTIFICATION, TABLE-OF-CONTENTS
6-9	DCS REGISTER LAYOUTS, MICROWORD BIT TABLE a number of graphic tables
10-24	MICROWORD FIELD DEFINITIONS, BASE MACHINE defines U<47:00> and their functions
25-26	DCS FIELD DEFINITIONS defines U<54:48> and their functions
27-37	"SIMPLE" MACROS these macros are combinations of FIELD definitions only
38-52	"ADVANCED" MACROS these macros are combinations of SIMPLE macros
53-380	---DCS-MICRODIAGNOSTIC-TESTS---
381-384	EOP/VERIFY-MODE MICROCODE microcode used in END-OF-PASS and VERIFY-MODE processing
385-393	GENERAL SUBROUTINES, COMMON CODE this section contains some VERY commonly used subroutines
394-395	JAMUPP MICROCODE all JAMUPP conditions enter here ...
396-402	BUT(---) TAKEOFF MICROWORDS most all DCS tests start their microbranch from this list of BUT's
403-417	BUT(---) TARGET MICROWORDS and most all DCS tests end up here, where they compare their ENUA::TNUA in this 256. word table
418	BIT MAP OF DCS ADDRESS SPACE

- a BIT MAP of the entire DCS address space, 1=USED, 0=FREE
- X1-X10 SYMBOL/LINE-NUMBER/LOCATION CROSS REFERENCE  
a very useful reference to find the location in the DCS listing of a particular symbolic label
- X11-X18 LOCATION/SYMBOL CROSS REFERENCE  
an expansion of the above BIT MAP, substituting the symbolic label for the I/O's present in the above
- X19 FREE/USED LOCATION SUMMARY  
a quantitative summary of FREE/USED microlocations, by PAGE and TOTAL

### 7.3 What is/is-not tested

The DCS micro diagnostic code has been designed to detect and isolate errors within the "internals" of the KD11-K processor hardware. As such, it does not attempt (nor is it possible) to detect certain errors, which are processor related, but require devices external to the processor. Errors related to NPR/BR arbitration sequences, multiple BR priority level interrupt sequences, POWER FAIL / RESTART, etc. are a few of the elements of this general class. The following sections enumerate both classes of logic: TESTED by DCS, and UNTESTABLE by DCS.

#### 7.3.1 TESTED by DCS

Essentially, the DCS is designed to test the "heart" of the KD11-K processor - those elements that must be functional to "bring up" the processor to a level whereupon further macro diagnostic programs could then be loaded and started (ie, M9301-YH boot/diagnostic, processor and peripheral diagnostics, etc.) to successfully isolate more complex processor and system related errors. Processor operation with either KT/KJ or CACHE enabled was not considered as part of this "heart" - as the processor will run perfectly fine without either of these facilities (albeit at a degraded performance level). Thus the DCS is able to concentrate on more thorough coverage and resolution in those portions of the hardware that are least "visible and testable" from a macro diagnostic, and very suitable to micro diagnosis.

Rather than present a full module by module list of which logic the DCS tests, refer to section 7.4 (below), which is an itemized "execution-time" summary of the DCS micro diagnostic tests. The following section lists those areas that the DCS cannot test (ie, uncontrollable or unobservable logic) or



would have required a prohibitive number of microwords to test effectively.

### 7.3.2 UNTESTABLE by DCS

From a functional point of view, the DCS operates on the KD11-K processor from the "inside out". Both the CACHE and KT/KJ are turned off (disabled); there is no FP11-E HFP unit assumed to be present; likewise no ECS/WCS options. No external I/O options are assumed on the UNIBUS, except the standard DL11-W console interface / line clock. The lowest 4K memory bank is assumed to be present. These restrictions impose constraints on the logic that can be exercised by DCS. The following list attempts to detail as specifically as possible, on a module by module basis, those functional areas of the processor that the DCS cannot fully diagnose:

- K2 UNWORD-
  - full effects of processor "INIT"
  - NUR(11) [DCS can't "see" it]
  - CROM contents and address drivers
  - FP11-E related (FLPGO, HFP(CC), etc.)
  - UCON's HFP/KT/WCS
  - HFP FLAG-ROM contents
- K3 IR-DECODE-
  - CROM contents and address drivers
  - CROM extension roms
  - full effects of processor "INIT"
  - full rom contents (location by location) of:
    - BYTE/CC, INSTR-5, FLTPT, CC(V/C), PS(CC)-BRANCH,
    - BYTE-DECODE, KT-DECODE roms
  - KT/KJ enable/select logic
- K4 DATAPATH-
  - full processor "INIT" effects
  - data I/O validity in some A/B/C-SPAD locations
  - SP/BY/KT selection
  - CACHE INVALIDATE logic
- K5 CACHE/KT-
  - processor UNIBUS operation with CACHE enabled
  - processor UNIBUS operation with KT/KJ enabled
  - status conditions of above, including
    - RED/YELLOW ZONE, MM-ABORT, CACHE errors, etc.
- K6 TIMING-
  - ECL clock logic (must be "clocking" ...)
  - 74537 etc CLOCK DRIVERS - mostly HARDWARE
  - MAINTENANCE clock
  - some JAMUPP/PULSE-SUPPRESS conditions (see K7)
  - UNIBUS master arbitration, NPR/BR/PROCESSOR
  - CACHE control (NPR track, HIT, etc.)
    - remember DCS turns CACHE OFF

- UNIBUS address drivers - full test of same
- full check of INTRNL-ADDR rom contents
- full processor "INIT" effects
- K7 STATUS-
  - full processor "INIT" generation/effects
  - POWER FAIL/RESTART logic
  - BR-4/5/7 requests, BG arbitration
  - SACK timeout (BG/NPR)
  - NPR/NPG logic
  - SERVICE conditions:
    - FP11-E/YELLOW-ZONE/CONSOLE/PWRFAIL/CACHE
  - JANUARY conditions:
    - KT-ABORT/RED-ZONE/CACHE+MEMORY+WCS-parity-error
  - CONSOLE interface:
    - KEYCODE input/DISPLAY output/LOCAL and REMOTE
    - [see 11/60 microcode listing for "CONSOLE MICRO TEST", which tests these functions]
  - STATUS mux bits:
    - bits related to above conditions not asserted
    - BATTERY BACKUP for MOS MEMORY OK

#### 7.4 Global Test Order

The following list provides a summary of the major functional blocks of the DCS micro diagnostic code, presented in execution-time order. Note the progression from the innermost portions of the processor logic (microsequencing, IR decode); through the intermediate areas (ALU, SHIFT TREE, etc.); out to the external interface logic (UNIBUS cycles, INTR sequences).

#### NOTE

Most of the capitalized terms refer to specific hardware elements in the KD11-K processor. No attempt is made to explain their meaning - the unfamiliarized reader is referred to the processor logic block diagrams, print set, maintenance manual, and the micro diagnostic code listing for their definition.

The notation "[xxx/yyy]" signifies the micro tests in the range "TESTxxx" to "TESTyyy". All DCS tests are numbered octally, and are executed in ascending order. Some test numbers are further broken down into "TESTxxxA", "TESTxxxB", etc., when their functions are logically similar. Certain test numbers are non-existent (eg, there are no tests with numbers in the range 200-277).

- [001/007] - NUA sequencing logic
- UPF sequencing, page changing

- [010/011] - microsubroutine operation
  - RETURN register, BUTA(SUBR-B)/BUTA(RETURN) decode
- [012A/050B] - IR decode logic and microbranch
  - INSTR -1, -5, -FLTPT decode
  - microbranch selection / execution
  - misc IR decode related microbranch logic
  - processor UCON IR load, EMIT constant generation
- [101] - D-REGISTER/DBUF/BUSDIN/IR datapath
  - D-REGISTER load, ALU control (zeroes)
  - DBUF load, BUSDIN enable (UCON)
- [102A/104B] - C-SCRATCHPAD
  - address modes (2-bit/4-bit), address lines
  - BUSDIN/CSP/ALU-B/D-REGISTER/DBUF/IR datapath, [1s/0s]
- [105A/105E] - SR load/store
  - SR load/store/XMUX-enable
  - BUT(SR<3:0>) microbranch
  - SR/ALU-A/D-REGISTER/DBUF/IR datapath, [1s/0s]
- [114A/122A4] - ALU logic functions, D[C] sources
  - ALU function/mode decode
  - ALU logic function execution
  - D[C] sources (ALUxx, CIN, save); D[C] microbranch
- [130A1/136B2] - ALU arithmetic functions
  - ALU arithmetic function/mode decode
  - ALU arithmetic function execution
  - ALU carry logic (in, out, lookahead)
  - D[C] sources (COUTxx, CIN)
- [320A/320F] - D[C] select logic, D=ZERO logic
  - D[C] 1/8 addressing
  - BUT(D=ZERO) decode logic
- [350A/352D] - A/B-SCRATCHPADS
  - addressing, lines and mode (SF, DF, RIF)
  - data patterns [1s/0s]
- [361A/371B] - SR/GUARD/XMUX, RES control
  - SR shift (left, right, nop)
  - GUARD register (shift, enable/disable, test)
  - RES/SR control
  - FLTPT assemble port
- [372A/372B] - CUA(PROC mux) / BUTA(SUBR-A)
  - BUTA(SUBR-A) decode/execution, RETURN register
- [373A/373B] - JAMUPP and BUTA(DIAGNOSE)
  - active BUTA(DIAGNOSE) decode
  - JAMUPP clock suppress logic via external JAM

- [374A1/376A] - A/B-SCRATCHPADS
  - rewrite modes test (A/B, HI/LO, etc.)
  - BYTE WRITE, DAD control
  - R-IOA-1 / FLTPT-INHIBIT addressing
- [410A/410E] - BYTE / BYTE-CONSTANT / D=ZERO (loop)
  - BYTE/WORD rom decode, microbranch
  - BYTE-CONSTANT CSP addressing, DAD control
  - BUTR(D=ZERO) decode (full test)
- [500C/500F] - PREFETCH / OVERLAP / SP-DEFEAT (loop)
  - PREFETCH rom decode, microbranch
  - OVERLAP rom decode
  - SP-WRITE-DEFEAT decode/control
- [503A/510F] - processor UCON registers / control
  - FLAGS/EXFLAGS - read/write/microbranch
  - FPS - read/write/microbranch
  - PS - read/write/microbranch
  - MULTIPLE BUT - input/select/output
  - SERVICE/INTR decode logic, microbranch
- [511A1/511B4] - MOVE FROM SAME STACK (MFSS) logic
  - decode / microbranch
- [512A1/512E2] - KT SRC/DST addressing logic for ASP/BSP
  - rom decode / control
- [520A/520E] - INSTR BRANCH rom
  - IR/PS inputs, microbranch output
- [533A/537] - SHIFT TREE (AMUX/BMUX/CMUX/SENDMUX)
  - data path [ls/0s]
  - function decode / mux select
  - RES control / select
  - COUNTER load / read
- [551A/551C] - base machine COUNTER
  - active BUTA(COUNT) decode/microbranch
  - COUNTER count execution
- [610A1/610D2] - PS condition code NZVC generation
  - INSTR CLASS decode
  - BYTE/WORD CC mux select
  - CC rom addressing/data
- [620A/624D] - microbreak and JAMUPP
  - MICROBREAK REGISTER load/enable/compare
  - JAMUPP via microbreak, JAM register & STATUS
  - CUA TRACKING, lock/unlock
  - JAMUPP inhibit micro-operation logic
  - JAMUPP CLEAR
- [701A/701D] - BA register

-18. BIT load/read via STATUS mux  
-microbranch conditions

[710A/722C] - UNIBUS function decode, error conditions  
-ODD ADDR/INTERNAL ADDR/SSYN TIMEOUT errors  
-18./16. bit BA modes, I/O PAGE decode, CONSOLE mode  
-SERVICE/JAM register inputs (STATUS mux)  
-bus function decode (DATI, DATO, etc.)

[730A/731E] - UNIBUS cycles to/from memory  
-DATI(B)(P)/DATO(B) execution, side effects  
-UNIBUS data lines, control lines CD, CI  
-ALLOW ODD ADDR, BYTE/WORD operations  
-DBUF, UNIBUS data latches load/enable  
-DATI-CLKIR decode/execution  
-DATIP/PROC-BBSY/BUTA(LAST) logic  
-clock suppress / restart logic

[740A/740D] - UNIBUS cycle function modification  
-BUTA(INSTR-1)/PREFETCH alteration of BUS CODE  
-bus cycle YANK (SP DEFEAT) decode

[761A/763D] - UNIBUS interrupt (BR INTR) logic  
-bus reset, microbranch on status  
-line clock INTR enable, at BR6  
-PS PRIORITY level/INTR PRIORITY level interaction  
-SERVICE port conditions  
-ALLOW BUS GRANT / VECTOR LOAD logic

## 8.0 ERROR HANDLING

### 8.1 What is an 'ERROR' ?

The concept of an 'ERROR' in DCS terms is very simple. It involves the use of the ENUA (Expected NUA), TNUA (Tracking NUA), and DCS COUNTER registers; all of which are local to the DCS module. The 11/60 processor itself has no control over the setting/clearing of 'ERROR'; in fact, it cannot directly determine whether 'ERROR' is set or clear.

The ENUA register (12 bits) is loaded from the EMIT field of the microword, under control of a DCS rom extension bit. It is setup at the beginning of a test to reflect the "EXPECTED" micro address after the test microbranch ("BUT") is executed.

The TNUA register (12 bits) is loaded continuously as the DCS microcode executes, TRACKING the progress of the microaddress field. This register contains the value of the "RECEIVED" micro address after the test microbranch is executed.

The DCS-CNTR is loaded with a value from (00)-(17) (octal), from the EMIT field of the microword, under DCS rom extension control. This register continuously counts up every microcycle. When the contents of this register is (17), the DCS hardware compares ENUA and TNUA, and does the following:

```
Set 'ERROR'="1" if DCS-CNTR=(17) and ENUA<>TNUA
else leave 'ERROR' unchanged from its previous value.
```

This is the manner by which DCS is able to set 'ERROR'. All DCS tests use this method.

Note also that the DCS hardware "locks up" the loading of the ENUA and TNUA registers after 'ERROR' is set, preserving their contents. Thus only the FIRST 'ERROR' will be recorded. There is no provision to detect subsequent errors until the previous ones are eliminated. See the DCS Maintenance Manual and Print Set for more detailed information.

## 8.2 FAULT DIRECTORY Format and Use

### 8.2.1 Basic Structure

The FAULT DIRECTORY is essentially a tabular summary of all ERROR codes the DCS is able to generate - a total of 432 entries occupying 52 pages. Each individual ERROR code entry in the FAULT DIRECTORY contains a short description of the test, and the module replacement information pertaining to that test. For ease of reference, the ERROR codes have been organized into ascending numerical order, in the range 4000(8)-6777(8).

### 8.2.2 Basic Use - with an example

This section describes how to use the FAULT DIRECTORY after the DCS has been run, and has indicated an error is present in the KD11-K processor.

Assume for the purpose of explanation that the DCS was started, and has returned the following values:

ERCD = 4616 (ERror CoDe)  
TNUA = 7405 (Tracking NUA)

with EOP=<OFF>, and ERROR=<ON>

1. Going to the FAULT DIRECTORY, we find the entry for ERROR code 4616 to be on page 9, entry number 73.
2. Some general information about the failing test is first obtained:
  - a) 'Symbolic label' - A reference to the DCS microtest which failed, in this instance TEST-115-A2.
  - b) 'Line number' - A reference to the line number in the DCS microcode listing where the failing test is located (here, line number 5983).
  - c) 'ENUA' - The Expected NUA of this test, in this case 7412. Note that the obtained TNUA (7405) is not the same as the test's ENUA (7412); thus the ERROR.
  - d) The remainder of the line contains a short description of the function performed by this test; in this instance we note the test was diagnosing the ALU portion of the DATA-PATH module.
3. We now note that the TNUA we obtained was 7405. Scanning downward in the column of TNUA entries for this test, we find it listed as the fourth entry. More information, specific to this particular error, can now be obtained:
  - a) 'Module sequence' - These 3+ columns contain (scanning left to right) the top 3+ choices of processor modules to inspect/replace, in order to locate and correct the fault(s). The module choices are listed using "slot" notation (ie, K#, where #=the slot), and a "confidence factor" to indicate the percent confidence that replacing this particular module will eliminate the fault(s). The best choice is the module called out in the first column ("01"); then "02" etc. Note that the percentages are rounded to the nearest 5%, and may therefore not always add up to exactly 100%.
  - b) To the right of the module choices is summarized the

IC information obtained from the FAULT INSERTION effort of the DCS/KD11-K (signified by "FI"). IC information is referenced to a particular module by the notation:

K4=E23,E33-36,E89-E90,E101; K2=E12,E15,E69;

-----  
- N O T E -  
CALLOUT OF SPECIFIC IC'S ON A MODULE IS --NOT-- INTENDED INTENDED TO BE AN "EXHAUSTIVE-ONLY-THESE-ARE-THE-ONES" LIST. IT IS INTENDED TO PROVIDE REFERENCE TO A SPECIFIC FUNCTIONAL AREA OF A MODULE, AND GIVE REFERENCE TO THOSE IC'S WHICH CAUSED THE FAILURE DURING THE FAULT INSERTION EFFORT OF HARD STUCK-HIGH/-LOW AND ADJACENT-PIN-SHORT TYPE FAULTS. AGAIN, DO NOT ASSUME THIS LIST TO BE ALL INCLUSIVE OF THE POSSIBLE CHOICES FOR FAULTY IC'S.  
-----

Another type of entry is of the format:

K404=ALU/CARRY-LOOKAHEAD;  
or Kmpp=functional-description-of-logic-block

which references a particular module (#m) and page (#pp) in the KD11-K Processor Print set. This notation is used when specific fault insertion data is not available for a test.

4. In the instance when the TNUA obtained does not match any of those provided under a given test/ERROR code entry, a wild-card character ("?") has been used to allow a match with any octal digit. Thus 740? matches 7400, 7401, ..., 7407. These entries should be used for further information or when a specific TNUA is not present.
5. If there is no TNUA listed which matches the obtained TNUA, and also no wildcard entry is present; then the information about the functional nature of the test (from above), along with an intelligent interpretation of the obtained TNUA, will be required. The following table lists some TNUA's that might be obtained in such a case:

(see table on following page)



TNUA	Cause
4000	DCS forced to its starting address
4777\ 4756 > 4747/	an unexpected JAMUPP condition occurred
7361	in UNIBUS function tests, a JAMUPP did not occur when expected
7400- 7777	the "standard" BUT() target area for DCS micro-tests

### 8.3 'SCOPE Loop' Facility

#### 8.3.1 General Information

The 'SCOPE loop' implementation provided by DCS is almost identical to that provided in the standard MAINDEC macro diagnostic program. What the 'SCOPE loop' does is to repeatedly execute the same sequence of diagnostic test code; this allows the technician to 'scope' appropriate logic signals in an effort to zero-in on the fault.

The DCS 'SCOPE loop' occurs ONLY and ALWAYS when 'ERROR' is set. There are no user options to change the size or range of the loop - all these parameters have been fixed in microcode and hardware. The loops have been setup to be as tight, and as useful, as was possible. Most are in the range of 10-30 microwords, although some (three, in particular) are larger.

#### 8.3.2 Implementation and Use

A DCS extension rom control bit is used to enable the 'SCOPE loop' check at selected points in DCS code execution. These points are recognized by the following:

```
SCOPE123:
  (possible some other functions)
  NEXT, BUTD[SCOPE], !NO ERROR: "TEST124" [+1. WORDS]
          J/TEST124      ! ERROR: "LOOP123" [-5. WORDS]
```

The two comments "ERROR/NO-ERROR" tell the user where the DCS code will branch, depending upon the current state of 'ERROR'. Usually, the 'NO-ERROR' condition falls thru to the next word (eg, +1. words). For the 'ERROR' case, the loop is ALWAYS backwards (ie, up the page, toward the point where the error was detected). The "-number" notation gives a relative count of the number (approximately) of micro words backwards in the jump.

This facility can be used in two modes - dynamic and static. Either mode must be entered via the use of the DCS 'RUN/STOP' switch set to the 'RUN' position, as this then enables the DCS code to execute continuously. The results are generally undefined if the switch is set to 'STOP', and the 'DIAG' button was used to enter the loop.

Dynamic mode requires the use of an oscilloscope, logic analyzer, etc., and the determination of an appropriate logic signal on which to sync. The DCS microcode then automatically remains in this tight loop to allow observation of the suspected faulty signals, at processor cycle speed.

Static mode is entered in the same manner as dynamic; but afterwards the "SINGLE-MICROSTEP/MAINTENANCE-CLOCK" feature of the 11/60 processor (on K6 TIMING module, the two switches - see prints) is enabled. This allows the processor to be single micro - stepped, under user control. The additional debug features of the DCS can now be employed: the BUSDIN/DOUT display LEDs (16), and the (2) "free" LEDs. See the DCS Maintenance Manual for further details. Also available are the NUA (Next-U-Address) LEDs on the 11/60 processor "UWORD" module (K2). Note that these 'point' at the NEXT microword to be executed, not the current.

## 9.0 REVISION HISTORY

### 9.1 Revision Number

After a successful 'END-OF-PASS' indication, console internal exam functions can be used to obtain the 'DCS Microcode Revision Number', stored in the macro machine general register RS. BIT<15> of this number will also be set, indicating successful 'END-OF-PASS' was reached. The initial version of the microcode will display:

(100101)

with subsequent versions to be:

(100102)  
(100103) etc

If, however, an 'ERROR' is obtained, one might still be able to obtain the DCS microcode revision number. In this instance BIT<15> will/should be clear, and the lower bits the revision code:

(000101)  
(000102) etc

However, this number must be taken with caution, as the error may or may not have influenced the storing of the revision number in the register.

### 9.2 Revisions To DCS Code

Note that ALL revisions to the DCS microcode / FAULT DIRECTORY are to be documented in this section, with the following information supplied:

- a) A short description of the problem(s) found, and how they were corrected.
- b) Updated MAINDEC (MD) and DCS (uCODE) revision information.
- c) Date of fix, and person responsible for fix.
- d) The test/ERROR codes affected by the changes.

Note that the changes MUST also be incorporated into the DCS microcode listing, and/or the FAULT DIRECTORY listing at the appropriate points. Actual microcode changes will be entered as ECO's to individual ROM patterns on the DCS module (M7B71). Contact PDP-11/60 Support Engineering for the procedure.

REVISION MD/uCODE	Date	Who	Explanation
A/101	18-Jan-77	DMN	Initial Release
B/101	22-Jun-77	DMN	No microcode changes; documentation added/updated.

## 10.0 DCS VERIFICATION / SELF-TEST

'VERIFY MODE' is a self-check mode designed to verify the operation of the DCS module and its associated error detection/indication support logic.

### 10.1 Requirements

This mode of operation requires that a known good PDP-11/60 system (as described in Section 2) be used to test/verify a DCS module, so that errors detected by the DCS are due to the DCS module under test, and not due to the other system components. The set of PDP-11/60 processor macro diagnostics, or a known good DCS module, can be used to perform such a verification of the host system.

### 10.2 Verification Method

The method (or algorithm) used to perform the DCS self-verification is as follows:

Hardware on the DCS module is conditioned to execute a single pass thru the DCS microcode, via setting the DCS 'NORMAL/VERIFY' switch to the 'VERIFY' position. This also alters the 12 bit hardware counter on the DCS module from a 'Pass Counter' to the 'Verify Counter'.

At the start of a 'Verify Pass', this counter is preset to a specific value; predetermined so that when 'END-OF-PASS' is signaled by the DCS microcode, this counter will have a value of octal (7777), or be at the point of overflow (carry out) enabled.

As the DCS executes in 'VERIFY MODE', this counter is incremented whenever:

1. A microword is executed from page 7, or
2. A microword is executed with the 'VERIFY' bit (a page 4-6 only DCS ROM extension bit) asserted. These 'VERIFY' bits have been scattered, more or less at random, throughout the DCS microcode. Thus this counter will be incremented at random intervals during a 'Verify Pass'.

The DCS code executes approximately 350,000 microwords per pass; thus the counter will overflow between 2-85 times (depending upon the number of 'VERIFY' bits and page 7 references encountered) before the 'END-OF-PASS' / 'Verify Counter' overflow match. Physically, the verify count is retained modulo 4096 (12 bits), with only the low order bits of the count used in the comparison.

A verification will be considered successful only if a verify counter overflow point exactly matches the microword which signals 'END-OF-PASS' (done only once) in the DCS microcode.

### 10.3 Procedure And Indications

#### 10.3.1 Procedure

To run the DCS in verify mode, the following procedure is followed:

1. Install the DCS in the PDP-11/60 as detailed in Section 3
2. Set the DCS switches:  
'RUN/STOP' = 'STOP' and  
'NORMAL/VERIFY' = 'VERIFY'
3. Now set:  
'RUN/STOP' = 'RUN'
4. The DCS now executes a single 'Verify Pass'
5. At the end of the 'Verify Pass' the DCS enters a microcode loop, in which:
  - An error is forced with specific 'ENUA', 'TNUA', and 'ERROR code' values
  - 'END-OF-PASS' is repeatedly signaled
  - A 'Scope Loop' branch is executed

See the DCS microcode listing, under 'Verify Mode Code' for the exact sequence of operations.

6. At this time examine the DCS module LEDs for comparison with their expected contents, as noted below.
7. To return control to the PDP-11/60 after a 'Verify Pass', position:  
'RUN/STOP' = 'STOP' and  
'NORMAL/VERIFY' = 'NORMAL'

And then generate a "CONSOLE INIT" ("START/HALT") on the operator's console

#### 10.3.2 Indicators

Only the status described below is acceptable to signal a successful DCS verification. Assuming a known good PDP-11/60 system, any deviation from the description (below) should be considered an indication of a fault in the DCS module under

test.

After a 'Verify Pass', indications on the DCS module will be:

'TNUA' = (7522)

'ERROR' = (4255)

(Note the alternating ON/OFF pattern)

'ENUA' = (7523) was loaded to force  
an 'ERROR' indication

'EOP' LED - ON, Approx. 1/2 brilliance  
'ERROR' LED - ON, continuously

This will be the only instance when both  
the 'EOP' and 'ERROR' LEDs should be on  
simultaneously.

Indications on the PDP-11/60 console should be:

"RUN"	LED	-	OFF	continuously
"PROC"	LED	-	OFF	continuously
"USER"	LED	-	OFF	continuously
"CONSOLE"	LED	-	OFF/ dimly lit	
"BATTERY"	LED	-	<indeterminate>	

Octal display = (212121), with the  
decimal points either on or off.

## 11.0 MISCELLANEOUS

### 11.1 ACT/APT/XXDP

The DCS module is not directly supported by ACT/APT/XXDP software at this time.

### 11.2 Macro Instruction Interface

#### 11.2.1 DCS Presence

Presence of a DCS module in slot 1 is indicated by a bit in the "WHAMI" register:

#### NOTE

See PDP-11/60 documentation for a full description of the "MED" instruction.

```
MED      022      ;READ WHAMI -> RO
BIT      %BIT08,RO ;BIT<08>=1, DCS PRESENT
                        ;BIT<08>=0, NO DCS IN SLOT-1
```

#### 11.2.2 DCS Register Access

Access to several of the internal registers and status bits is also possible via the "MED" instruction:

```
MED      ,152      ;READ DCS 'TNUA'
```

After execution, RO's contents is as follows:

```
0000+@TNUA<11:00>
```

Similarly:

```
MED      ,153      ;READ DCS 'EOP/ERROR'
```

and RO's contents:

```
ERROR#01#EOP#ERRCOD<11:00>
```

```
BIT<15> = ERROR(1)H
BIT<14:13> = "01", code for DCS module
BIT<12> = EOP(1)H
```

#### 11.2.3 Macro Instruction Startup Of DCS

The DCS may also be started via a "MED" instruction. Note, however, that this is a one-way transfer of control; there is



no means to re-enter an executing macroinstruction program without operator intervention at the operator's console.

This method simulates the operator depression of the "CNTRL/DIAG" keys via loading the KD11-K microaddress pointer (NUA) with the starting address of the "DIAGNOSE" key service routine.

```
MOV    #011410,RO    ;(1141) = "DIAGNOSE"  
MED    347          ;WRITE NUA  
;NEVER COME BACK TO HERE  
;DCS ALWAYS EXITS TO "CONSOLE" MODE
```

-----  
 (###) ERROR      Symbolic label      Line      ENUA      TNUA      ->Module sequence->  
          code                            number                            #1/%%   #2/%%   #3/%%      Test summary - Print reference - Chip information  
 -----

(###)	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/%% #2/%% #3/%%	Test summary - Print reference - Chip information
1)	4000	TEST001	2633	6252			
							NUA SEQUENCING, PAGE (4) -> (2) [2==6] UBF=(34)
	4000					K3/99 .....	FI; K3=E6, E15
	4203					K2/99 .....	FI; K2=E41, E47
	4252					K2/90 K3/5 K4/5	FI; K2=E13-E14, E34-E35, E40-E41, E46-E47, E58-E59, E62, E64-E65, E67-E68, E72-E74, E78-E80, E91; K3=E34, E111; K4=E3
	4253					K2/99 .....	FI; K2=E14
	4256					K2/99 .....	FI; K2=E34, E46, E59, E68, E74, E80
	4257					K2/99 .....	FI; K2=E40, E65
	4747					K2/75 K3/25	FI; K2=E2, E29, E52-E53, E79; K3=E6, E26
	4756					K2/99 .....	FI; K2=E70
	4757					K3/99 .....	FI; K3=E4, E7
	4776					K2/99 .....	FI; K2=E60
	4777					K2/75 K3/10 K6/5	K4/5 K7/5 FI; K2=E1-E7, E10-E18, E21-E25, E27-E39, E41-E54, E57-E59, E61-E81, E84, E90, E93-E96, E100, E104; K3=E1-E20, E22, E24-E26, E32, E39, E41, E48, E101; K6=E40, E43, E60-E61, E66, E78-E81, E83, E92; K4=E18, E26, E29-E30, E38, E71, E87; K7=E36, E52, E74
	5130					K2/99 .....	FI; K2=E94
	5247					K2/99 .....	FI; K2=E47
	5276					K2/99 .....	FI; K2=E6, E33, E60
	6000					K2/99 .....	FI; K2=E7-E8, E19, E40
	6052					K2/55 K3/45	FI; K2=E1, E7, E117; K3=E58, E91
	6201					K2/99 .....	FI; K2=E70-E71
	6212					K3/55 K2/45	FI; K3=E58, E91; K2=E2, E7
	6231					K2/99 .....	FI; K2=E62
	6235					K2/99 .....	FI; K2=E62
	6242					K3/55 K2/45	FI; K3=E21, E81; K2=E2, E8
	6250					K3/55 K2/45	FI; K3=E21, E81; K2=E2, E8
	6253					K2/55 K3/45	FI; K2=E2, E60, E66, E69, E75, E81; K3=E21, E23, E44, E68
	6256					K2/55 K3/45	FI; K2=E2, E8-E9, E21, E27, E60, E66, E69, E75, E81; K3=E21-E22, E37-E39, E54, E56, E74-E75
	6257					K3/80 K2/20	FI; K3=E37-E38, E74; K2=E8
	6272					K2/95 K3/5	FI; K2=E2-E3, E6-E7, E12, E15, E18, E24, E30, E36; K3=E52
	6277					K2/99 .....	FI; K2=E2, E13
	6352					K2/80 K3/20	FI; K2=E1, E3, E6-E7, E12, E15, E18, E24, E30, E36; K3=E51, E58, E70
	6372					K2/99 .....	FI; K2=E7
	6652					K2/80 K3/20	FI; K2=E1, E19, E42, E59, E65, E68, E74, E90; K3=E23, E63
	7077					K2/99 .....	FI; K2=E117
	7231					K2/99 .....	FI; K2=E14
	7252					K2/99 .....	FI; K2=E35, E47, E58
	7752					K2/99 .....	FI; K2=E48
	7777					K2/99 .....	FI; K2=E48

-----  
 Module codes:    K1/DCS    K2/UWORD    K3/IRDECODE    K4/DATAPATH    K5/KTCACHE    K6/TIMING    K7/STATUS  
 -----

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/% #2/% #3/%	Test summary - Print reference - Chip information
2)	4255	VFY003	18313	7523	7522	K6/50 K7/40 K3/5	THIS ERROR-CODE EXPECTED IN DCS "VERIFY MODE" THIS TNUA EXPECTED IN DCS "VERIFY MODE" FI: K6=E3-E5, E12, E14, E22, E26, E31-E32, E34, E37-E40, E42-E43, E46, E54, E56, E61-E62, E64, E69, E71-E72, E76, E78-E81, E86-E88, E90, E94-E96, E98-E102, E104-E105, E108; K7=E2, E4, E8, E12, E14-E15, E18-E23, E25-E27, E34-E35, E37-E40, E43, E45-E47, E49, E52, E54, E59, E62-E63, E66, E68, E70-E71, E87, E95, E97-E98; K3=E2, E6-E8, E10, E12, E15, E21-E26, E34, E52, E54, E64; K2=E2-E3, E6-E9, E21, E36, E50, E60, E66, E75, E81; K4=E8, E16
					7777	K1/99	DCS VERIFY MODE, BAD TNUA LATCHED
3)	4377	TEST007	2749	7303	7301	K2/99	NUA SEQUENCING, PAGE (4) -> (3) [3==7], UBF=(35) FI: K2=E61
					7303	K2/65 K4/35 K3/5	FI: K2=E4, E10, E16, E22, E25-E26, E28, E31, E37, E43, E49, E52, E70-E71, E76-E77, E97; K4=E1, E12-E15, E29, E38, E70; K3=E33; K6=E34
					7307	K2/60 K3/40	FI: K2=E9, E21, E27; K3=E21-E22
					7343	K2/99	FI: K2=E7
					7376	K3/99	FI: K3=E34
					7777	K2/99	FI: K2=E23, E29
4)	4450	TEST730C1	17211	7402	7400	K5/99	BUTA(LAST) CLEARS PROC BBSY (DATIP); EMIT=(052525) ON BUS0 IN FI: K5=E94
					7401	K6/50 K7/35 K5/15	FI: K6=E4, E61, E70, E79, E103; K7=E31, E39, E46, E49; K5=E95, E97, E105
5)	4451	TEST7400	17653	7402	4747	K5/65 K6/35	INSTRI*-OVERLAP, BUS CYCLE YANKED FI: K5=E95; K6=E40
					7401	K6/99	FI: K6=E43
6)	4452	TEST740C	17619	7402	4747	K2/99	-INSTRI*PREFETCH, BC<0>=BC<0> FI: K2=E41
					7401	K2/99	FI: K2=E105
7)	4453	TEST740B	17591	7402	7401	K2/65 K3/35	INSTRI*PREFETCH, BC<0>="0" FI: K2=E82, E105; K3=E46
8)	4454	TEST731E	17467	7432	4747	K3/99	BUS DATI-CLKIR, BA=(000000); IR=(000125) E88/(432) LOADED FI: K3=E1
					7402	K6/99	FI: K6=E5
					7407	K6/99	FI: K6=E4, E108
					7412	K6/85 K3/15	FI: K6=E3, E5; K3=E45

Module codes: K1/DCS K2/WORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence->			Test summary - Print reference - Chip information
						#1/%%	#2/%%	#3/%%	
9)	4455	TEST731D	17428	7402	BUS DATI-CLKIR, BA=(000000); SERVICE=(000340)	4747 K5/99	../. .	../. .	FI; K5=E39
					7400 K6/99	../. .	../. .	FI; K6=E108	
10)	4456	TEST731C	17401	7432	BUS DATI, BA=(000000); MEM(000000)=(000125) VIA DATO/DATOB/DATI	7400 K6/99	../. .	../. .	FI; K6=E93
					7402 K6/99	../. .	../. .	FI; K6=E36, E43, E85, E90, E93	
					7405 K4/99	../. .	../. .	FI; K4=E28	
					7420 K5/99	../. .	../. .	FI; K5=E94, E105	
					7434 K6/85	K7/15	../. .	FI; K6=E11, E55, E71; K7=E49	
11)	4457	TEST731B	17363	7412	IR=(125200) E78/(412); NO BUS FUNCTIONS CLOCK-IR EXCEPT DATI-CLKIR	7400 K2/65	K3/35	../. .	FI; K2=E34, E84; K3=E48
12)	4460	TEST730C	17188	7402	DATIP HOLDING BUS (PROC BBSY); DU=(125252) ON BUSDIN	7401 K6/65	K7/35	../. .	FI; K6=E70, E79, E103; K7=E17, E62
13)	4461	TEST730D	17233	7402	BUS DATO, BA=(000000), DATIB*BYTE*000, BA=(000001); SERVICE=(000340)	4747 K6/50	K2/50	../. .	FI; K6=E47; K2=E105
					7400 K6/90	K7/15	../. .	FI; K6=E38-E39, E54; K7=E49	
					7401 K6/99	../. .	../. .	FI; K6=E38	
14)	4462	TEST730E	17286	7402	BUS DATO, BA=(000000), DATIB*BYTE*000, BA=(000001); DU=BUSDIN=(052525)	7400 K6/90	K5/10	K4/5	FI; K6=E5, E46, E66-E67, E72, E74-E75, E80, E83-E84, E86, E88, E90, E92-E93, E108; K5=E53, E63, E99-E101, E109, E112-E113, E115, E122; K4=E28
					7403 K6/99	../. .	../. .	FI; K6=E92-E93	
15)	4464	TEST722B	17004	7402	INVALIDATE ODD ADDR JAM; JAM=(101004)	???. K6/99	../. .	../. .	K605=UNIBUS-FUNCTION-DECODE
16)	4465	TEST722C	17029	7402	INVALIDATE 16. BIT PBA, -I/O PAGE(6); SERVICE=(002340)	7400 K6/99	../. .	../. .	FI; K6=E59
					???. K6/99	../. .	../. .	K605=UNIBUS-FUNCTION-DECODE	
17)	4466	TEST730B	17149	7402	BUS DATIP, BA=(000001); DU=(125252) AFTER ON BUSDIN	4747 K7/65	K6/35	../. .	FI; K7=E16, E25, E32; K6=E55, E76, E88
					7400 K6/95	K5/5	../. .	FI; K6=E66-E67, E74-E75, E83-E84, E92-E93; K5=E53, E63, E66	
					7401 K6/65	K7/35	../. .	FI; K6=E11, E19, E47, E55, E63, E66-E67, E69, E71, E74-E76, E79, E83-E84, E92-E93, E98-E99, E101, E103, E105; K7=E10, E17, E21, E32-E33, E49, E101	
					7403 K6/99	../. .	../. .	FI; K6=E92-E93	

-----  
 Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/% #2/% #3/%	Test summary - Print reference - Chip information
18)	4471	TEST730A	17084	7402	4747	K6/65 K7/25 K5/10	BUS DATA, BA=(000000), D=(125252); CHECK DBUF=0 AFTER FI; K6=E2-E3, E19-E20, E26, E40, E76, E85, E87-E88, E96-E98; K7=E2, E10, E32-E33, E36, E57-E58, E60; K5=E35, E42-E43, E62
					7400	K5/99	FI; K5=E90, E107-E108, E114, E120-E121
					7401	K6/65 K7/20 K4/10	K3/5 FI; K6=E11, E56, E64, E69, E72, E85, E101, E103; K7=E10, E33; K4=E28, E55; K3=E42
					7403	K6/50 K5/25 K2/25	FI; K6=E101; K5=E91; K2=E100
19)	4472	TEST13282	7572	7417	7413	K4/90 K3/10	BUTR(X#D(C)XX), D(C)=COUT15="1"; A(0)+B(1)+C(1)=D(0)+C(1) K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
20)	4473	TEST13682	8010	7402	7403	K4/90 K3/10	(122645)+(132264)+(0)=(170360); CHECK COUT15=(0) K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
21)	4474	TEST006	2730	4377	4777	K2/99	MUA SEQUENCING, NO 'BUT' FI; K2=E17, E63
22)	4475	TEST13681	7983	7434	7400	K4/99	(122645)+(132264)+(0)=(170360); CARRY LOOKAHEAD LOGIC FI; K4=E47
					74??	K4/90 K3/10	K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
23)	4476	TEST136A2	7961	7403	7266	K2/80 K3/20	(055132)+(132264)+(1)=(007417); CHECK COUT15=(1) FI; K2=E8-E9, E21, E27; K3=E21
					7402	K4/90 K3/10	K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
24)	4477	TEST13582	7904	7413	7417	K4/90 K3/10	(055132)+(113226)+(0)=(170360); CHECK COUT15=(0) K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
25)	4500	TEST133A2	7630	7403	7402	K4/90 K3/10	(103607)+(103607)+(1)=(007417); CHECK COUT15=(1) K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
26)	4501	TEST13381	7652	7434	7402	K4/50 K2/50	(074170)+(074170)+(0)=(170360); CARRY LOOKAHEAD LOGIC FI; K4=E49; K2=E102
					7420	K4/99	FI; K4=E49
					7421	K3/99	FI; K3=E71
					74??	K4/90 K3/10	K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
27)	4502	TEST13382	7679	7402	7403	K4/90 K3/10	(074170)+(074170)+(0)=(170360); CHECK COUT15=(0) K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
28)	4503	TEST134A2	7742	7403	7402	K4/90 K3/10	(045513)+(141703)+(1)=(007417); CHECK COUT15=(1) K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE

-----  
Module codes: K1/DCS K2/WORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence->			Test summary - Print reference - Chip information
						#1/%%	#2/%%	#3/%%	
29)	4504	TEST134B1	7764	7434	(132264)+(036074)+(0)=(170360); CARRY LOOKAHEAD LOGIC 74?? K4/90 K3/10 .....	K404=ALU/CARRY-LOOKAHEAD;	K313=ALU-FCN-DECODE		
30)	4505	TEST134B2	7791	7401	(132264)+(036074)+(0)=(170360); CHECK COUT15=(0) 7403 K5/99 .....	FI; K5=E95			
					740? K4/90 K3/10 .....	K404=ALU/CARRY-LOOKAHEAD;	K313=ALU-FCN-DECODE		
31)	4506	TEST135A2	7855	7417	(122645)+(064551)+(1)=(007417); CHECK COUT15=(1) 7413 K4/90 K3/10 .....	K404=ALU/CARRY-LOOKAHEAD;	K313=ALU-FCN-DECODE		
32)	4507	TEST135B1	7877	7434	(055132)+(113226)+(0)=(170360); CARRY LOOKAHEAD LOGIC 74?? K4/90 K3/10 .....	K404=ALU/CARRY-LOOKAHEAD;	K313=ALU-FCN-DECODE		
33)	4511	TEST134A1	7705	7434	(045513)+(141703)+(1)=(007417); CARRY LOOKAHEAD LOGIC 74?? K4/90 K3/10 .....	K404=ALU/CARRY-LOOKAHEAD;	K313=ALU-FCN-DECODE		
34)	4513	TEST135A1	7818	7434	(122645)+(064551)+(1)=(007417); CARRY LOOKAHEAD LOGIC 74?? K4/90 K3/10 .....	K404=ALU/CARRY-LOOKAHEAD;	K313=ALU-FCN-DECODE		
35)	4515	TEST136A1	7931	7434	(055132)+(132264)+(1)=(007417); CARRY LOOKAHEAD LOGIC 7421 K4/75 K3/25 .....	FI; K4=E64; K3=E74			
					74?? K4/90 K3/10 .....	K404=ALU/CARRY-LOOKAHEAD;	K313=ALU-FCN-DECODE		
36)	4516	TEST132A2	7527	7403	BUTR(D(C)BA00), D(C)=COUT15="1"; A(1)+B(0)+C(1)=D(0)+C(0) 7402 K4/90 K3/10 .....	K404=ALU/CARRY-LOOKAHEAD;	K313=ALU-FCN-DECODE		
37)	4517	TEST132B1	7547	7434	ALU=DIVIDE=A-MINUS-B; A(052525), B(052525), D(000000), D(C)="1" 7400 K3/80 K4/20 .....	FI; K3=E41,E53,E61; K4=E16			
					7402 K3/99 .....	FI; K3=E51			
					74?? K4/90 K3/10 .....	K404=ALU/CARRY-LOOKAHEAD;	K313=ALU-FCN-DECODE		
38)	4521	TEST132A1	7502	7434	ALU=A-MINUS-B; A(125252), B(125252), D(000000) 7400 K4/70 K3/30 .....	FI; K4=E49; K3=E42,E92			
					7402 K4/55 K3/45 .....	FI; K4=E6,E11,E17,E26,E96; K3=E41,E82			
					74?? K4/90 K3/10 .....	K404=ALU/CARRY-LOOKAHEAD;	K313=ALU-FCN-DECODE		
39)	4523	TEST133A1	7600	7434	(103607)+(103607)+(1)=(007417); CARRY LOOKAHEAD LOGIC 7400 K4/75 K3/25 .....	FI; K4=E8,E49,E71; K3=E53			
					7402 K3/99 .....	FI; K3=E43,E84			
					7420 K4/99 .....	FI; K4=E49,E88			
					74?? K4/90 K3/10 .....	K404=ALU/CARRY-LOOKAHEAD;	K313=ALU-FCN-DECODE		
40)	4530	TEST130B1	7334	7434	ALU=A-PLUS-B-PLUS-1; A(125252), B(125252), D(052525) 7400 K4/70 K3/15 K2/15 .....	FI; K4=E49,E101-E102,E110; K3=E53; K2=E50			
					[Continued] 7402 K4/80 K3/20 .....	FI; K4=E49,E98,E101; K3=E41			

-----  
 Module codes: K1/DCS K2/LWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/% #2/% #3/%			Test summary - Print reference - Chip information
	4530	TEST13081	[Continued]						ALU=A-PLUS-B-PLUS-1; A(125252), B(125252), D(052525) 7406 K4/99 ..../.. FI; K4=E110 7420 K4/99 ..../.. FI; K4=E49,E102,E112 7421 K4/99 ..../.. FI; K4=E71 7436 K4/99 ..../.. FI; K4=E112 74?? K4/90 K3/10 ..../.. K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
41)	4532	TEST13082	7365	7403	BUTR(COUNT#D(C)), D(C)=COUT15="1"; A(1)+B(1)+CI(0)=D(0)+CO(1) 7402 K2/50 K3/30 K4/25 FI; K2=E35,E41,E61,E77,E85; K3=E4,E6-E7,E11,E15,E43; K4=E63-E64				
42)	4537	TEST3200	8197	7403	BUTR(COUT07#DOUT07#XX), TARGET="01*" 7401 K3/85 K4/15 ..../.. FI; K3=E53,E55-E56,E95; K4=E10 7407 K3/99 ..../.. FI; K3=E74				
43)	4541	TEST131A1	7392	7434	ALU=A-PLUS-B-PLUS-PS(C); A(125252), B(052525), D(177777), PS(C)=(0) 7400 K3/75 K4/25 ..../.. FI; K3=E35,E41,E46,E71,E84; K4=E49 7402 K3/99 ..../.. FI; K3=E82 74?? K4/90 K3/10 ..../.. K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE				
44)	4543	TEST320A	8098	7403	D(C) ADDR SELECT, CODE="010"=ALU00="1" ONLY ONE SET 7402 K4/99 ..../.. K404=D(C)-SELECT-LOGIC				
45)	4544	TEST131A2	7423	7401	BUTR(D(C)#BA00), D(C)=COUT15="0"; A(1)+B(0)+CI(0)=D(1)+CO(0) 7403 K4/90 K3/10 ..../.. K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE				
46)	4546	TEST131B1	7443	7434	ALU=DIVIDE=A-PLUS-B; A(052525), B(125252), D(177777), D(C)="0" 7400 K3/99 ..../.. FI; K3=E82 7402 K3/99 ..../.. FI; K3=E61,E84 74?? K4/90 K3/10 ..../.. K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE				
47)	4550	TEST130A2	7314	7402	BUTR(COUNT#D(C)), D(C)=COUT15="0"; A(0)+B(0)+CI(1)=D(1)+CO(0) 7403 K4/99 ..../.. FI; K4=E63-E64				
48)	4552	TEST320B	8138	7400	BUT(D<14:00>=0#015), D=(000001), TARGET="00" 7402 K2/99 ..../.. FI; K2=E40,E88,E94,E99,E112				
49)	4553	TEST762A1	17899	NONE	DL11-W ENABLED FOR BR6 INTR; DID NOT RESPOND W/ BG-SERVICE-L WITHIN 22. MS. 4553 K7/85 K6/10 K5/5 K3/5 FI; K7=E3,E5-E6,E11,E13,E16-E17,E19-E20,E25-E26,E28,E35-E36,E39,E76,E80; K6=E25,E34,E62; K5=E1; K3=E74				
50)	4554	TEST320E	8217	7403	D(C) ADDR SELECT, CODE="101"=COUT07="1" ONLY ONE SET 7401 K4/99 ..../.. FI; K4=E63				
51)	4556	TEST320F	8256	7405	BUTR(COUT07#DOUT07#XX), TARGET="10*" 7407 K3/99 ..../.. FI; K3=E56,E95				

Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7 STATUS

###	ERROR code	Symbolic label	Line number	EMUA	TNUA	->Module sequence-> #1/%% #2/%% #3/%%	Test summary - Print reference - Chip information
52)	4562	TEST320C	8157	7413	BUTR(X#D(C)XX), D(C)=COUT07="0", COUT15="1"	7417 K4/99 .....	FI; K4=E63-E64
53)	4566	TEST551C	14539	NONE	B.M. DATAPATH COUNTER, BUTA(COUNT-IS-377#D(E)), COUNT FROM (000):(377)	4531 K3/99 .....	FI; K3=E56
					7402 K2/80 K4/20 .....	K207=ACTIVE-BUT; K409=CNTR	
54)	4570	TEST122A2	7172	7403	BUTR(COUNT#D(C)), D(C)=CINMUX="1", ALU=A-I08-B	7402 K3/99 .....	FI; K3=E43,E61,E71
55)	4571	TEST721A	16840	4777	DAT08*-BYTE, ODD ADDR JAM, BA=(120001)	7341 K6/99 .....	K605=UNIBUS-FUNCTION-DECODE
56)	4572	TEST537A	14413	7402	SHIFTER, 'SENDMUX' CMUX<00> OUTPUT ON CMUX/LEFT-1	7400 K4/90 K5/5 K3/5	FI; K4=E77-E78,E80; K5=E97; K3=E61
57)	4573	TEST721B	16894	7402	DAT08*-BYTE, ODD ADDR JAM; JAM=(101004)	7407 K7/99 .....	K705=JAM-FLAGS, K708=STATUS-MUX
58)	4575	TEST551A	14469	NONE	B.M. DATAPATH COUNTER, BUTA(SR1-0#COUNT-IS-377), COUNT FROM (000):(377)	4531 K3/75 K4/25 .....	FI; K3=E36,E54; K4=E91
					4547 K3/90 K4/10 .....	FI; K3=E36,E54,E56,E64; K4=E91	
					4555 K4/90 K2/10 .....	FI; K4=E20,E84,E87,E90-E91; K2=E55	
					7402 K2/80 K4/20 .....	K207=ACTIVE-BUT; K409=CNTR	
59)	4576	TEST551B	14498	7400	B.M. DATAPATH COUNTER, BUTA(COUNT-IS-377), COUNT FROM (000):(377)	7401 K3/99 .....	FI; K3=E64
					7402 K3/99 .....	FI; K3=E44	
					7402 K2/80 K4/20 .....	K207=ACTIVE-BUT; K409=CNTR	
60)	4577	TEST721C	16918	7402	DAT08*-BYTE, 16. BIT PBA, -I/O PAGE(5); SERVICE=(002340)	7400 K6/99 .....	FI, K6=E59
61)	4601	TEST115A1	5939	7412	ALU=NOT-A; A(052525), B(177777), D(125252), BITS<15:12>=(12)	7400 K4/85 K7/10 K2/5	FI; K4=E1,E4-E5,E7,E15-E16,E22,E24,E34,E40-E42, E98; K7=E41,E70; K2=E70
					7402 K4/99 .....	FI; K4=E33-E34	
					7403 K4/75 K2/25 .....	FI; K4=E26; K2=E39	
					7405 K4/55 K3/45 .....	FI; K4=E14,E21,E23; K3=E43,E52-E53,E84	
					7407 K4/99 .....	FI; K4=E4	
					7410 K4/99 .....	FI; K4=E34	
					7413 K4/99 .....	FI; K4=E26,E34,E62	
					7416 K4/99 .....	FI; K4=E17,E34,E54	
					7417 K4/99 .....	FI; K4=E1,E6,E8,E10,E14-E18,E20-E21,E23-E24, E26-E27,E29,E37,E44,E46	

Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS



###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/% #2/% #3/%			Test summary - Print reference - Chip information	
62)	4602	TEST11584	6044	7402	BUTR(COUNT#D(C)), D(C)=PS(C)="0"	7403	K4/50	K3/45	K2/10	FI; K4=E2-E3,E56,E63; K3=E36,E57,E63; K2=E70
63)	4603	TEST1'4A2	5906	7402	BUTR(COUNT#D(C)), D(C)=CINMUX=PS(C)="0", ALU=ZERO	7403	K4/50	K3/50	../.:	FI; K4=E56,E63,E72,E78; K3=E22,E41,E54,E71-E73
						7406	K3/99	../.:	../.:	FI; K3=E74
64)	4604	TEST115A4	5967	7403	BUTR(COUNT#D(C)), D(C)=CINMUX="1", ALU=NOT-A	7402	K4/55	K3/45	../.:	FI; K4=E2-E3,E56,E63,E71-E72,E78,E86; K3=E41,E43,E46,E54,E61-E62,E71,E75
65)	4605	TEST121C2	7009	7412	ALU=A-XOR-B; A(000000), B(125252), D(125252), BITS<11:06>=(52)	7434	K7/99	../.:	../.:	FI; K7=E20
						74??	K4/90	K3/10	../.:	K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
66)	4606	TEST1168	6326	7434	ALU=ZERO; A(125252), B(052525), D(000000)	7402	K3/99	../.:	../.:	FI; K3=E102
						74??	K4/90	K3/10	../.:	K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
67)	4607	TEST114A	5874	7434	ALU=ZERO; A(177777), B(177777), D(000000)	7400	K3/99	../.:	../.:	FI; K3=E43,E82
						74??	K4/90	K3/10	../.:	K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
68)	4611	TEST115C1	6104	7412	ALU=NOT-A; A(052525), B(000000), D(125252), BITS<15:12>=(12)	7400	K3/99	../.:	../.:	FI; K3=E51,E82
						7405	K4/99	../.:	../.:	FI; K4=E6,E17,E23,E31-E34,E40-E43
						74??	K4/90	K3/10	../.:	K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
69)	4612	TEST115B3	6077	7432	ALU=NOT-A; A(125252), B(177777), D(052525), BITS<05:00>=(25)	7420	K4/90	K5/10	../.:	FI; K4=E28,E37,E40-E41; K5=E66
						74??	K4/90	K3/10	../.:	K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
70)	4613	TEST115B2	6060	7405	ALU=NOT-A; A(5:5252), B(177777), D(052525), BITS<11:06>=(25)	7400	K4/90	K2/10	../.:	FI; K4=E37,E41-E42,E89,E108; K2=E26
						7402	K4/99	../.:	../.:	FI; K4=E98
						74??	K4/90	K3/10	../.:	K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
71)	4614	TEST115B1	6021	7405	ALU=NOT-A; A(125252), B(177777), D(052525), BITS<15:12>=(05)	7400	K4/99	../.:	../.:	FI; K4=E16
						7401	K4/99	../.:	../.:	FI; K4=E34
						7404	K4/99	../.:	../.:	FI; K4=E34
						7407	K4/99	../.:	../.:	FI; K4=E34
						7412	K4/99	../.:	../.:	FI; K4=E14,E23
						7415	K4/99	../.:	../.:	FI; K4=E34
						7417	K4/90	K6/10	../.:	FI; K4=E4,E12-E13,E21-E24; K6=E34

Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

ERROR ###) code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/% #2/% #3/%	Test summary - Print reference - Chip information
72) 4615	TEST115A3	6000	7425	7420	ALU=NOT-A; A(052525), B(177777), D(125252), BITS<05:00>=(52) K4/90 K5/10 ..../ FI; K4=E31,E37,E40-E41,E44,E61,E75-E76; K5=E63,E66 74?? K4/90 K3/10 ..../ K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE	
73) 4616	TEST115A2	5983	7412	7400	ALU=NOT-A; A(052525), B(177777), D(125252), BITS<11:06>=(52) K4/80 K2/15 K3/5 FI; K4=E5-E7,E11,E17-E18,E20,E24,E26,E28,E32, E41-E43,E60,E66-E68,E88-E89,E108; K2=E32-E33, E38-E39,E44-E45,E50-E51,E53-E54; K3=E27,E41,E82 7402 K3/99 ..../ FI; K3=E53,E61 7403 K5/99 ..../ FI; K5=E53 7405 K4/90 K3/15 ..../ FI; K4=E10,E22,E24,E98; K3=E84 7434 K4/50 K2/50 ..../ FI; K4=E71; K2=E117 74?? K4/90 K3/10 ..../ K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE	
74) 4617	TEST115C2	6126	7412	7400	ALU=NOT-A; A(052525), B(000000), D(125252), BITS<11:06>=(52) K4/99 ..../ FI; K4=E41-E42 74?? K4/90 K3/10 ..../ K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE	
75) 4620	TEST116A3	6271	7412	7400	ALU=NOT-A-AND-B; A(000000), B(125252), D(125252), BITS<11:06>=(52) K4/99 ..../ FI; K4=E41-E42,E89 74?? K4/90 K3/10 ..../ K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE	
76) 4621	TEST116A2	6257	7403	7402	BUTR(COUNT#D(C)), D(C)=ALU15="1" D=(125252) K4/99 ..../ FI; K4=E2-E3,E56,E63-E64	
77) 4623	TEST116A1	6235	7412	7400	ALU=NOT-A-AND-B; A(000000), B(125252), D(125252), BITS<15:12>=(12) K4/99 ..../ FI; K4=E15,E24,E31-E34,E40-E43 74?? K4/90 K3/10 ..../ K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE	
78) 4624	TEST115D3	6203	7432	74??	ALU=NOT-A; A(125252), B(000000), D(052525), BITS<05:00>=(25) K4/90 K3/10 ..../ K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE	
79) 4625	TEST115D2	6186	7405	74??	ALU=NOT-A; A(125252), B(000000), D(052525), BITS<11:06>=(25) K4/90 K3/10 ..../ K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE	
80) 4626	TEST115D1	6164	7405	74??	ALU=NOT-A; A(125252), B(000000), D(052525), BITS<15:12>=(05) K4/90 K3/10 ..../ K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE	
81) 4627	TEST115C3	6143	7425	7420	ALU=NOT-A; A(052525), B(000000), D(125252), BITS<05:00>=(52) K4/50 K3/50 ..../ FI; K4=E40; K3=E11 74?? K4/90 K3/10 ..../ K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE	
82) 4630	TEST116C5	6429	7401	7403	BUTR(D(C)#BA00), D(C)=D(C)="0" K4/99 ..../ FI; K4=E63-E64,E78	

Module Codes: K1/DCS K2/WORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/% #2/% #3/%			Test summary - Print reference - Chip information	
83)	4631	TEST116C4	6413	7432	ALU=NOT-A-AND-B; A(000000), B(052525), D(052525), BITS<05:00>=(25)	4631	K7/99	../. ..	FI; K7=E43	
					74??	K4/90	K3/10	../. ..	K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE	
84)	4632	TEST116C3	6396	7405	ALU=NOT-A-AND-B; A(000000), B(052525), C(052525), BITS<11:06>=(25)	74??	K4/90	K3/10	../. ..	K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
85)	4633	TEST116C2	6381	7402	BUTR(COUNT#D(C)), D(C)=ALU15="0", D=(052525)	7403	K4/99	../. ..	FI; K4=E63-E64	
86)	4635	TEST116C1	6359	7405	ALU=NOT-A-AND-B; A(000000), B(052525), D(052525), BITS<15:12>=(05)	74??	K4/90	K3/10	../. ..	K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
87)	4636	TEST116A5	6305	7403	BUTR(D(C)#A00), D(C)=D(C)="1"	4747	K2/99	../. ..	FI; K2=E46, E55	
					7401	K4/90	K3/10	../. ..	FI; K4=E2-E3, E56, E63-E64, E93, E104, E113-E114; K3=E44, E62	
88)	4637	TEST116A4	6288	7425	ALU=NOT-A-AND-B; A(000000), B(125252), D(125252), BITS<05:00>=(52)	7420	K4/99	../. ..	FI; K4=E40, E88	
					74??	K4/90	K3/10	../. ..	K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE	
89)	4640	TEST117B2	6592	7412	ALU=A-AND-NOT-B; A(177777), B(052525), D(125252), BITS<11:06>=(52)	7400	K4/99	../. ..	FI; K4=E108	
					74??	K4/90	K3/10	../. ..	K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE	
90)	4641	TEST117B1	6570	7412	ALU=A-AND-NOT-B; A(177777), B(052525), D(125252), BITS<15:12>=(12)	7400	K4/99	../. ..	FI; K4=E4, E10, E23-E24	
					7410	K4/99	../. ..	FI; K4=E80		
					74??	K4/90	K3/10	../. ..	K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE	
91)	4642	TEST117A4	6549	7413	BUTR(X#D(C)#XX), D(C)=CINMUX=D(C)="0", ALU=A-AND-NOT-B	7417	K3/65	K4/35	../. ..	FI; K3=E71, E75; K4=E63
					7433	K3/99	../. ..	FI; K3=E68		
					7453	K3/99	../. ..	FI; K3=E70		
					7513	K3/99	../. ..	FI; K3=E70		
					7613	K3/99	../. ..	FI; K3=E70		
					7653	K3/99	../. ..	FI; K3=E70		
92)	4643	TEST117A3	6534	7432	ALU=A-AND-NOT-B; A(177777), B(125252), D(052525), BITS<05:00>=(25)	7420	K4/65	K3/35	../. ..	FI; K4=E88-E89; K3=E61
					74??	K4/90	K3/10	../. ..	K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE	
93)	4644	TEST117A2	6517	7405	ALU=A-AND-NOT-B; A(177777), B(125252), D(052525), BITS<11:06>=(25)	[Continued]	7402	K7/99	../. ..	FI; K7=E37

Module codes: K1/DCS K2/UMWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###	ERROR code	Symbolic label	Line number	ENUA	->Module sequence->			Test summary - Print reference - Chip information
					TNUA	#1/%%	#2/%%	
	4644	TEST117A2	[Continued]					ALU=A-AND-NOT-B; A(177777), B(125252), D(052525), BITS<11:06>=(25) 7434 K5/99 ..../i0 ..../: FI: K5=E95 74?? K4/90 K3/i0 ..../: K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
94)	4645	TEST1160	6452	7434				ALU=ZERO; A(052525), B(125252), D(000000) 74?? K4/90 K3/i0 ..../: K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
95)	4647	TEST117A1	6495	7405				ALU=A-AND-NOT-B; A(177777), B(125252), D(052525), BITS<15:12>=(05) 7400 K4/85 K3/15 ..../: FI: K4=E5, E12, E21, E23; K3=E71 7417 K3/99 ..../: FI: K3=E84 74?? K4/90 K3/i0 ..../: K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
96)	4650	TEST12084	6799	7402				BUTR(COUNT#D(C)), D(C)=CINMUX="0", ALU=A-AND-B 7403 K3/99 ..../: ..../: FI; K3=E71
97)	4651	TEST12083	6784	7432				ALU=A-AND-B; A(052525), B(177777), D(052525), BITS<05:00>=(25) 7420 K4/99 ..../: ..../: FI: K4=E40, E88 74?? K4/90 K3/i0 ..../: K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
98)	4652	TEST12082	6767	7405				ALU=A-AND-B; A(052525), B(177777), D(052525), BITS<11:06>=(25) 7400 K4/99 ..../: ..../: FI: K4=E41-E42 74?? K4/90 K3/i0 ..../: K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
99)	4653	TEST12081	6745	7405				ALU=A-AND-B; A(052525), B(177777), D(052525), BITS<15:12>=(05) 7412 K4/99 ..../: ..../: FI: K4=E13, E15, E20, E22, E24, E31-E34, E40-E43 74?? K4/90 K3/i0 ..../: K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
100)	4654	TEST120A3	6724	7425				ALU=A-AND-B; A(125252), B(177777), D(125252), BITS<05:00>=(52) 74?? K4/90 K3/i0 ..../: K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
101)	4655	TEST120A2	6707	7412				ALU=A-AND-B; A(125252), B(177777), D(125252), BITS<11:06>=(52) 74?? K4/90 K3/i0 ..../: K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
102)	4656	TEST117C1	6633	7434				ALU=A-AND-NOT-B; A(000000), B(000000), D(000000) 74?? K4/90 K3/i0 ..../: K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
103)	4657	TEST117B3	6609	7425				ALU=A-AND-NOT-B; A(177777), B(052525), D(125252), BITS<05:00>=(52) 7420 K4/99 ..../: ..../: FI: K4=E88-E89 74?? K4/90 K3/i0 ..../: K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
104)	4660	TEST121B4	6960	7401				BUTR(D(C)#BA00), D(C)=ALU00="0", D=(125252) 7403 K4/99 ..../: ..../: FI; K4=E56, E63
105)	4661	TEST121B3	6945	7425				ALU=A-XOR-B; A(177777), B(052525), D(125252), BITS<05:00>=(52) 74?? K4/90 K3/i0 ..../: K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
106)	4662	TEST121B2	6928	7412				ALU=A-XOR-B; A(177777), B(052525), D(125252), BITS<11:06>=(52) 74?? K4/90 K3/i0 ..../: K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE

-----  
Module codes: K1/DCS K2/UMWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

000)	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> 01/%% 02/%% 03/%%			Test summary - Print reference - Chip information
107)	4663	TEST121B1	6906	7412	ALU=A-XOR-B;	A(177777),	B(052525),	D(125252),	BITS<15:12>=(12) 74?? K4/90 K3/10 ... K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
108)	4665	TEST121A1	6832	7405	ALU=A-XOR-B;	A(000000),	B(052525),	D(052525),	BITS<15:12>=(05) 74?? K4/90 K3/10 ... K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
109)	4667	TEST120A1	6685	7412	ALU=A-AND-B;	A(125252),	B(177777),	D(125252),	BITS<15:12>=(12) 74?? K4/90 K3/10 ... K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
110)	4671	TEST122A1	7148	7434	ALU=A-IOR-B;	A(000000),	B(000000),	D(000000)	7400 K4/99 ... FI; K4=E1, E7-E8, E11, E13, E22, E92, E96, E100-E102, E109-E112 7402 K4/99 ... FI; K4=E7, E10, E21-E22, E101 7417 K4/99 ... FI; K4=E110 7420 K4/99 ... FI; K4=E22, E102, E112 7421 K4/99 ... FI; K4=E22, E112 7422 K4/99 ... FI; K4=E112 7424 K4/99 ... FI; K4=E21, E112 7426 K4/99 ... FI; K4=E102 7430 K4/99 ... FI; K4=E102 7433 K4/99 ... FI; K4=E102 74?? K4/90 K3/10 ... K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
111)	4673	TEST130A1	7283	7434	ALU=A-PLUS-B-PLUS-0;	A(052525),	B(052525),	D(125252)	4777 K2/99 ... FI; K2=E79 7400 K4/60 K3/40 ... FI; K4=E1, E6, E49, E64, E88-E89, E98, E101, E108, E110; K3=E35, E42, E82, E92 7402 K4/70 K3/30 ... FI; K4=E21, E49, E98, E100-E102, E109-E112; K3=E41, E43, E52-E53, E84 7420 K4/99 ... FI; K4=E49, E102, E112 7421 K4/85 K3/15 ... FI; K4=E49, E71; K3=E71 7423 K4/99 ... FI; K4=E112 7427 K4/99 ... FI; K4=E112 7431 K4/99 ... FI; K4=E102 74?? K4/90 K3/10 ... K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
112)	4674	TEST121A4	6885	7403	BUTR(D(C)BA00),	D(C)=ALU00="1"	D=(052525)		7401 K4/99 ... FI; K4=E56, E63, E114
113)	4675	TEST121A3	6871	7432	ALU=A-XOR-B;	A(000000),	B(052525),	D(052525),	BITS<05:00>=(25) 7420 K4/99 ... FI; K4=E10 74?? K4/90 K3/10 ... K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
114)	4676	TEST121A2	6854	7405	ALU=A-XOR-B;	A(000000),	B(052525),	D(052525),	BITS<11:06>=(25) 74?? K4/90 K3/10 ... K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE

Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/%% #2/%% #3/%%	Test summary - Print reference - Chip information
115)	4677	TEST117C2	6655	7417	BUTR(X#D(C)#XX), D(C)=CINMUX=D(C)="1", ALU=A-AND-NOT-B 4756 K3/50 K2/50 ..../.. FI; K3=E12,E15; K2=E80-E81 7413 K4/90 K2/10 ..../.. FI; K4=E2-E3,E56,E63,E104; K2=E1		
116)	4701	TEST350	8337	NONE	(NUA SEQUENCING LOGIC ERROR) ???? K1/99 ..../.. ..../.. INTERNAL DCS ERROR		
117)	4705	TEST352A	8670	7434	BSPLO ADDRESSING, USING BSP/RIF ADDRESS=(02) 7403 K4/99 ..../.. ..../.. FI; K4=E13 7417 K4/99 ..../.. ..../.. FI; K4=E13 74?? K4/99 ..../.. ..../.. K407=BSP-ADDRS-MUX/REG(RIF)		
118)	4707	TEST121C1	6987	7412	ALU=A-XOR-B; A(000000), B(125252), D(125252), BITS<15:12>=(12) 74?? K4/90 K3/10 ..../.. K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE		
119)	4710	TEST763D	18148	7407	BUTR(BG-SERVICE) NEGATED; AFTER INTR DETECTED/SERVICED 4747 K5/99 ..../.. ..../.. FI; K5=E81-E82 7403 K7/99 ..../.. ..../.. K703=BR-REQUEST-INTR,K704=BR-GRANT-INTR		
120)	4711	TEST762F	18031	7403	BUTR(SERVICE) ASSERTED; DL11-W BR6 INTR PRESENT 7402 K2/99 ..../.. ..../.. FI; K2=E100		
121)	4712	TEST351D	8634	7434	ASPHI ADDRESSING, USING ASP/RIF ADDRESS=(05) 4747 K2/99 ..../.. ..../.. FI; K2=E1 74?? K4/99 ..../.. ..../.. K406=ASP-ADDRS-MUX/REG(RIF)		
122)	4713	TEST352B	8692	7434	BSPLO ADDRESSING, USING BSP/RIF ADDRESS=(03) 74?? K4/99 ..../.. ..../.. K407=BSP-ADDRS-MUX/REG(RIF)		
123)	4716	TEST121C4	7040	7417	BUTR(X#D(C)#XX), D(C)=ALU07="1", D=(125252) 7413 K4/99 ..../.. ..../.. FI; K4=E56,E63		
124)	4717	TEST121C3	7025	7425	ALU=A-XOR-B; A(000000), B(125252), D(125252), BITS<05:00>=(52) 7420 K4/99 ..../.. ..../.. FI; K4=E88 74?? K4/90 K3/10 ..../.. K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE		
125)	4720	TEST763B	18108	7402	VECTOR LOADED FROM DL11-W LINE CLOCK; CHECK=(000100) 7400 K7/85 K6/15 ..../.. FI; K7=E13,E17,E20-E21,E28; K6=E34		
126)	4721	TEST763C	18128	7401	BUTR(VECTOR-LOAD) NEGATED; AFTER INTR DETECTED/SERVICED 7403 K7/80 K6/20 ..../.. FI; K7=E10,E21,E94; K6=E26		
127)	4722	TEST351B	8591	7434	ASPL0 ADDRESSING, USING ASP/RIF ADDRESS=(03) 74?? K4/99 ..../.. ..../.. K406=ASP-ADDRS-MUX/REG(RIF)		

-----  
Module codes: K1/DCS K2/JWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/% #2/% #3/%			Test summary - Print reference - Chip information
128)	4723	TEST351C	8612	7434	ASPHI	ADDRESSING, USING ASP/RIF ADDRESS=(04)			
					4777	K2/99	../. .	../. .	FI; K2=E1
					7271	K2/99	../. .	../. .	FI; K2=E28
					7400	K4/99	../. .	../. .	FI; K4=E14
					74??	K4/99	../. .	../. .	K406=ASP-ADDRS-MUX/REG(RIF)
129)	4724	TEST352C	8713	7434	BSPHI	ADDRESSING, USING BSP/RIF ADDRESS=(04)			
					7400	K4/99	../. .	../. .	FI; K4=E12
					74??	K4/99	../. .	../. .	K407=BSP-ADDRS-MUX/REG(RIF)
130)	4725	TEST352D	8735	7434	BSPHI	ADDRESSING, USING BSP/RIF ADDRESS=(05)			
					74??	K4/99	../. .	../. .	K407=BSP-ADDRS-MUX/REG(RIF)
131)	4726	TEST121D2	7083	7405	ALU=A-XOR-B;	A(177777), B(125252), D(052525), BITS<11:06>=(25)			
					74??	K4/90	K3/10	../. .	K4(4=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
132)	4727	TEST121D1	7061	7405	ALU=A-XOR-B;	A(177777), B(125252), D(052525), BITS<15:12>=(05)			
					74??	K4/90	K3/10	../. .	K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
133)	4730	TEST762E	18006	7402	DL11-W BR6	INTR PENDING, LAST BUS DAT0; SERVICE=(043740)			
					7400	K3/99	../. .	../. .	FI; K3=E50
					7401	K2/99	../. .	../. .	FI; K2=E100
					740?	K7/99	../. .	../. .	K708=STATUS-MUX
134)	4732	TEST351A	8569	7434	ASPL0	ADDRESSING, USING ASP/RIF ADDRESS=(02)			
					7400	K4/99	../. .	../. .	FI; K4=E9, E14
					7402	K7/99	../. .	../. .	FI; K7=E52
					7403	K4/65	K2/35	../. .	FI; K4=E15, E86; K2=E52
					7417	K4/99	../. .	../. .	FI; K4=E4, E15
					74??	K4/99	../. .	../. .	K406=ASP-ADDRS-MUX/REG(RIF)
135)	4734	TEST122A3	7188	7402	BUT(D<14:00>=0#D15),	D=(000000), TARGET="10"			
					7400	K2/95	K3/5	../. .	FI; K2=E40, E56, E88-E89, E94-E96, E104-E105; K3=E56, E66
					7403	K3/99	../. .	../. .	FI; K3=E54, E95
					7406	K3/99	../. .	../. .	FI; K3=E74
136)	4735	TEST122A4	7202	7401	BUT(D<14:00>=0#D15),	D=(125252), TARGET="01"			
					7400	K3/99	../. .	../. .	FI; K3=E54-E55, E63, E95
					7403	K2/85	K3/15	../. .	FI; K2=E40, E56, E90; K3=E56
137)	4736	TEST121D4	7115	7413	BUTR(X#D(C)#XX),	D(C)=ALU07="0", D=(052525)			
					7417	K4/99	../. .	../. .	FI; K4=E56, E63

-----  
 Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/% #2/% #3/%			Test summary - Print reference - Chip information
138)	4737	TEST121D3	7100	7432	ALU=A-XOR-B; A(177777), B(125252), D(052525), BITS<05:00>=(25) 7434 K2/99 ..:/.. ..:/.. FI; K2=E2 7477 K4/90 K3/10 ..:/.. K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE				
139)	4740	TEST762C	17946	7403	BUTR(BG-SERVICE) ASSERTED; BR6 INTR PENDING, PROC PRI0=(5) 7407 K7/99 ..:/.. ..:/.. FI; K7=E3,E5,E11,E20,E25				
140)	4741	TEST762D	17980	7417	BUTR(BGSERVICE+FPSERVICE) ASSERTED; BR6 INTR PRESENT 7407 K3/80 K7/20 ..:/.. ..:/.. FI; K3=E50,E76; K7=E80				
141)	4742	TEST350C	8471	7434	ASPL0 ADDRESSING, USING ASP/DF ADDRESS FROM (7):(0) 7400 K4/80 K3/20 K5/5 K2/5 FI; K4=E1,E5,E7,E10,E15-E16,E23,E25,E31-E34, E40-E43,E99,E110; K3=E45-E47,E59; K5=E63; K2=E55 7402 K4/65 K3/35 K6/5 FI; K4=E1,E6,E15,E17,E22,E33,E86,E92,E99-E102, E109-E112; K3=E19,E24-E26,E45,E47; K6=E40 7403 K3/99 ..:/.. ..:/.. FI; K3=E59 7406 K4/60 K3/40 ..:/.. ..:/.. FI; K4=E6; K3=E46 7417 K4/99 ..:/.. ..:/.. FI; K4=E43 7420 K4/99 ..:/.. ..:/.. FI; K4=E25,E31-E32,E102,E112 7421 K4/99 ..:/.. ..:/.. FI; K4=E31 7422 K4/99 ..:/.. ..:/.. FI; K4=E31 7424 K4/99 ..:/.. ..:/.. FI; K4=E31 7426 K4/99 ..:/.. ..:/.. FI; K4=E32 7430 K4/99 ..:/.. ..:/.. FI; K4=E32				
142)	4743	TEST350D	8499	7434	ASPHI ADDRESSING, USING ASP/SF ADDRESS FROM (0):(7) 7400 K4/99 ..:/.. ..:/.. FI; K4=E6-E7,E14,E17,E42,E99-E100,E109 7402 K4/75 K3/25 ..:/.. ..:/.. FI; K4=E17,E34,E99; K3=E45 7417 K4/99 ..:/.. ..:/.. FI; K4=E15,E109 7420 K4/99 ..:/.. ..:/.. FI; K4=E40-E41,E74,E92,E99,E111 7421 K4/99 ..:/.. ..:/.. FI; K4=E111 7422 K4/99 ..:/.. ..:/.. FI; K4=E111 7424 K4/99 ..:/.. ..:/.. FI; K4=E111 7426 K4/99 ..:/.. ..:/.. FI; K4=E92 7433 K4/99 ..:/.. ..:/.. FI; K4=E99				
143)	4744	TEST740A	17556	7402	INSTR1*-PREFETCH, BC<0>=BC<0> 4747 K2/99 ..:/.. ..:/.. FI; K2=E24 7401 K2/99 ..:/.. ..:/.. FI; K2=E105 7403 K5/99 ..:/.. ..:/.. FI; K5=E97				
144)	4745	TEST761A	17726	7402	BUTR(SERVICE) NEGATED; AFTER CLEAR ALL SERVICE CONDITIONS, UNIBUS INIT 4747 K7/99 ..:/.. ..:/.. FI; K7=E57 7403 K7/99 ..:/.. ..:/.. K707/9=UNIBUS-UCON-INIT,K704=BG-GRANT-INTR				

-----  
Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS



ERROR	code	Symbolic label	Line number	ENUA	TNUA	->Module sequence->			Test summary - Print reference - Chip information
----	----	-----	-----	-----	-----	#1/%%	#2/%%	#3/%%	-----
145)	4750	TEST131B2	7474	7413	BUTR(X#D(C)XX), D(C)=COUT15="0": A(0)+B(1)+CI(0)=D(1)+CO(0)	7403	K4/90	K3/10	FI; K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
146)	4751	TEST762B	17924	7407	BUTR(BG-SERVICE) NEGATED; BR6 INTR PENDING, PROC PRIO=(6)	7403	K7/99	..../..	FI; K7=E11, E76
					7407	K7/99	..../..	..../..	K703=BR-REQUEST-INTR
147)	4752	TEST350A	8417	7434	BSPLO ADDRESSING, USING BSP/SF ADDRESS FROM (0):(7)	7400	K4/90	K2/10	FI; K4=E1, E5, E7-E9, E12, E14, E17, E21, E26, E35, E92, E99-E102, E109-E112; K2=E38, E55
					7402	K4/99	..../..	..../..	FI; K4=E22, E100
					7403	K4/99	..../..	..../..	FI; K4=E47
					7407	K4/99	..../..	..../..	FI; K4=E20
					7410	K4/99	..../..	..../..	FI; K4=E6, E13
					7413	K7/99	..../..	..../..	FI; K7=E25
					7417	K4/99	..../..	..../..	FI; K4=E4, E13, E15, E17, E22, E26, E109
					7420	K4/99	..../..	..../..	FI; K4=E92, E111
					7421	K4/99	..../..	..../..	FI; K4=E111
					7422	K4/99	..../..	..../..	FI; K4=E111
					7424	K4/99	..../..	..../..	FI; K4=E111
					7426	K4/99	..../..	..../..	FI; K4=E92
					7430	K4/99	..../..	..../..	FI; K4=E92
					7431	K4/99	..../..	..../..	FI; K4=E92, E99, E111
148)	4753	TEST350B	8444	7434	BSPHI ADDRESSING, USING BSP/DF ADDRESS FROM (7):(0)	7400	K4/80	K2/15	FI; K4=E6-E7, E11-E15, E17, E21-E22, E26, E55, E87, E96, E101, E110; K2=E32, E44, E50, E53; K7=E66; K5=E66
					7417	K4/85	K5/15	..../..	FI; K4=E13, E15, E22; K5=E66
					7420	K4/99	..../..	..../..	FI; K4=E102, E112
149)	4755	TEST731A	17321	7402	BUS DAT08*BYTE*000, BA=(000001); DBUF=D=(000000)	4415	K2/99	..../..	FI; K2=E30
					4747	K6/65	K2/35	..../..	FI; K6=E5; K2=E18
					????	K6/99	..../..	..../..	K605=UNIBUS-FUNCTION-DECODE
150)	4760	TEST720C	16813	7402	DATIB*-BYTE, 16. BIT PBA, -I/O PAGE(3); SERVICE=(100340)	7400	K6/99	..../..	FI; K6=E59
151)	4761	TEST763A	18062	4733	ALLOW-BG(1)H GIVEN TO BR6 INTR; BUTR (VECTOR-LOAD) ASSERTED	4731	K7/95	K2/5	FI; K7=E4-E5, E8, E10, E12-E13, E17-E18, E20-E22, E25, E29, E33, E35, E37, E41, E51, E59, E61, E76-E77, E79, E94, E101; K2=E70; K6=E56; K5=E53
					4756	K7/99	..../..	..../..	FI; K7=E6-E7

Module codes: K1/DCS K2/JWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7 STATUS

ERROR	code	Symbolic label	Line number	ENUA	TNUA	->Module sequence->			Test summary - Print reference - Chip information		
						#1/%%	#2/%%	#3/%%			
152)	4771	TEST761B	17765	7401	BUTR(VECTOR-LOAD) NEGATED; AFTER UNIBUS INIT	7403	K7/99	... ..	K707/9=UNIBUS-UCON-INIT, K704=BG-GRANT-INTR		
						7403	K3/99	... ..	FI; K3=E44		
153)	4773	TEST762A	17829	4343	(NUA SEQUENCING LOGIC ERROR)	4747	K6/75	K5/20	K7/5	K4/5	FI; K6=E11, E19, E25, E29, E33, E49-E50; K5=E2, E10, E33, E80, E82, E97; K7=E66; K4=E48
154)	4774	TEST761C	17786	7407	BUTR(BG-SERVICE) NEGATED; AFTER UNIBUS INIT	7403	K7/99	... ..	K707/9=UNIBUS-UCON-INIT, K704=BG-GRANT-INTR		
155)	4775	TEST010	2802	5245	BUTA(SUBR-B) -> BUTA(RETURN) SEQUENCE, RETURN="1010 1010 0101"=(5245)	4245	K2/65	K3/35	... ..	FI; K2=E35, E41, E61, E77, E85; K3=E4, E7, E15, E43	
						4376	K2/90	K3/15	... ..	FI; K2=E33, E35, E39-E40, E45-E46, E51, E54; K3=E34	
						4777	K2/99	... ..	... ..	FI; K2=E5, E13-E14, E41, E57, E63, E82, E85-E86, E91-E92, E97, E106, E110, E116, E118	
						5005	K2/99	... ..	... ..	FI; K2=E25	
						5045	K2/85	K4/15	... ..	FI; K2=E7, E13, E25, E31, E37, E43, E49, E52, E70, E97; K4=E12-E13	
						5105	K2/99	... ..	... ..	FI; K2=E13	
						5125	K2/99	... ..	... ..	FI; K2=E13	
						5205	K2/90	K4/15	... ..	FI; K2=E4, E7, E10, E13, E15, E22, E25, E28, E71, E97; K4=E1	
						5240	K2/99	... ..	... ..	FI; K2=E26	
						5241	K2/85	K4/15	... ..	FI; K2=E4, E8, E10, E14, E16, E22, E26, E28, E62, E91; K4=E20, E86	
						5242	K2/99	... ..	... ..	FI; K2=E14	
						5244	K2/70	K4/30	... ..	FI; K2=E8, E14, E26, E58, E64, E67, E71, E73, E79, E91; K4=E12, E14	
						5245	K2/85	K4/15	... ..	FI; K2=E25, E31, E35, E37, E43, E49, E52, E58, E62, E64, E67, E73, E77, E79, E91, E97; K4=E3, E70	
						5247	K2/99	... ..	... ..	FI; K2=E8, E14, E26	
						5252	K2/99	... ..	... ..	FI; K2=E14	
						5255	K2/99	... ..	... ..	FI; K2=E8, E14, E26	
						5256	K2/99	... ..	... ..	FI; K2=E8	
						5257	K2/99	... ..	... ..	FI; K2=E14	
						5265	K2/99	... ..	... ..	FI; K2=E7, E13, E25	
						5345	K2/99	... ..	... ..	FI; K2=E7, E13, E25	
						5365	K2/99	... ..	... ..	FI; K2=E7, E13	
						5376	K2/99	... ..	... ..	FI; K2=E40	
						7245	K2/99	... ..	... ..	FI; K2=E35, E41, E77	
						7247	K2/99	... ..	... ..	FI; K2=E40	
						7273	K3/55	K2/45	... ..	FI; K3=E4, E6-E7, E11, E15, E43; K2=E61, E77, E85	
						7401	K2/99	... ..	... ..	FI; K2=E41	
						7777	K2/99	... ..	... ..	FI; K2=E11	

Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7-STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/%% #2/%% #3/%%	Test summary - Print reference - Chip information
156)	5146	TEST005	2711	4474		NUA SEQUENCING PAGE (5) -> (4), UBF=(34) 4574 K3/50 K2/50 ../.. FI; K3=E58; K2=E7 5441 K3/60 K2/40 ../.. FI; K3=E10,E18,E20; K2=E30,E54	
157)	5463	TEST722A	16951	4777		INVALIDATE, ODD ADDR JAM, BA=(140001) 7341 K6/75 K4/15 K3/15 FI; K6=E71,E76,E81,E88,E96; K4=E51; K3=E11 ???? K6/99 ../.. K605=UNIBUS-FUNCTION-DECODE	
158)	5465	TEST720A	16730	4777		DATIB*-BYTE, ODD ADDR JAM, BA=(060001) 7341 K5/50 K3/50 ../.. FI; K5=E67,E94; K3=E34 ???? K6/99 ../.. K605=UNIBUS-FUNCTION-DECODE	
159)	5467	TEST713A	16594	4777		DATI-NOINT, ILLEGAL INTERNAL ADDR JAM, BA=(177776) ???? K6/99 ../.. K605=UNIBUS-FUNCTION-DECODE, INTERNAL-ADDR-DETECT	
160)	5471	TEST712A	16460	4777		DATI, INTERNAL ADDRESS JAM, BA=(177776) ???? K6/99 ../.. K605=UNIBUS-FUNCTION-DECODE, INTERNAL-ADDR-DETECT	
161)	5473	TEST711A	16328	4777		DATOB*BYTE, SSYN TIMEOUT JAM, BA=(160001) 7341 K6/70 K4/20 K7/5 K5/5 FI; K6=E4,E11,E25,E62,E64,E78-E79,E98-E99, E101-E102; K4=E18,E28; K7=E60; K5=E55 ???? K6/99 ../.. K605=UNIBUS-FUNCTION-DECODE	
162)	5475	TEST710A	16157	4777		DATO, ODD ADDRESS ERROR JAM, BA=(160001) 4747 K6/99 FI; K6=E60 7341 K6/75 K7/15 K4/10 K5/5 K3/5 FI; K6=E2,E4,E10,E38,E40,E42-E43, E46-E47,E54-E56,E60,E64,E69,E71-E72,E76,E79,E81, E89,E95,E98-E99,E103,E108; K7=E22,E25,E39,E43, E45-E46,E65; K4=E9,E18,E26,E30,E71; K5=E44,E64; K3=E52 7401 K5/99 ../.. FI; K5=E97 7445 K6/99 ../.. FI; K6=E43 ???? K6/99 ../.. K605=UNIBUS-FUNCTION-DECODE	
163)	5477	TEST702A	16040	7402		LOAD BA<17:16>="01", 18. BIT MODE, READ THRU STATUS-MUX(SERVICE)<9:8> 4747 K2/99 FI; K2=E34 7400 K6/40 K5/25 K4/20 K7/10 K2/5 FI; K6=E11,E51-E54,E59; K5=E4,E13, E15,E26,E54,E84,E97; K4=E20,E65; K7=E48,E56; K2=E61	
164)	5500	TEST374D2	10178	7434		A/B SP WRITE FCN WR(B,HI,A-ADDR) AND WR(B,HI,B-ADDR) 74?? K4/99 ../.. K405=SP-REWRITE-CNTL,K406/7=A/B-SPADS	
165)	5502	TEST374E2	10251	7434		A/B SP WRITE FCN WR(AB,LO,A-ADDR) AND WR(AB,LO,B-ADDR) [Continued] 7400 K2/99 ../.. FI; K2=E117	

-----  
Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/%% #2/%% #3/%%			Test summary - Print reference - Chip information
	5502	TEST374E2	[Continued]			A/B SP WRITE FCN WR(AB,LO,A-ADDR) AND WR(AB,LO,B-ADDR)			
					7420	K4/99	..../..	..../..	FI; K4=E10
					74??	K4/99	..../..	..../..	K405=SP-REWRITE-CNTL,K406/7=A/B-SPADS
166)	5504	TEST374F2	10324	7434		A/B SP WRITE FCN WR(AB,HI,A-ADDR) AND WR(AB,HI,B-ADDR)			
					74??	K4/99	..../..	..../..	K405=SP-REWRITE-CNTL,K406/7=A/B-SPADS
167)	5505	TEST712B	16512	7402		DATI, INTERNAL ADDR; JAM=(001000)			
					7400	K6/70	K7/15	K3/15	K5/5 FI; K6=E46,E49-E51,E81,E99; K7=E19,E58,E61, E81; K3=E48,E69; K5=E13,E72
					7401	K6/99	..../..	..../..	FI; K6=E76
					???	K6/99	..../..	..../..	K605=INTERNAL-ADDR-DETECT
168)	5507	TEST410	10666	NONE		(NUA SEQUENCING LOGIC/DCS ERROR)			
					???	K1/99	..../..	..../..	INTERNAL DCS ERROR
169)	5511	TEST620A	15262	7406		BUTM(INIT-JAM) NEGATED="0" AFTER CLR-JAM-ERRORS UCON			
					7407	K7/90	K2/10	..../..	FI; K7=E18,E41,E51; K2=E112
170)	5512	TEST701D	16011	7402		BUTR(DIC)BA00), BA<00>="0"			
					7403	K3/99	..../..	..../..	FI; K3=E64
171)	5513	TEST376A	10595	5440		BUTA(R-IOR-1) DOES NOT CAUSE A BRANCH			
					5441	K3/99	..../..	..../..	FI; K3=E64
172)	5517	TEST375A	10512	7432		DAD/3 CAUSES BYTE-WRITE(LO) TO ASPLO; DAD/1 OR DAD/2 NOT			
					7400	K3/80	K4/20	..../..	FI; K3=E17,E20,E36,E38,E46; K4=E5,E99
					7402	K3/85	K4/15	..../..	FI; K3=E27,E36,E38,E46,E49,E59-E60; K4=E5,E33
					7433	K4/99	..../..	..../..	FI; K4=E25
					7434	K3/99	..../..	..../..	FI; K3=E59
173)	5521	TEST102A	5225	7432		CSP ADDRESS, LOC(02); EMIT/CSP/ALU-B/D/DBUF/IR WITH DATA=(000125)			
					7400	K4/60	K3/40	..../..	FI; K4=E47,E55,E88-E89,E95-E96,E98,E105,E108; K3=E5,E8-E10,E14,E32,E35,E42,E92
					7402	K4/70	K3/30	..../..	FI; K4=E55,E95,E98; K3=E12-E13,E32,E35,E42
					7403	K4/99	..../..	..../..	FI; K4=E96
					7410	K3/75	K4/25	..../..	FI; K3=E1-E3,E5,E13-E14,E42; K4=E1,E30
					7417	K4/99	..../..	..../..	FI; K4=E105,E108
					7420	K4/70	K6/10	K2/10	K3/5 K7/5 K5/5 FI; K4=E1,E4,E7-E8,E11-E12, E31-E32,E35,E40-E41,E52,E56,E65-E66,E72,E74,E79, E88-E92,E96,E100-E102,E105-E106,E109-E112; K6=E65,E67,E73,E75,E108; K2=E83,E94,E98,E100, E102-E104,E107-E108,E111,E113-E114; K3=E72; K7=E85-E86; K5=E63,E66
					7421	K4/99	..../..	..../..	FI; K4=E87
					7427	K6/55	K4/30	K7/15	FI; K6=E73; K4=E89; K7=E37
					7433	K4/90	K3/15	..../..	FI; K4=E89,E105; K3=E8
			[Continued]		7434	K4/75	K3/10	K6/10	K7/5 K2/5 FI; K4=E1-E5,E7,E9-E11,E18,E20-E26,

ERROR ###) code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/%% #2/%% #3/%%			Test summary - Print reference - Chip information
5521	TEST102A	[Continued]			CSP ADDRESS, LOC(02);	EMIT/CSP/ALU-B/D/DBUF/IR WITH DATA=(000125)		
				7434	K4/75	K3/10	K6/10	E28-E30, E38, E47, E51, E55-E56, E59, E64, E66, E70-E71, E74, E87-E89, E96, E98-E99, E108; K3=E2-E2, E12, E35, E42-E43, E53, E59, E82, E84; K6=E31, E34, E42, E82, E103; K7=E76; K2=E102
174)	5523	TEST701B	15962	7403	BUTR(D(C)BA00), BA<00>="1"			
				7402	K3/99	...	...	K308=MICROBRANCH
175)	5524	TEST374C2	10105	7434	A/B SP WRITE FCN WR(B,LO,A-ADDR) AND WR(B,LO,B-ADDR)			
				7400	K3/99	...	...	FI; K3=E45
				7402	K4/99	...	...	FI; K4=E10, E17, E21-E22
				7420	K3/99	...	...	FI; K3=E93
				7421	K4/99	...	...	FI; K4=E21-E22
				74??	K4/99	...	...	K405=SP-REWRITE-CNTL, K406/7=A/B-SPADS
176)	5525	TEST004	2692	5146	NVA SEQUENCING, NO 'BUT'			
				5147	K3/99	...	...	FI; K3=E38
				5156	K3/99	...	...	FI; K3=E37
				5346	K3/99	...	...	FI; K3=E23
177)	5531	TEST103A	5370	7412	CSP ADDRESS, LOC(13);	EMIT/CSP/ALU-B/D/DBUF/IR WITH DATA=(125200)		
				7400	K4/99	...	...	FI; K4=E59, E66, E74, E92, E102, E109-E112
178)	5533	TEST610B1	14869	7413	PS(CC)=(13); IR=(105300),	CC-ROM-ADDR(132)		
				7401	K3/99	...	...	FI; K3=E55
				7403	K3/99	...	...	FI; K3=E63, E72
				7411	K3/90	K4/10	...	FI; K3=E55, E62, E65, E85; K4=E72
				7417	K3/60	K2/40	...	FI; K3=E55, E72; K2=E105
179)	5535	TEST610C1	14987	7407	PS(CC)=(07); IR=(072000),	CC-ROM-ADDR(437)		
				7405	K3/99	...	...	FI; K3=E33
				7406	K3/99	...	...	FI; K3=E62
				7415	K2/99	...	...	FI; K2=E8
180)	5541	TEST105A	5592	7432	SR LOAD/READ, SR=(000125),	RES=SR/LOAD; ALU-A/D/DBUF/IR PATH		
				7400	K4/85	K3/10	K2/5	FI; K4=E1, E3, E11, E16, E18, E20, E27, E30, E35-E36, E45-E47, E52, E80; K3=E5, E10, E14, E51, E66; K2=E89
				7402	K4/85	K3/10	K2/5	FI; K4=E30, E35, E45-E47; K3=E8, E14; K2=E101
				7403	K4/50	K3/50	...	FI; K4=E80; K3=E9
				7406	K3/99	...	...	FI; K3=E9
				7410	K4/99	...	...	FI; K4=E36, E52
				7412	K3/99	...	...	FI; K3=E8
				7417	K4/99	...	...	FI; K4=E27, E36, E47, E52
				7420	K4/90	K3/10	...	FI; K4=E1, E3, E7, E11, E16, E20, E26, E31-E32, E34-E35, E37, E40-E42, E44, E47-E48, E52-E54, E57, E60-E62, E67-E68, E75-E76, E80; K3=E41, E43, E51, E67, E71, E75, E82

[Continued]

-----  
Module codes: K1/DCS K2/WORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS  
-----

ERROR	code	Symbolic label	Line number	ENUA	TNUA	->Module sequence->			Test summary - Print reference - Chip information
						#1/%%	#2/%%	#3/%%	
	5541	TEST105A	[Continued]		SR LOAD/READ, SR=(000125),				RES=SR/LOAD; ALU-A/D/DBUF/IR PATH
					7427 K4/99	.../..	.../..		FI; K4=E1, E7, E16, E33, E35
					7430 K4/99	.../..	.../..		FI; K4=E1
					7431 K4/99	.../..	.../..		FI; K4=E36, E45
					7433 K4/99	.../..	.../..		FI; K4=E35, E37, E47, E66
					7434 K4/99	K3/5	.../..		FI; K4=E1, E4, E7, E11, E16, E30-E32, E35-E36, E39, E43, E45, E47-E48, E52-E53, E55, E57, E65, E80; K3=E102
181)	5545	TEST61001	15101	7410	PS[CC]=(10); IR=(072000),				CC-ROM-ADDR(216)
					7412 K3/99	.../..	.../..		FI; K3=E33, E62
182)	5547	TEST104A	5516	7425	CSP BASCON ADDR, LOC(16);				EMIT/CSP/ALU-B/D/DBUF/IR WITH DATA=(000152)
					7400 K4/99	.../..	.../..		FI; K4=E38, E96
					7417 K4/99	.../..	.../..		FI; K4=E38
					7420 K4/99	.../..	.../..		FI; K4=E74, E111-E112
					7434 K3/60	K4/35	K2/5		FI; K3=E1-E3, E12-E13; K4=E1, E30, E38; K2=E70
183)	5551	TEST050A	5077	7405	BUT(INSTRS), IR=(172500);				E88(725)=(05)
					74?? K3/99	.../..	.../..		K305=INSTRS-DECODE
184)	5553	TEST101A	5161	7434	ALU=ZERO, D/DBUF/IR PATH;				BUT(INSTRS) FOR IR=ZERO
					7400 K4/65	K6/20	K3/10	K7/5	FI; K4=E33-E34, E42-E43, E51, E59, E82, E95, E100-E101, E109-E110; K6=E73, E82, E91, E96; K3=E1, E43, E51, E71, E82; K7=E52, E54; K2=E70
					7402 K4/80	K6/15	K3/5		FI; K4=E33-E34, E51, E82, E95, E100-E101; K6=E91; K3=E13
					7410 K3/99	.../..	.../..		FI; K3=E43, E82, E102
					7411 K6/99	.../..	.../..		FI; K6=E65, E82
					7412 K7/65	K2/30	K3/5		FI; K7=E28, E41, E52, E54, E70; K2=E61, E83; K3=E43
					7417 K6/99	.../..	.../..		FI; K6=E82
					7420 K4/75	K6/15	K7/5	K5/5	FI; K4=E31-E32, E38, E40-E41, E47, E55, E66, E74, E79, E92, E102, E111-E112; K6=E65, E73; K7=E41, E48, E65, E70; K5=E14, E54; K3=E82, E84; K2=E34, E85
					7421 K4/90	K6/15	.../..		FI; K4=E31, E40, E47, E72, E74, E111-E112; K6=E65
					7422 K7/75	K4/20	K6/5		FI; K7=E6-E7, E15, E23, E31, E45, E48, E65-E67, E69, E74, E78, E91-E92; K4=E31, E40, E72, E74, E111-E112; K6=E43, E65
					7424 K4/80	K6/20	.../..		FI; K4=E31, E40, E74, E79, E111-E112; K6=E65
					7426 K6/50	K3/50	.../..		FI; K6=E73; K3=E51, E82
					7427 K4/99	.../..	.../..		FI; K4=E74
					7431 K3/99	.../..	.../..		FI; K3=E43
					74?? K4/90	K3/10	.../..		K404=ALU/CARRY-LOOKAHEAD; K313=ALU-FCN-DECODE
185)	5554	TEST1020	5330	7425	CSP ADDRESS, LOC(01);				EMIT/CSP/ALU-B/D/DBUF/IR WITH DATA=(000152)
					7400 K4/99	.../..	.../..		FI; K4=E4
					7402 K5/99	.../..	.../..		FI; K5=E44
			[Continued]		7420 K4/60	K6/10	K3/5	K2/5	FI; K4=E35, E52, E65-E66, E72, K5/5 K7/5

Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

ERROR ###) code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/%% #2/%% #3/%%			Test summary - Print reference - Chip information
5554	TEST1020	[Continued]						CSP ADDRESS, LOC(01); EMIT/CSP/ALU-B/D/DBUF/IR WITH DATA=(000152) 7420 K4/60 K6/10 K3/5 E74, E79, E88-E92, E102, E106, E111-E112; K6=E65, E67, E73, E75; K3=E72; K2=E94, E104; K5=E66, E106; K7=E85-E86 7434 K4/99 ../.. ../.. FI; K4=E38, E70, E88-E89, E98, E108
186) 5561	TEST047A	4988	7412					BUT(INSTRS), IR=(122566); E88(325)=(12) 7400 K3/99 ../.. ../.. FI; K3=E88 7405 K3/99 ../.. ../.. FI; K3=E88 7413 K3/99 ../.. ../.. FI; K3=E88 7416 K3/99 ../.. ../.. FI; K3=E88
187) 5563	TEST701A	15925	7402					LOAD BA<15:00>=(052525), READ THRU BA/KT-ALU/PBA/STATUS-MUX(PBA) 7400 K5/55 K6/15 K7/15 K4/15 K3/5 FI; K5=E3, E7, E9-E11, E13, E25, E27, E30, E32, E37-E39, E43, E45-E47, E50, E58-E60, E65-E67, E69-E70, E72, E79-E82, E93-E94; K6=E19, E25, E29, E33, E36, E38, E49, E54, E59, E70, E76, E90, E96, E104, E108; K7=E48, E56, E64-E65, E73, E80-E82, E88; K4=E29, E39, E48, E57, E65, E87; K3=E9, E14, E33, E64 7401 K4/45 K3/35 K6/20 FI; K4=E57, E65, E87; K3=E5, E8, E10, E33; K6=E31 7403 K7/50 K4/50 ../.. FI; K7=E80; K4=E39
188) 5564	TEST102C	5303	7412					CSP ADDRESS, LOC(04); EMIT/CSP/ALU-B/D/DBUF/IR WITH DATA=(125200) 7400 K4/65 K6/15 K2/10 K5/5 K3/5 FI; K4=E35-E36, E45, E51, E56, E59, E66, E72, E79-E80, E82, E89, E91-E92, E95, E98, E100-E102, E105, E108-E110; K6=E73, E75, E82, E84, E91, E93; K2=E95-E96, E104; K5=E53, E63; K3=E56, E95 7403 K4/75 K6/25 ../.. FI; K4=E45, E51, E64; K6=E91 7405 K4/99 ../.. ../.. FI; K4=E98 7420 K3/99 ../.. ../.. FI; K3=E17, E20, E27, E36 7432 K3/99 ../.. ../.. FI; K3=E32 7434 K4/75 K2/25 ../.. FI; K4=E29, E38, E70, E88-E89, E98, E108; K2=E5, E11, E17, E23, E29, E84
189) 5565	TEST624B	15817	7434					CLOCKING D REGISTER PROPOGATED THRU MICROBREAK JAM (P2-T) 74?? K4/99 ../.. ../.. K403/4=U-CLKD-CNTRL
190) 5571	TEST624A	15724	4777					MICROBREAK JAM AT MICROADDRESS=(6255); FLAG<8> CLEARED IN JAM WORD ???? K3/99 ../.. ../.. K311=MICROBREAK-LOGIC
191) 5572	TEST103B	5423	7432					CSP ADDRESS, LOC(15); EMIT/CSP/ALU-B/D/DBUF/IR WITH DATA=(000125) 7411 K2/80 K3/20 ../.. FI; K2=E2, E8-E9, E21, E27, E60, E66, E69, E75, E81; K3=E21-E22, E76 7415 K3/99 ../.. ../.. FI; K3=E21 7420 K4/99 ../.. ../.. FI; K4=E74, E111-E112
192) 5573	TEST623	15680 [Continued]	7402					BUTA(CUA-TRACK) RESETS CUA TO TRACKING NUA VALUE 4747 K3/99 ../.. ../.. FI; K3=E91

-----  
Module codes: K1/DCS K2/UMWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/% #2/% #3/%	Test summary - Print reference - Chip information
	5573	TEST623	[Continued]		7400	K2/99 .../..	BUTA(CUA-TRACK) RESETS CUA TO TRACKING NUA VALUE FI; K2=E3, E6, E9, E18, E24-E25, E30, E33, E39, E45, E51, E54, E60, E66, E75, E81, E86-E87, E113
193)	5574	TEST1028	5277	7405	7400	K4/65 K6/30 K2/5 K5/5	CSP ADDRESS, LOC(10); EMIT/CSP/ALU-B/D/DBUF/IR WITH DATA=(152500) FI; K4=E36, E38, E42-E43, E51, E59, E82, E89, E98, E105, E108-E110; K6=E82, E84, E91; K2=E96; K5=E63
					7402	K4/65 K6/15 K5/10 K2/10 K3/5	FI; K4=E45, E51, E64, E78, E95, E98, E100-E101; K6=E91, E93; K5=E53; K2=E95, K3=E95
					7403	K2/99 .../..	FI; K2=E34
					7412	K4/99 .../..	FI; K4=E89, E92, E102, E106
					7434	K4/90 K7/5 K3/5 K2/5	FI; K4=E4, E38, E70, E88-E89, E98-E99, E108; K7=E21; K3=E102; K2=E84
194)	5575	TEST622C	15657	7402	740?	K7/99 .../..	CLR-JAM-ERRORS RESETS STATUS-MUX(JAM)=(001000) AFTER UBREA <del>m</del> JAMUPP K705=UBRX-JAM-FLAGS, K707=JAM-CLEAR, K708=STATUS-MUX
195)	5576	TEST103C	5449	7425	7402	K4/99 .../..	CSP ADDRESS, LOC(16); EMIT/CSP/ALU-B/D/DBUF/IR WITH DATA=(000152) FI; K4=E100-E101
					7420	K3/99 .../..	FI; K3=E93
196)	5577	TEST622B	15634	7401	7403	K7/99 .../..	BUTR(OTHER-JAM) NEGATED="0" AFTER CLR-JAM-ERRORS FI; K7=E51-E52, E57
197)	5600	TEST105B1	5666	7412	7410	K4/99 .../..	BUT(SR3-0) SR=(000152); SR(3:0)="1010" FI; K4=E44
					7413	K3/99 .../..	FI; K3=E67
					7416	K3/99 .../..	FI; K3=E75
198)	5601	TEST622A	15605	7403	7401	K7/99 .../..	UCONS 'START-DELAY' & 'CLR-NPR-TIMEOUT' DONT EFFECT 'CLR-JAM-ERRORS' FI; K7=E36
199)	5602	TEST610C2	15038	7401	7403	K3/99 .../..	PS(CC)=(01); IR=(072000), CC-ROM-ADDR(037) FI; K3=E62
					7405	K3/99 .../..	FI; K3=E55, E66
200)	5603	TEST621F	15521	7402	7400	K2/99 .../..	PROC-MUX(CUA-PORT)=(055226); CUA-LOCKED, EXFLAGS SET, PREFETCH*JAM CLEAR FI; K2=E46, E55, E90, E100, E105
201)	5604	TEST374A2	9959	7434	7400	K4/80 K2/20 .../..	A/B SP WRITE FCN WR(A, LO, A-ADDR) AND WR(A, LO, B-ADDR) FI; K4=E11, E17, E23, E47; K2=E34
					7402	K4/75 K3/25 .../..	FI; K4=E17, E26; K3=E93
					7420	K4/70 K2/30 .../..	FI; K4=E10, E17, E23-E24, E26; K2=E32, E38, E44, E50, E53
					74??	K4/99 .../..	K405=SP-REWRITE-CNTL, K406/7=A/B-SPADS
202)	5605	TEST621E	15495	7402	7400	K2/75 K7/25 .../..	STATUS-MUX(JAM-PORT)=(001001) AFTER MICROBREAK JAM FI; K2=E56, E76, E83, E100; K7=E56, E73

Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS



###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/% #2/% #3/%	Test summary - Print reference - Chip information
203)	5606	TEST374B2	10032	7434	A/B SP WRITE FCN WR(A,HI,A-ADDR) AND WR(A,HI,B-ADDR) 74?? K4/99 .....	K405=SP-REWRITE-CNTL,K406/7=A/B-SPADS	
204)	5607	TEST6210	15476	7403	BUTR(OTHER-JAM) ASSERTED="1" AFTER MICROBREAK JAM 7401 K7/70 K3/30 .....	FI; K7=E32,E38,E57,E73; K3=E44	
205)	5610	TEST105A1	5619	7405	BUT(SR3=0), SR=(000125); SR(3:0)="0101" 7407 K3/95 .....	FI; K3=E77 7415 K3/99 .....	
206)	5611	TEST621C	15458	7406	BUTH(INIT-JAM) STAYS NEGATED AFTER MICROBREAK JAM 7407 K2/99 .....	FI; K2=E46	
207)	5612	TEST610B2	14919	7406	PS(CC)=(06); IR=(005300), CC-ROM-ADDR(253) 7402 K3/99 .....	FI; K3=E55,E63,E66 7404 K3/55 K4/45 .....	
208)	5613	TEST621B	15440	7402	CLOCKING D-REGISTER PROPAGATED THRU MICROBREAK JAM (P3-T) 74?? K4/99 .....	K403/4=U-CLKD-CNTRL	
209)	5614	TEST376A1	10619	7434	SP/SF-ADDRESS: FLTPT/BIT02="0"; R-IOR-1/BIT00="1" FORCED 7400 K3/40 K2/40 K4/25 .....	FI; K3=E50,E57,E94,E97; K2=E55,E108,E112; K4=E4,E9, E34,E101 7410 K2/65 K3/35 .....	
210)	5615	ERROR621A	15426	0000	MICROBREAK JAMUPP AT (5522) ATTEMPTED; DID NOT OCCUR 5615 K3/75 K2/10 K5/10 K6/5 K7/5 .....	FI; K3=E32,E41,E52,E61,E81,E91,E101, E111-E112; K2=E35,E41,E70,E77; K5=E60-E61,E68, E77; K6=E39-E40; K7=E48,E57	
211)	5616	TEST713B	16640	7402	DATI-NOINT, ILLEGAL INTERNAL ADDR; JAM=(001040) 7400 K6/70 K7/15 K5/15 .....	FI; K6=E46,E51; K7=E74,E88; K5=E38,E72	
212)	5617	TEST621A	15344	4777	MICROBREAK JAM AT MICROADDRESS=(5522); ACTIVE-BUT, WR-CSP LATCHES CLEARED 4747 K3/99 .....	FI; K3=E91,E111	
213)	5620	TEST105A	5743	7415	SR LOAD/READ, SR=(152500); SR(15:12)="1101" 7405 K5/99 .....	FI; K5=E60 7411 K4/75 K6/10 K5/5 K2/5 .....	

Module codes: K1/DCS K2/UMWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/% #2/% #3/%			Test summary - Print reference - Chip information
214)	5621	TEST620C	15309	7402					STATUS-MUX(JAM-PORT)=(001000) AFTER CLR-JAM-ERRORS UCON 7400 K7/95 K2/5 K6/5 FI; MOS BATTERY BACKUP MUST BE GOOD !! 7400 K7/95 K2/5 K6/5 FI; K7=E48-E49, E52, E56-E58, E64-E66, E73, E80-E82, E88; K2=E37, E64, E66, E70; K6=E51 7401 K7/85 K2/15 ..../.. FI; K7=E28, E37; K2=E100 7403 K7/99 ..../.. ..../.. FI; K7=E80
215)	5622	TEST610A2	14906	7406					PS(CC)=(06); IR=(105200), CC-ROM-ADDR(145) 7404 K3/90 K4/10 ..../.. FI; K3=E55, E62, E65, E72, E85; K4=E72 7407 K3/99 ..../.. ..../.. FI; K3=E62, E72
216)	5623	TEST620B	15286	7401					BUTR(OTHER-JAM) NEGATED="0" AFTER CLR-JAM-ERRORS UCON 7403 K7/80 K6/20 K3/5 FI; K7=E18, E26, E32, E38, E49, E57-E58, E74; K6=E2-E3, E62, E64, E105; K3=E44
217)	5624	TEST61002	15145	7410					PS(CC)=(10); IR=(072000), CC-ROM-ADDR(116) 7414 K3/99 ..../.. ..../.. FI; K3=E63, E66
218)	5625	TEST375B	10563	7432					DAD/3 CAUSES BYTE-WRITE(LO) TO BSPL0; DAD/1 OR DAD/2 NOT 7400 K3/99 ..../.. ..../.. FI; K3=E46 7402 K4/99 ..../.. ..../.. FI; K4=E99-E100
219)	5627	TEST610A1	14744	7405					PS(CC)=(05); IR=(105200), CC-ROM-ADDR(665) 7400 K3/99 ..../.. ..../.. FI; K3=E57 7401 K3/70 K2/30 ..../.. FI; K3=E55, E72; K2=E105 7404 K3/99 ..../.. ..../.. FI; K3=E62, E72 7406 K3/99 ..../.. ..../.. FI; K3=E62, E65, E85 7407 K3/80 K4/20 ..../.. FI; K3=E34, E55, E62, E65, E72; K4=E72 7411 K3/99 ..../.. ..../.. FI; K3=E34, E55, E65 7412 K3/99 ..../.. ..../.. FI; K3=E5, E8-E10, E14, E36, E72 7415 K3/99 ..../.. ..../.. FI; K3=E55, E62-E63, E72
220)	5631	TEST046A	4920	7426					BUT(INSTR5), IR=(000200); E78(200)=(26) 6000 K2/99 ..../.. ..../.. FI; K2=E47 7426 K2/99 ..../.. ..../.. FI; K2=E25 74?? K3/99 ..../.. ..../.. K305=INSTR5-DECODE
221)	5633	TEST045A	4878	7405					BUT(DMO#SMO#BYTE), IR=(004300); DMO, -SMO, BYTE (SWAB) 7404 K3/99 ..../.. ..../.. FI; K3=E85
222)	5635	TEST044A	4833	7447					BUT(INSTR1), IR=(076250), CLASS-A(XFC) 7417 K3/99 ..../.. ..../.. FI; K3=E90, E99 7464 K3/99 ..../.. ..../.. FI; K3=E90, E110

-----  
Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/% #2/% #3/%			Test summary - Print reference - Chip information
223)	5636	TEST105D	5716	7405	SR LOAD/READ, SR=(152500);	ALU-A/D/DBUF/IR PATH			
					7400 K4/99 .....	FI; K4=E27, E35-E36, E39, E42-E43, E48, E67			
					7402 K2/95 K4/5 .....	FI; K2=E6, E9, E12, E18, E26, E30, E33, E39, E45, E51, E54, E60, E66, E69, E75, E86-E88, E94, E99, E103; K4=E77			
224)	5637	TEST043B	4796	7711	BUT(INSTR1), IR=(161707), CLASS-G(DOP*-SMD)				
					5126 K6/99 .....	FI; K6=E42			
					7710 K3/99 .....	FI; K3=E105			
					7717 K3/99 .....	FI; K3=E107			
					7750 K3/99 .....	FI; K3=E90			
225)	5641	TEST043A	4767	7404	BUT(DMO#SMD#BYTE), IR=(161707); DMO, -SMD, -BYTE (SUB)				
					7400 K3/99 .....	FI; K3=E74			
					7406 K3/99 .....	FI; K3=E109			
226)	5642	TEST105C	5690	7412	SR LOAD/READ, SR=(125200);	ALU-A/D/DBUF/IR PATH			
					7400 K4/99 .....	FI; K4=E27, E32-E37, E39, E41-E43, E45-E46, E48, E60, E62, E67-E68, E72			
					7403 K4/99 .....	FI; K4=E33-E34, E39, E45-E46, E54, E64, E77-E78			
					7405 K4/99 .....	FI; K4=E45-E46			
					7420 K4/99 .....	FI; K4=E28			
					7434 K4/99 .....	FI; K4=E29			
227)	5643	TEST042B	4731	7714	BUT(INSTR1), IR=(144020), CLASS-G(DOP*-SMD)				
					7417 K3/99 .....	FI; K3=E90, E110			
					7514 K3/99 .....	FI; K3=E99			
					7614 K3/99 .....	FI; K3=E100			
					7710 K3/99 .....	FI; K3=E106, E120			
					7715 K2/99 .....	FI; K2=E77			
					7717 K3/99 .....	FI; K3=E97, E105-E107			
228)	5645	TEST042A	4703	7401	BUT(DMO#SMD#BYTE), IR=(144020); -DMO, -SMD, BYTE				
					7400 K3/70 K2/30 .....	FI; K3=E54, E60, E65, E85; K2=E19, E41, E77-E78			
					7403 K5/50 K3/50 .....	FI; K5=E98; K3=E109			
					7406 K2/99 .....	FI; K2=E78			
229)	5646	TEST104B	5543	7432	CSP BASCON ADDR, LOC(15);	EMIT/CSP/ALU-B/D/DBUF/IR WITH DATA=(000125)			
					7400 K4/99 .....	FI; K4=E38			
					7402 K4/99 .....	FI; K4=E51, E100-E101			
230)	5647	TEST041B	4667	7517	BUT(INSTR1), IR=(120777), CLASS-B(DOP*-MOV*SMD*-DMO)				
					7617 K3/99 .....	FI; K3=E115			
231)	5651	TEST041A	4638	7412	BUT(IR15-12), IR=(120777); IR<15:12>="1010"				
					7042 K2/99 .....	FI; K2=E1			

-----  
 Module codes: K1/DCS K2/WORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7 STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/%# #2/%# #3/%#			Test summary - Print reference - Chip information
232)	5652	TEST105B	5642	7425	SR LOAD/READ, SR=(000152); 7420 K4/70 K3/30 .....	ALU-A/D/DBUF/IR PATH FI; K4=E31-E32, E35, E37, E40-E41, E44, E52-E53, E57, E61, E75-E76; K3=E43, E51, E74, E76-E77, E82, E102			
					7432 K4/99 .....	FI; K4=E55			
233)	5653	TEST040B	4602	7517	BUT(INSTR1), IR=(050777), 7417 K3/99 .....	CLASS-B(DOP*-MOV*SMD*-DMO) FI; K3=E80			
234)	5655	TEST040A	4574	7405	BUT(IR15-12), IR=(050777); 7401 K3/99 .....	IR<15:12>="0101" FI; K3=E121			
235)	5657	TEST037A	4531	7446	BUT(INSTR1), IR=(060205), 7417 K7/35 K3/35 ^2/35 7447 K3/99 .....	CLASS-A(DOP*SMD*DMO) FI; K7=E17; K3=E115; K2=E73 FI; K3=E94			
236)	5661	TEST036A	4486	7455	BUT(INSTR1), IR=(150506), 7457 K3/99 .....	CLASS-A(DOP*SMD*DMO) FI; K3=E94			
237)	5663	TEST035A	4442	7443	BUT(INSTR1), IR=(030701), 7417 K3/99 .....	CLASS-A(DOP*SMD*DMO) FI; K3=E60, E80, E88, E90, E95, E97, E105, E107, E110, E115-E117, E120			
					7441 K3/45 K6/30 K5/30 7442 K3/99 .....	FI; K3=E43, E94, E107; K6=E91-E92; K5=E67, E98 FI; K3=E94, E105			
					7460 K3/99 .....	FI; K3=E90			
					7540 K3/99 .....	FI; K3=E120			
238)	5665	TEST034A	4398	7511	BUT(INSTR1), IR=(005112), 7517 K3/99 .....	CLASS-B(SOP*-DMO) FI; K3=E98			
					7751 K3/99 .....	FI; K3=E98			
239)	5667	TEST033A	4354	7517	BUT(INSTR1), IR=(106274), 7417 K3/99 .....	CLASS-B(SOP*-DMO) FI; K3=E80, E100, E108, E120			
					7515 K2/99 .....	FI; K2=E114			
					7562 K3/99 .....	FI; K3=E108			
					7716 K3/99 .....	FI; K3=E100			
240)	5670	TEST047B	5019	7413	BUT(IR11#FLOAT), IR=(122566); 74?? K3/99 .....	FP-ROM(534)=(13) K305=FP-INSTR-DECODE			
241)	5671	TEST032B	4317	7612	BUT(INSTR1), IR=(110125), 7451 K3/99 .....	CLASS-C(MOV*SMD*-DMO) FI; K3=E109			
					7602 K3/99 .....	FI; K3=E4, E6-E7, E11, E15, E43, E118			
					7752 K3/99 .....	FI; K3=E85, E94, E104			

-----  
Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/%% #2/%% #3/%%			Test summary - Print reference - Chip information
242)	5673	TEST032A	4288	7412	BUT(MOV/DR7#IRS-3), IR=(110125); IR(5:3)="010", MOV	7402	K3/80	K5/20	FI; K3=E57,E97; K5=E68
243)	5674	TEST103D	5476	7405	CSP ADDRESS, LOC(07); EMIT/CSP/ALU-B/D/DBUF/IR WITH DATA=(152500)	7322	K3/99	FI; K3=E22	
					7400	K4/99	FI; K4=E59,E109-E110		
244)	5675	TEST031B	4253	7604	BUT(INSTR1), IR=(010242), CLASS-C(MOV#SMO*-DMO)	7404	K3/99	FI; K3=E99	
					7417	K3/90	K5/15	FI; K3=E87,E90,E113,E119; K5=E98	
					7514	K3/55	K6/30	K5/15	FI; K3=E115-E117; K6=E91-E92; K5=E67
					7607	K3/99	FI; K3=E98		
					7614	K3/99	FI; K3=E97,E118		
					7640	K3/99	FI; K3=E90		
					7700	K3/99	FI; K3=E99,E110		
					7704	K3/99	FI; K3=E98,E120		
245)	5677	TEST031A	4224	7402	BUT(DMO#SMO#BYTE), IR=(010242); -DMO, SMO, -BYTE	7005	K3/99	FI; K3=E56	
					7400	K3/99	FI; K3=E47,E56,E60,E80,E90,E109,E119-E120		
					7406	K3/99	FI; K3=E109		
246)	5701	TEST030A	4117	7552	BUT(INSTR1), IR=(005204), CLASS-D(SOP#DMO)	7313	K3/99	FI; K3=E2	
					7553	K3/99	FI; K3=E87		
247)	5703	TEST027A	4074	7561	BUT(INSTR1), IR=(106102), CLASS-D(SOP#DMO)	???	K3/99	K304/5=INSTR1-DECODE	
248)	5705	TEST026A	4029	7574	BUT(INSTR1), IR=(105403), CLASS-D(SOP#DMO)	7570	K3/99	FI; K3=E108	
					7574	K2/99	FI; K2=E67		
					7757	K6/99	FI; K6=E82		
249)	5707	TEST025A	3986	7543	BUT(INSTR1), IR=(006303), CLASS-D(SOP#DMO)	7540	K3/99	FI; K3=E99	
					7557	K3/99	FI; K3=E105		
250)	5710	TEST014D	3436	7401	BUT(IR11#FLOAT), IR=(000125); FP-ROM(020)=(01)	7400	K3/99	FI; K3=E86	
251)	5711	TEST024B	3949	7557	BUT(INSTR1), IR=(005706), CLASS-D(SOP#DMO)	5125	K3/99	FI; K3=E12	
					7306	K3/99	FI; K3=E5		
					7407	K3/99	FI; K3=E44		

[Continued]

Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###)	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/%% #2/%% #3/%%			Test summary - Print reference - Chip information
	5711	TEST024B	[Continued]			BUT(INSTR1),	IR=(005706),	CLASS-D(SOP*DMO)	
						7417 K3/99	.../...	.../...	FI; K3=E80, E88, E108, E117
						7547 K3/99	.../...	.../...	FI; K3=E118
						7550 K3/99	.../...	.../...	FI; K3=E120
						7553 K3/99	.../...	.../...	FI; K3=E106
						7555 K3/99	.../...	.../...	FI; K3=E103, E107
						7556 K3/99	.../...	.../...	FI; K3=E103, E105
						7577 K3/99	.../...	.../...	FI; K3=E108
252)	5712	TEST030B	4147	7424		BUT(IR11#FLOAT),	IR=(005204);	FP-ROM(240)=(04)	
						7420 K3/99	.../...	.../...	FI; K3=E86
						7425 K4/60	K2/20	K3/20	FI; K4=E32, E41-E43, E59, E66, E71, E79, E82, E92, E102, E109-E110; K2=E96, E101, E104, E110, E119; K3=E12, E43, E52, E71, E82, E86, E102
						7426 K3/99	.../...	.../...	FI; K3=E86
253)	5713	TEST030C	4166	7403		BUTR(IR11),	IR=(005204);	IR11="1"	
						7401 K3/99	.../...	.../...	K307=MICROBRANCH
254)	5715	TEST024A	3920	7402		BUTR(DR6/7L),	IR=(005706);	DR6	
						7403 K3/99	.../...	.../...	FI; K3=E93
255)	5717	TEST023A	3877	7560		BUT(INSTR1),	IR=(106004),	CLASS-D(SOP*DMO)	
						5122 K3/99	.../...	.../...	FI; K3=E14, E23
						5401 K3/99	.../...	.../...	FI; K3=E1
						7401 K3/99	.../...	.../...	FI; K3=E19
						7417 K3/99	.../...	.../...	FI; K3=E47, E60, E74, E80, E88, E90, E99, E104, E108-E110, E116, E119-E120
						7460 K3/99	.../...	.../...	FI; K3=E100
						7510 K3/99	.../...	.../...	FI; K3=E109
						7520 K3/99	.../...	.../...	FI; K3=E100
						7540 K3/99	.../...	.../...	FI; K3=E85, E108, E110
						7557 K3/99	.../...	.../...	FI; K3=E97, E105-E108, E118
						7561 K3/85	K4/15	.../...	FI; K3=E79, E87, E103, E105; K4=E9
						7570 K3/99	.../...	.../...	FI; K3=E76, E118
						7760 K3/99	.../...	.../...	FI; K3=E96, E104, E114, E119
256)	5721	TEST022A	3833	7757		BUT(INSTR1),	IR=(003063),	CLASS-F(BRANCH)	
						0000 K3/99	.../...	.../...	FI; K3=E6
						5200 K3/99	.../...	.../...	FI; K3=E11
						7417 K3/60	K7/40	.../...	FI; K3=E96, E104; K7=E13, E101
						7517 K3/99	.../...	.../...	FI; K3=E25
257)	5722	TEST015C	3519	7655		BUT(INSTR1),	IR=(000152),	CLASS-E(JMP)	
						7651 K3/99	.../...	.../...	FI; K3=E103, E106
						7654 K3/99	.../...	.../...	FI; K3=E93, E105
						7657 K3/99	.../...	.../...	FI; K3=E69, E89, E103

Module codes: K1/DCS K2/UNWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/%% #2/%% #3/%%			Test summary - Print reference - Chip information
258)	5723	TEST021B	3775	7757	BUT(INSTR1),	IR=(002315),	CLASS-F(BRANCH)		
					7417	K3/99	..../..	..../..	FI; K3=E89,E96,E104,E109
					7751	K3/99	..../..	..../..	FI; K3=E116
259)	5725	TEST021A	3746	7400	BUT(DMO#SMO#BYTE),	IR=(002315);	-DMO, -SMO, -BYTE		
					7401	K3/99	..../..	..../..	FI; K3=E54,E85,E87-E88,E116
					7402	K3/99	..../..	..../..	FI; K3=E56,E109-E110,E114,E119
					7403	K3/50	K6/25	K5/25	FI; K3=E112,E114; K6=E83; K5=E59
					7404	K3/99	..../..	..../..	FI; K3=E74,E90,E109,E119-E120
					7406	K7/99	..../..	..../..	FI; K7=E20
					7600	K2/99	..../..	..../..	FI; K2=E36
					7606	K2/99	..../..	..../..	FI; K2=E15
260)	5726	TEST050B	5108	7412	BUT(INSTR5),	IR=(175200);	E88(752)=(12)		
					7434	K7/99	..../..	..../..	FI; K7=E13,E29,E94,E101
					74??	K3/99	..../..	..../..	K305=INSTR5-DECODE
261)	5727	TEST020C	3711	7757	BUT(INSTR1),	IR=(001257),	CLASS-F(BRANCH)		
					7417	K3/99	..../..	..../..	FI; K3=E95-E96,E104,E114
					7740	K3/99	..../..	..../..	FI; K3=E99
					7755	K3/99	..../..	..../..	FI; K3=E117
262)	5730	TEST047C	5039	7406	BUT(MOV/DR7#IR5-3),	IR=(122566);	IR<5:3>="110",	-MOV	
					740?	K3/99	..../..	..../..	K308=MICROBRANCH
263)	5731	TEST020B	3689	7405	BUT(MOV/DR7#IR5-3),	IR=(001257);	IR<5:3>="101",	-MOV	
					7407	K3/99	..../..	..../..	FI; K3=E54
					7415	K3/99	..../..	..../..	FI; K3=E94,E120
264)	5733	TEST020A	3662	7412	BUT(IR9-6),	IR=(001257);	IR<9:6>="1010"		
					7232	K3/99	..../..	..../..	FI; K3=E58
					7402	K3/70	K6/30	..../..	FI; K3=E76,E83,E85-E86,E88; K6=E82-E83
					7413	K6/99	..../..	..../..	FI; K6=E73
265)	5734	TEST030D	4189	7405	BUT(INSTR5),	IR=(005204);	E88(452)=(05)		
					7400	K3/99	..../..	..../..	FI; K3=E88
					7403	K3/99	..../..	..../..	FI; K3=E88
					7407	K3/99	..../..	..../..	FI; K3=E88
					7415	K3/99	..../..	..../..	FI; K3=E88
					7417	K3/99	..../..	..../..	FI; K3=E88
					7425	K3/99	..../..	..../..	FI; K3=E68
					7426	K3/99	..../..	..../..	FI; K3=E38,E78,E109
					7437	K3/99	..../..	..../..	FI; K3=E38
					74??	K3/99	..../..	..../..	K305=INSTR5-DECODE

-----  
 Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS  
 -----

ERROR	Line	->Module sequence->			Test summary - Print reference - Chip information	
###) code	Symbolic label	number	ENUA	TNUA	#1/% #2/% #3/%	
266) 5735	TEST017B	3626	7757	BUT(INSTR1),	IR=(000522), CLASS-F(BRANCH)	
				7417 K3/99	... ..	FI; K3=E109-E110
				7652 K3/99	... ..	FI; K3=E114
				7717 K3/99	... ..	FI; K3=E89
				7752 K3/99	... ..	FI; K3=E96
267) 5737	TEST017A	3598	7405	BUT(IR9-6),	IR=(000522); IR<9:6>="0101"	
				7401 K3/90	K5/10 ... ..	FI; K3=E50,E79,E83-E86,E88,E114; K5=E98
				7405 K3/99	... ..	FI; K3=E23
				7415 K6/65	K5/35 ... ..	FI; K6=E82-E83; K5=E59
268) 5741	TEST016A	3555	7757	BUT(INSTR1),	IR=(100000), CLASS-F(BRANCH)	
				7400 K3/99	... ..	FI; K3=E51
				7417 K3/75	K6/15 K5/10	FI; K3=E43,E95,E98,E110,E113; K6=E91-E92; K5=E98
				7512 K6/99	... ..	FI; K6=E74
				7557 K3/99	... ..	FI; K3=E99
				7657 K3/99	... ..	FI; K3=E58,E70,E100
				7717 K3/99	... ..	FI; K3=E100
				7760 K3/99	... ..	FI; K3=E100
				7770 K3/99	... ..	FI; K3=E95,E110,E115,E119
				7777 K3/99	... ..	FI; K3=E108
269) 5743	TEST015A	3473	7425	BUT(INSTR5),	IR=(000152); E78(152)=(25)	
				5076 K3/99	... ..	FI; K3=E10
				5673 K3/99	... ..	FI; K3=E3
				7305 K3/99	... ..	FI; K3=E20
				7400 K3/99	... ..	FI; K3=E33,E106
				7420 K4/50	K3/35 K6/15	FI; K4=E12,E14; K3=E75,E103; K6=E65
				7424 K3/99	... ..	FI; K3=E67,E78,E88
				7434 K2/99	... ..	FI; K2=E103,E111,E114
				7435 K3/99	... ..	FI; K3=E76,E78
				74?? K3/99	... ..	K305=INSTR5-DECODE
				7655 K3/99	... ..	FI; K3=E36
270) 5745	TEST014A	3368	7432	BUT(INSTR5),	IR=(000125); E78(125)=(32)	
				7400 K3/99	... ..	FI; K3=E88
				7420 K3/50	K6/25 K5/20	K2/5 FI; K3=E66-E67,E77-E78,E84-E88,E93,E101, E103,E112; K6=E65-E66,E73-E74,E83,E92; K5=E67, E70,E106; K2=E112
				7422 K3/99	... ..	FI; K3=E78,E88
				7430 K3/99	... ..	FI; K3=E78,E88
				7434 K2/55	K7/15 K6/15	K3/15 FI; K2=E33,E71,E102,E108; K7=E17; K6=E91; K3=E43
				7436 K3/99	... ..	FI; K3=E75,E78
				[Continued] 74?? K3/99	... ..	K305=INSTR5-DECODE

Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS



###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/%% #2/%% #3/%%			Test summary - Print reference - Chip information	
	5745	TEST014A	[Continued]			BUT(INSTR5),	IR=(000125);	E78(125)=(32)		
					7672	K3/99	..../..	..../..	FI; K3=E98	
271)	5746	SFDFT05R	10415	7777		(NUA SEQUENCING LOGIC ERROR)				
					7777	K1/99	..../..	..../..	INTERNAL DCS ERROR	
272)	5747	TEST013G	3331	7417		BUT(INSTR1),	IR=(000000),	CLASS-OTHER(NOT-A-THRU-G)		
					7400	K3/99	..../..	..../..	FI; K3=E97,E115	
					7407	K3/99	..../..	..../..	FI; K3=E76,E97,E118	
					7410	K3/99	..../..	..../..	FI; K3=E97-E98	
					7413	K3/99	..../..	..../..	FI; K3=E73,E106	
					7415	K3/99	..../..	..../..	FI; K3=E77,E107	
					7416	K3/99	..../..	..../..	FI; K3=E67,E105	
					7437	K3/99	..../..	..../..	FI; K3=E68,E108	
					7440	K3/99	..../..	..../..	FI; K3=E90,E95,E99,E113,E115-E117,E121	
					7457	K3/99	..../..	..../..	FI; K3=E70,E97,E100	
					7460	K3/99	..../..	..../..	FI; K3=E47,E57,E90,E94,E97,E108,E110	
					7510	K3/99	..../..	..../..	FI; K3=E120	
					7540	K3/99	..../..	..../..	FI; K3=E108	
					7650	K3/99	..../..	..../..	FI; K3=E96,E98	
					7710	K3/99	..../..	..../..	FI; K3=E110	
					7757	K3/99	..../..	..../..	FI; K3=E59,E110	
					7776	K2/99	..../..	..../..	FI; K2=E32	
273)	5750	TEST013F	3312	434		BUT(INSTR5),	IR=(000000);	E78(000)=(34)		
					7400	K3/95	K4/5	K2/5	FI; K3=E38,E59,E68,E88-E89,E96,E98-E99,E104-E107, E109-E110,E113-E114,E116-E119; K4=E4; K2=E25	
					7402	K3/99	..../..	..../..	FI; K3=E104,E106,E113,E115-E117	
					7414	K3/99	..../..	..../..	FI; K3=E89	
					7420	K3/99	..../..	..../..	FI; K3=E78	
					7421	K2/55	K3/25	K5/20	FI; K2=E91,E111; K3=E78,E93; K5=E94	
					7422	K2/50	K3/30	K5/20	FI; K2=E91,E111; K3=E78,E93; K5=E94	
					7424	K3/55	K2/35	K5/15	FI; K3=E76,E78,E88,E93; K2=E91,E103; K5=E94	
					7426	K3/99	..../..	..../..	FI; K3=E78	
					7427	K2/99	..../..	..../..	FI; K2=E71	
					7430	K3/99	..../..	..../..	FI; K3=E75,E78,E88	
					7435	K3/99	..../..	..../..	FI; K3=E67,E78	
					7436	K3/99	..../..	..../..	FI; K3=E77-E78	
					7437	K3/99	..../..	..../..	FI; K3=E78	
274)	5751	TEST013E	3291	7400		BUT(MOV/DR7#IR5-3),	IR=(000000);	IR<5:3>="000" -MOV		
					7401	K2/50	K3/30	K5/15	K6/5	FI; K2=E84,E91,E103; K3=E67,E93; K5=E94; K6=E65
					7402	K3/45	K2/35	K5/15	K6/5	FI; K3=E77,E103,E107,E109; K2=E91,E114; K5=E105; K6=E73
					7404	K2/45	K3/35	K5/20		FI; K2=E91,E114; K3=E75,E103; K5=E105
					7410	K3/99	..../..	..../..		FI; K3=E47,E57,E76,E113,E115-E117,E119-E120

Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNJA	->Module sequence-> #1/% #2/% #3/%			Test summary - Print reference - Chip information		
275)	5752	TEST0148	3394	7403	BUTR(DR6/7L), IR=(000125); DR5=-(DR6+7)	7402	K3/99	../..	../..	FI; K3=E37-E39,E64,E66,E79	
276)	5753	TEST0130	3272	7400	BUT(IR9-6), IR=(000000); IR<9:6>="0000"	7401	K3/99	../..	../..	FI; K3=E67	
						7402	K3/99	../..	../..	FI; K3=E77,E103	
						7404	K3/99	../..	../..	FI; K3=E75	
						7410	K3/99	../..	../..	FI; K3=E76	
277)	5754	TEST621H	15572	7402	CHECK "WR-CSP" LATCH CLEARED IN JAMMED U-WORD, WRITE NOT DONE	7401	K7/65	K3/35	../..	FI; K7=E25,E41; K3=E32	
278)	5755	TEST013C	3253	7401	BUTR(IR11), IR=(000000); IR11="0"	7403	K3/99	../..	../..	FI; K3=E39,E56	
279)	5756	TEST624D	15864	7403	BUTR(PREFETCH*JAM) ASSERTED="1", GETS PREFETCH-H AT JAMUPP="1"	740?	K3/99	../..	../..	K306=PREFETCH*JAM-LOGIC	
						7747	K3/99	../..	../..	FI; K3=E17	
280)	5757	TEST013B	3233	7400	BUT(IR11#FLOAT), IR=(000000); IR<11>="0", FP-ROM(000)=(00)	7401	K2/45	K3/30	K5/15	K6/5	FI; K2=E97,E113; K3=E67,E86,E103; K5=E105; K6=E66,E73
						7402	K2/45	K3/35	K5/15	K6/5	FI; K2=E97-E98; K3=E77,E86,E114; K5=E97; K6=E82
						7404	K2/50	K3/35	K5/15	FI; K2=E97-E98; K3=E75,E86,E114; K5=E97	
						7410	K2/45	K3/40	K5/15	FI; K2=E97,E107; K3=E76,E86,E114; K5=E97	
						7417	K3/99	../..	../..	FI; K3=E86	
						7420	K2/50	K3/25	K5/15	K6/10	FI; K2=E97,E107; K3=E68,E114; K5=E97; K6=E82
291)	5761	TEST013A	3205	7400	BUT(IR15-12), IR=(000000); IR<15:12>="0000"	7401	K2/55	K3/25	K5/20	FI; K2=E85,E102; K3=E67,E113; K5=E95	
						7402	K2/50	K3/30	K5/20	FI; K2=E85,E102; K3=E77,E113; K5=E95	
						7404	K2/45	K3/35	K5/20	FI; K2=E85,E108; K3=E75,E113; K5=E95	
						7405	K2/99	../..	../..	FI; K2=E102-E103,E108,E111,E114	
						7410	K2/45	K3/30	K5/15	K6/10	FI; K2=E85,E108; K3=E76,E113; K5=E95; K6=E91
						7417	K2/60	K5/20	K7/10	K6/5	FI; K2=E1,E17,E23,E34,E55-E56,E62,E70-E71,E76,E82-E83,E85,E92,E100-E101,E105; K5=E54,E84,E94-E95,E97,E105; K7=E28,E37,E54; K6=E101; K4=E26,E29,E38,E87
282)	5762	TEST014C	3414	7652	BUT(INSTR1), IR=(000125), CLASS-E(JMP)	7417	K3/99	../..	../..	FI; K3=E96,E98,E103,E107	
						7452	K3/99	../..	../..	FI; K3=E58,E70,E99	
						7612	K3/99	../..	../..	FI; K3=E100	
			[Continued]			7640	K3/99	../..	../..	FI; K3=E90	

Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

ERROR ###) code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/% #2/% #3/%			Test summary - Print reference - Chip information	
5762	TEST014C	[Continued]		BUT(INSTR1),	IR=(000125),	CLASS-E(JMP)			
				7642	K3/99	.../...	.../...	FI; K3=E90, E98, E119	
				7650	K3/99	.../...	.../...	FI; K3=E103, E107	
				7653	K3/99	.../...	.../...	FI; K3=E58, E79, E93, E105, E109	
				7656	K3/99	.../...	.../...	FI; K3=E69, E75, E89, E103, E106, E109	
				7657	K3/99	.../...	.../...	FI; K3=E98, E105-E107	
				7740	K3/99	.../...	.../...	FI; K3=E100	
				7752	K3/99	.../...	.../...	FI; K3=E99, E120	
283)	5763	TEST012G	3169	7474	BUT(INSTR1),	IR=(177777),	CLASS-A(FLOATING-POINT), FLAG<4:5>="00"		
				###	.../...	.../...	.../...	!!! WATCH FOR BUSDIN PULLED LOW BY OTHER DRIVER !!!	
				7434	K3/99	.../...	.../...	FI; K3=E37, E58, E70, E97-E98, E100	
				7437	K3/99	.../...	.../...	FI; K3=E90, E97, E105-E107, E118	
				7454	K3/99	.../...	.../...	FI; K3=E68, E108	
				7464	K3/99	.../...	.../...	FI; K3=E76, E113, E118	
				7470	K3/99	.../...	.../...	FI; K3=E75, E106	
				7475	K4/45	K3/40	K2/15	K7/5 FI; K4=E3, E33-E34, E51, E55, E71, E82, E100-E101; K3=E3, E43, E51, E67, E82, E94, E102, E105; K2=E77, E95, E106; K7=E43	
				7476	K4/50	K3/35	K2/15	K5/5 FI; K4=E3, E42-E43, E55, E59, E82, E109-E110; K3=E41, E43, E52, E77, E82, E94, E107; K2=E96, E100, E106; K5=E59	
				7477	K3/65	K4/40	.../...	FI; K3=E1, E41, E43, E51-E52, E61, E82, E102; K4=E3, E26, E29, E55, E70-E71	
				7574	K3/99	.../...	.../...	FI; K3=E70, E97-E98, E100, E108, E116-E117, E120-E121	
				7674	K3/99	.../...	.../...	FI; K3=E70, E90, E97-E100	
				7774	K3/99	.../...	.../...	FI; K3=E95-E97, E99-E100, E104, E110, E119	
284)	5764	TEST012F	3147	7402	BUTR(DR6/7L),	IR=(177777);	DR7		
				###	.../...	.../...	.../...	!!! WATCH FOR BUSDIN PULLED LOW BY OTHER DRIVER !!!	
				7403	K3/99	.../...	.../...	FI; K3=E64, E66	
285)	5765	TEST012E	3127	7417	BUT(MOV/DR7#IR5-3),	IR=(177777);	IR<5:3>="111", -MOV		
				###	.../...	.../...	.../...	!!! WATCH FOR BUSDIN PULLED LOW BY OTHER DRIVER !!!	
				7400	K2/99	.../...	.../...	FI; K2=E26, E91	
				7406	K3/75	K5/25	.../...	FI; K3=E93; K5=E94	
				7407	K3/60	K5/20	K2/10	K4/10 K6/5 K7/5 FI; K3=E57, E66, E71, E76, E78-E79, E81, E84, E92-E93, E95, E99, E101, E105-E107, E110, E113, E115-E119; K5=E48-E49, E67-E68, E94, E106, E113-E115; K2=E91, E103, E111-E112; K4=E12-E15, E106; K6=E65-E66; K7=E73, E82	
				7413	K3/35	K5/30	K2/15	K6/5 K4/5 K7/5 FI; K3=E75, E78, E86-E87, E101, E103; K5=E56, E61, E70, E92, E105-E106; K2=E91, E114; K6=E73-E74; K4=E106; K7=E88	
				7415	K5/40	K3/30	K2/10	K7/5 K6/5 K4/5 FI; K5=E56, E61, E67, E105-E106, E112; K3=E77-E78, E86-E87, E101, E103; K2=E91, E114; K7=E88; K6=E73-E74; K4=E106	
				[Continued]	7416	K5/35	K3/35	K2/15	K6/10 K4/5 K7/5 FI; K5=E49, E67-E68, E94, E106,

Module codes: K1/DCS K2/WORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

ERROR ###) code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/% #2/% #3/%			Test summary - Print reference - Chip information
5765	TEST012E	[Continued]						BUT(MOV/DR7#IR5-3), IR=(177777); IR<5:3>="111", -MOV 7416 K5/35 K3/35 K2/15 E120; K3=E67,E78,E86,E93,E101; K2=E91,E103; K6=E65-E66; K4=E106; K7=E82 7437 K3/99 .../... FI; K3=E68
286) 5766	TEST621G	15550	7401					BUTR(PREFETCH#JAM) NEGATED; GETS PREFETCH-H AT JAMUPP="0" 7403 K3/99 .../... FI; K3=E44 7407 K3/99 .../... K306=PREFETCH#JAM-LOGIC 7763 K2/75 K3/25 .../... FI; K2=E10-E12,E37-E39,E64-E66; K3=E2,E4-E5
287) 5767	TEST012D	3108	7417					BUT(IR9-6), IR=(177777); IR<9:6>="1111" #### .../... !!! WATCH FOR BUSDIN PULLED LOW BY OTHER DRIVER !!! 7407 K3/99 .../... FI; K3=E76 7412 K3/99 .../... FI; K3=E34,E54,E64,E67-E68,E74-E75,E98 7414 K3/75 K5/25 .../... FI; K3=E103; K5=E105 7415 K3/45 K5/25 K2/10 K4/10 K6/5 K7/5 FI; K3=E34,E44,E50,E56, E76-E79,E84-E86,E88,E103,E112; K5=E51,E61,E70, E105-E106,E121; K2=E97,E113; K4=E12,E14,E105; K6=E73-E74; K7=E64 7416 K3/40 K5/30 K2/15 K7/5 K6/5 K4/5 FI; K3=E67,E78,E84-E86,E88, E96,E103,E112; K5=E51,E61,E70,E105-E106,E122; K2=E97,E113; K7=E64; K6=E73-E74; K4=E105 7437 K3/99 .../... FI; K3=E68 7457 K3/99 .../... FI; K3=E98
288) 5770	TEST0468	4950	7401					BUT(IR11#FLOAT), IR=(000200); FP-ROM(040)=(01) 7477 K3/99 .../... K305=FP-INSTR-DECODE
289) 5771	TEST012C	3088	7403					BUTR(IR11), IR=(177777); IR11="1" #### .../... !!! WATCH FOR BUSDIN PULLED LOW BY OTHER DRIVER !!! 5403 K2/99 .../... FI; K2=E34 7401 K3/99 .../... FI; K3=E23,E33-E34,E37-E39,E44,E56,E70,E76-E77 7407 K3/99 .../... FI; K3=E74 7443 K3/99 .../... FI; K3=E37,E98 7475 K2/90 K3/15 .../... FI; K2=E34,E40,E59,E65,E68,E74,E80; K3=E39 7477 K3/99 .../... FI; K3=E39 7767 K2/99 .../... FI; K2=E40
290) 5772	TEST0158	3500	7403					BUTR(DR6/7L), IR=(000152); DR2=-(DR6+7) 7402 K3/99 .../... FI; K3=E66,E79
291) 5773	TEST012B	3069	7437					BUT(IR11#FLOAT), IR=(177777); IR<11>="1", FP-ROM(776)=(17) #### .../... !!! WATCH FOR BUSDIN PULLED LOW BY OTHER DRIVER !!! 7400 K7/50 K6/15 K3/15 K2/15 FI; K7=E80; K6=E91; K3=E34; K2=E97 7401 K3/75 K5/25 .../... FI; K3=E114; K5=E97 7403 K3/99 .../... FI; K3=E34 7411 K7/99 .../... FI; K7=E80 [Continued] 7413 K3/99 .../... FI; K3=E34

Module codes: K1/DCS K2/WORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

ERROR ###) code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/%% #2/%% #3/%%			Test summary - Print reference - Chip information
--------------------	----------------	----------------	------	------	--	--	--	---

5773 TEST0128

[Continued]

BUT(IR11#FLOAT), IR=(177777); IR<11>="1", FP-ROM(776)=(17)

7416 K6/99 ..../.. ..../.. FI; K6=E91

7417 K3/65 K5/15 K2/10 K6/5 K7/5 K4/5 FI; K3=E34, E52, E54, E56, E64, E67-E68, E74-E75, E87-E88, E98-E99, E102, E114, E116-E117; K5=E20, E59-E60, E97-E98, E107; K2=E97, E107; K6=E82-E83; K7=E81; K4=E38, E105

7427 K3/55 K5/20 K2/10 K7/5 K6/5 K4/5 FI; K3=E34, E44, E56, E76-E77, E83, E85-E88, E99, E112, E114, E116; K5=E20, E59-E60, E97-E98, E104; K2=E97, E107; K7=E81; K6=E82-E83; K4=E105

7433 K3/99 ..../.. ..../.. FI; K3=E75, E86

7434 K3/40 K5/30 K2/15 K7/5 K6/5 K4/5 FI; K3=E76, E83, E85-E86, E88, E112, E114, E117; K5=E21, E59-E60, E97-E98, E108; K2=E97-E98; K7=E56; K6=E82-E83; K4=E105

7435 K3/99 ..../.. ..../.. FI; K3=E77, E86

7436 K3/50 K5/25 K2/10 K7/5 K6/5 K4/5 FI; K3=E50, E67, E75, E79, E83-E86, E88-E89, E112, E114; K5=E21, E59-E60, E97-E99; K2=E97-E98; K7=E56; K6=E82-E83; K4=E105

292) 5774 TEST021C

3797 7411

BUT(IR11#FLOAT), IR=(002315); FP-ROM(462)=(11)

7401 K3/99 ..../.. ..../.. FI; K3=E86

7416 K5/99 ..../.. ..../.. FI; K5=E59

74?? K3/99 ..../.. ..../.. K305=FP-INSTR-DECODE

293) 5775 TEST012A

3031 7417

BUT(IR15-12), IR=(177777); IR<15:12>="1111"

8888 ..../.. ..../.. !!! WATCH FOR BUSDIN PULLED LOW BY OTHER DRIVER !!!

3000 K7/99 ..../.. ..../.. FI; K7=E10, E94

3020 K7/99 ..../.. ..../.. FI; K7=E27

3220 K7/99 ..../.. ..../.. FI; K7=E10, E21, E27, E29, E42, E52

4102 K6/99 ..../.. ..../.. FI; K6=E70, E74, E92, E105

5760 K2/99 ..../.. ..../.. FI; K2=E61

7005 K2/55 K5/25 K3/25 FI; K2=E5, E29, E34, E38-E39, E50, E61, E68, E71, E76, E83-E85; K5=E20-E21, E48-E49, E51, E60; K3=E34, E37, E39, E51, E103, E112

7376 K2/99 ..../.. ..../.. FI; K2=E46

7400 K2/40 K3/25 K7/15 K5/15 K6/10 K4/5 FI; K2=E1, E5, E11, E17, E23, E25-E26, E29, E32-E34, E38-E39, E44-E46, E50-E51, E53-E54, E56, E59, E61-E62, E65, E70-E71, E74, E76-E77, E80, E82-E86, E91, E97-E98, E102-E103, E105, E107-E108, E111, E113-E114; K3=E23, E31-E34, E36-E37, E43, E45, E48, E51-E52, E57, E65-E66, E73, E92-E93, E101, E103, E112-E114; K7=E10, E13, E17, E20-E21, E27, E29, E33, E41, E52, E94; K5=E6, E20-E21, E31, E48-E49, E51, E54, E56, E61, E77, E95; K6=E65-E66, E70, E73-E74, E79-E80, E82-E83, E86, E95-E96, E98-E99, E101, E105, E108; K4=E29-E30, E38

[Continued] 7401 K3/99 ..../.. ..../.. FI; K3=E113

7402 K2/99 ..../.. ..../.. FI; K2=E29, E34, E61

Module codes: K1/DCS K2/UNORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/% #2/% #3/%			Test summary - Print reference - Chip information
	5775	TEST012A	[Continued]			BUT(IR15-12), IR=(177777); IR<15:12>="1111"			
					7403	K7/96	K2/5	.../..	FI; K7=E28, E56, E64-E65, E73, E81-E82, E88; K2=E61
					7404	K6/50	K2/50	.../..	FI; K6=E105; K2=E76
					7405	K2/55	K6/25	K3/15	K5/5 FI; K2=E34, E71, E83-E85, E98, E102-E103, E107-E108, E111, E113-E114; K6=E3, E5; K3=E34, E45; K5=E65
					7407	K3/50	K5/25	K2/10	K6/5 K7/5 K4/5 FI; K3=E21, E76, E83, E85, E87-E88, E108, E110, E113; K5=E31, E77, E91, E95, E98; K2=E85, E108; K6=E91-E92; K7=E80; K4=E95
					7412	K3/85	K5/10	K7/5	K2/5 FI; K3=E34, E54, E64, E67-E68, E74-E75; K5=E68, E92; K7=E33; K2=E61
					7413	K5/45	K3/30	K6/10	K2/10 K7/5 K4/5 FI; K5=E31, E59, E67, E70, E77, E84, E95, E98, E101; K3=E21, E75, E85, E113, E121; K6=E91-E92, E95, E105; K2=E85, E108; K7=E80; K4=E95
					7414	K5/99	.../..	.../..	FI; K5=E60-E61, E77
					7415	K3/40	K5/25	K2/15	K6/5 K4/5 K7/5 FI; K3=E21, E77, E85, E94, E104, E113; K5=E6, E67, E77, E90, E95, E98; K2=E85, E102; K6=E91-E92; K4=E95; K7=E65
					7416	K3/40	K5/30	K2/10	K6/5 K4/5 K7/5 FI; K3=E21, E67, E85, E94, E104, E113; K5=E6, E67, E77, E95, E98, E100; K2=E85, E102; K6=E91-E92; K4=E95; K7=E65
					7437	K3/99	.../..	.../..	FI; K3=E34, E38, E68
					7447	K3/99	.../..	.../..	FI; K3=E70, E98
294)	5776	TEST624C	15839	7402	PROC-MUX(CUA-PORT)=(062551); CUA LOCKED, EXFLAG CLEAR, PREFETCH+JAM SET				
					7216	K3/99	.../..	.../..	FI; K3=E48
					7400	K3/75	K2/25	.../..	FI; K3=E44, E48; K2=E111
295)	5777	TEST011	2847	6532	BUTA(SUBR-8) -> BUTA(RETURN) SEQUENCE, RETURN="1101 0101 1010"=(6532)				
					0777	K2/99	.../..	.../..	FI; K2=E55
					4644	K2/99	.../..	.../..	FI; K2=E46
					4747	K2/70	K4/20	K3/5	FI; K2=E5, E11, E23, E34, E46, E55; K4=E3; K3=E23
					5245	K6/99	.../..	.../..	FI; K6=E80
					6132	K2/99	.../..	.../..	FI; K2=E19, E41, E77-E78
					6432	K2/99	.../..	.../..	FI; K2=E7, E13, E25
					6512	K2/99	.../..	.../..	FI; K2=E7, E13, E25
					6532	K2/65	K4/35	.../..	FI; K2=E26, E97; K1=E15
					6533	K2/99	.../..	.../..	FI; K2=E8, E14, E26
					6536	K2/99	.../..	.../..	FI; K2=E8, E14, E26
					7005	K3/55	K2/45	.../..	FI; K3=E4, E6-E7, E11, E15, E43; K2=E70, E85
					7271	K3/60	K2/40	.../..	FI; K3=E4, E6-E7, E11, E15, E43; K2=E61, E77, E85
					7274	K2/70	K3/30	.../..	FI; K2=E1, E3, E6-E7, E12, E15, E18, E24, E30, E36; K3=E23, E58, E70
					7275	K2/99	.../..	.../..	FI; K2=E8, E25
					7301	K6/99	.../..	.../..	FI; K6=E104
					7417	K3/99	.../..	.../..	FI; K3=E23
					7532	K2/99	.../..	.../..	FI; K2=E35, E41, E77
					7577	K2/99	.../..	.../..	FI; K2=E42

Module codes: K1/DCS K2/WORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###)	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/% #2/% #3/%	Test summary - Print reference - Chip information
296)	6252	TEST002	2654	6631		NUA SEQUENCING, NO 'BUT' 2631 K2/99 ..../.. FI; K2=E55 6231 K3/55 K2/45 ..../.. FI; K3=E63,E111; K2=E1,E19 6777 K3/50 K2/50 ..../.. FI; K3=E11; K2=E32	
297)	6520	TEST720B	16788	7402		DATIB*-BYTE, ODD ADDR JAM; JAM=(101004) 7400 K6/99 ..../.. FI; K6=E52	
298)	6521	TEST7130	16689	7402		DATI-NOINT, I/O PAGE, 16. BIT I/O PAGE MODE; SERVICE=(101740) 7400 K6/99 ..../.. FI; K6=E51 7401 K2/99 ..../.. FI; K2=E8,E14,E26	
299)	6522	TEST713C	16664	7403		BUTR(OTHER-JAM#FP), ILLEGAL INTERNAL ADDR; OTHER-JAM="1" 7401 K7/99 ..../.. FI; K7=E74	
300)	6523	TEST7120	16555	7402		DATI, I/O PAGE, 16. BIT I/O PAGE MODE; SERVICE=(101740) 7400 K7/99 ..../.. FI; K7=E82 7403 K7/65 K6/35 ..../.. FI; K7=E80; K6=E89	
301)	6524	TEST712C	16530	7401		BUTR(OTHER-JAM#FP), VALID INTERNAL ADDR; OTHER-JAM="0" 7403 K2/85 K5/15 ..../.. FI; K2=E26,E88,E104,E110,E116; K5=E105	
302)	6526	TEST711C	16427	7402		CLEAR JAM ERRORS/SSYN TIMEOUT; JAM=(001000) 7400 K6/90 K7/10 ..../.. FI; K6=E3,E51,E80-E91,E99; K7=E58	
303)	6530	TEST711C	16404	7402		DATOB*BYTE, I/O PAGE, 16. BIT I/O PAGE MODE; SERVICE=(005740) 7400 K2/70 K4/20 K7/5 K6/5 FI; K2=E1,E8,E14,E26,E31,E37,E43,E49,E52, E71,E82,E91,E105; K4=E13,E15; K7=E81,E97; K6=E34 7401 K2/99 ..../.. FI; K2=E26	
304)	6534	TEST711B	16374	7402		DATOB*BYTE, SSYN TIMEOUT; JAM=(021200) 7400 K6/65 K7/30 K5/5 FI; K6=E50-E52,E81,E99; K7=E58,E64-E65,E84; K5=E84 7401 K6/50 K7/25 K5/20 K3/10 FI; K6=E47,E55,E76,E85; K7=E81,E84; K5=E64, E67,E93; K3=E34	
305)	6535	TEST512E2	13261	7412		KT-BSPHI(SF)=(06), NEGATED 7400 K3/99 ..../.. FI; K3=E47 74?? K3/99 ..../.. K312=KT-SP-ADDR-DECODE	
306)	6537	TEST51201	13185	7400		KT-ASPHI(SF)=(16), ASSERTED, -F2*SR6*PS15 7412 K3/99 ..../.. FI; K3=E47 74?? K3/99 ..../.. K312=KT-SP-ADDR-DECODE	
307)	6541	TEST373A	9738 [Continued]	4777		BUTA(DIAGNOSE) CAUSES XFER TO B.M. ROM/JAMUPP TO DCS RETURN 0000 K2/99 ..../.. FI; K2=E24,E30,E39,E45,E48,E51,E54,E60,E66,E68-E69.	

Module codes: K1/DCS K2/WORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

ERROR                      Line                      ->Module sequence->  
 ###) code    Symbolic label    number    ENUA    TNUA    #1/%    #2/%    #3/%    Test summary - Print reference - Chip information

###) code	Symbolic label	Line number	ENUA	TNUA	#1/%	#2/%	#3/%	Test summary - Print reference - Chip information
6541	TEST373A	[Continued]		BUTA(DIAGNOSE)	CAUSES	XFER	TO	B.M. ROM/JAMUPP TO DCS RETURN
				0000	K2/99	.../..	.../..	E78, E81
				0003	K3/99	.../..	.../..	FI; K3=E23
				0020	K2/99	.../..	.../..	FI; K2=E21
				0055	K2/99	.../..	.../..	FI; K2=E15
				0212	K2/99	.../..	.../..	FI; K2=E4, E10, E16, E22, E27-E28, E31, E36-E37, E42-E43, E49, E52, E58, E64, E67, E73, E79
				0777	K2/99	.../..	.../..	FI; K2=E24
				1020	K2/99	.../..	.../..	FI; K2=E50
				1025	K6/99	.../..	.../..	FI; K6=E42
				2000	K2/99	.../..	.../..	FI; K2=E48, E51, E57, E63
				2042	K2/99	.../..	.../..	FI; K2=E21, E74
				2153	K2/99	.../..	.../..	FI; K2=E75
				2211	K2/99	.../..	.../..	FI; K2=E75
				2247	K3/99	.../..	.../..	FI; K3=E22
				2310	K2/99	.../..	.../..	FI; K2=E75
				2314	K2/99	.../..	.../..	FI; K2=E75
				2405	K2/99	.../..	.../..	FI; K2=E75
				2442	K2/99	.../..	.../..	FI; K2=E21
				2526	K2/99	.../..	.../..	FI; K2=E24
				2530	K2/99	.../..	.../..	FI; K2=E24
				2577	K2/99	.../..	.../..	FI; K2=E48
				2671	K2/99	.../..	.../..	FI; K2=E57
				3007	K2/99	.../..	.../..	FI; K2=E57, E78
				3055	K2/99	.../..	.../..	FI; K2=E5, E11, E15, E17, E23, E29, E32, E38, E44, E50, E53, E59, E65, E68, E74, E80
				3353	K2/99	.../..	.../..	FI; K2=E75
				3462	K2/99	.../..	.../..	FI; K2=E42, E74
				3505	K2/99	.../..	.../..	FI; K2=E15, E21, E74
				3507	K2/99	.../..	.../..	FI; K2=E21
				3522	K2/99	.../..	.../..	FI; K2=E5, E11, E15, E17, E21, E27, E29, E32, E44, E50, E53, E68, E73-E74, E80
				3637	K2/99	.../..	.../..	FI; K2=E45, E68
				3777	K2/99	.../..	.../..	FI; K2=E42
				4621	K2/99	.../..	.../..	FI; K2=E48
				4747	K3/85	K2/15	.../..	FI; K3=E8-E9; K2=E18
				5501	K2/99	.../..	.../..	FI; K2=E48, E57, E63, E78
				5555	K2/99	.../..	.../..	FI; K2=E48, E57, E63, E78
				5557	K2/99	.../..	.../..	FI; K2=E78
				6073	K2/99	.../..	.../..	FI; K2=E73
				7005	K2/99	.../..	.../..	FI; K2=E6, E12, E18, E24, E33, E45, E75
				7037	K4/50	K2/50	.../..	FI; K4=E66; K2=E88
				7274	K2/99	.../..	.../..	FI; K2=E35, E47, E55
				7373	K2/99	.../..	.../..	FI; K2=E36, E73
				7637	K6/60	K7/40	.../..	FI; K6=E39-E40, E42-E43, E56; K7=E36, E52
				7777	K2/99	.../..	.../..	FI; K2=E47

-----

Module codes:    K1/DCS    K2/UWORD    K3/IRDECODE    K4/DATAPATH    K5/KTCACHE    K6/TIMING    K7/STATUS

-----



###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/%# #2/%# #3/%#			Test summary - Print reference - Chip information
308)	6542	TEST512A1	13026	7405	KT-ASPHI(SF)=(02), NEGATED, -SR6 7412 K3/99 ..../.. ..../.. FI: K3=E45,E47 74?? K3/99 ..../.. ..../.. K312=KT-SP-ADDR-DECODE				
309)	6544	TEST512A2	13053	7400	KT-BSPHI(DF)=(16), ASSERTED, DR6*PS15*FLOAT 7412 K3/85 K4/15 ..../.. FI: K3=E45,E47,E58,E69,E79; K4=E13,E15 74?? K3/99 ..../.. ..../.. K312=KT-SP-ADDR-DECODE				
310)	6546	TEST512B2	13108	7400	KT-BSPHI(SF)=(16), ASSERTED, F2*SR1*SMD*PS13*-FLOAT 7412 K3/65 K4/35 ..../.. FI: K3=E45,E47,E89; K4=E13,E15 74?? K3/99 ..../.. ..../.. K312=KT-SP-ADDR-DECODE				
311)	6552	MFSS05	12715	6545	BUTR(MF-SAME-STACK)=H, EXPECTED L 6547 K3/80 K2/20 ..../.. FI; K3=E56,E79,E89,E96; K2=E106,E118				
312)	6553	MFSS06	12725	6547	BUTR(MF-SAME-STACK)=L, EXPECTED H 6545 K3/75 K2/20 K5/5 FI; K3=E13,E47,E56,E63,E69,E79,E89,E96; K2=E62,E84, E106,E118; K5=E63				
313)	6556	TEST512C1	13130	7400	KT-ASPHI(DF)=(16), ASSERTED, F1*DR6*DMO*PS13*-FLOAT 7412 K2/99 ..../.. ..../.. FI: K2=E119 74?? K3/99 ..../.. ..../.. K312=KT-SP-ADDR-DECODE				
314)	6557	TEST507D	12243	7407	BUTM(MASK-PS(T)), MASK-PS(T)=1*1="1" 7406 K2/99 ..../.. ..../.. FI; K2=E78,E101,E105,E112				
315)	6561	TEST500	10968	6131	(NUA SEQUENCING LOGIC ERROR) ???? K1/99 ..../.. ..../.. INTERNAL DCS ERROR				
316)	6563	TEST374	9898	NONE	(NUA SEQUENCING LOGIC/DCS ERROR) ???? K1/99 ..../.. ..../.. INTERNAL DCS ERROR				
317)	6565	TEST512B1	13081	7412	KT-ASPHI(DF)=(06), NEGATED 7400 K3/99 ..../.. ..../.. FI: K3=E45,E47 74?? K3/99 ..../.. ..../.. K312=KT-SP-ADDR-DECODE				
318)	6566	TEST512C2	13157	7412	KT-BSPHI(SF)=(06), NEGATED 7400 K3/99 ..../.. ..../.. FI; K3=E47 7406 K4/99 ..../.. ..../.. FI: K4=E59 74?? K3/99 ..../.. ..../.. K312=KT-SP-ADDR-DECODE				
319)	6567	ERROR624A	15803	0005	MICROBREAK JAMUPP AT (6255) ATTEMPTED; DID NOT OCCUR 6567 K3/99 ..../.. ..../.. FI; K3=E81,E91,E101,E111-E112				

Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/% #2/% #3/%			Test summary - Print reference - Chip information
320)	6571	TEST372A	9606	6222	BUTA(SUBR-A)/BUTA(RETURN)	SEQUENCE: D<14:03> -> RETURN; "1100 1001 0010"=(6222			
					4022	K2/99	.../..	.../..	FI; K2=E92
					4222	K2/99	.../..	.../..	FI; K2=E48, E77, E89, E92-E93, E95, E102, E106, E118
					6000	K2/99	.../..	.../..	FI; K2=E90
					6022	K2/99	.../..	.../..	FI; K2=E3, E6, E12, E18, E24-E25, E30, E33, E39, E45, E51, E54, E60, E66, E69, E75, E81, E89, E92-E93, E96, E101, E107
					6076	K2/99	.../..	.../..	FI; K2=E84, E90
					6200	K6/50	K2/50	.../..	FI; K6=E34; K2=E86
					6202	K2/99	.../..	.../..	FI; K2=E3, E6, E12, E18, E24, E30, E33, E39, E45, E51, E54, E60, E66, E69, E75, E81, E86-E88, E104, E110, E113, E116
					6220	K2/99	.../..	.../..	FI; K2=E6, E9, E12, E18, E24, E26, E30, E33, E39, E45, E51, E54, E60, E66, E69, E75, E81, E86-E88, E104, E110, E114, E116
					6223	K2/99	.../..	.../..	FI; K2=E9, E26, E86, E103
					6226	K2/99	.../..	.../..	FI; K2=E9, E26, E86, E114
					6232	K2/99	.../..	.../..	FI; K2=E9, E26, E86, E113
					6262	K2/99	.../..	.../..	FI; K2=E3, E25, E86, E98
					6322	K2/99	.../..	.../..	FI; K2=E3, E25, E92, E98
					6362	K2/99	.../..	.../..	FI; K2=E98
					6376	K2/70	K7/15	K5/15	FI; K2=E61, E71, E83; K7=E36, E51; K5=E15
					6622	K2/99	.../..	.../..	FI; K2=E42, E77, E92, E107
					7222	K2/99	.../..	.../..	FI; K2=E72, E77, E92, E102
					7700	K2/99	.../..	.../..	FI; K2=E86
321)	6572	TEST373B	9814	7425	EXEC B.M. ROM CODE FOR "FLPADR"	ASSERT TO ASPHI/READ			
					7400	K4/55	K2/35	K3/10	FI; K4=E1, E16, E20, E51, E54, E59-E60, E67, E69, E84, E93, E104, E114; K2=E4, E7, E10, E16, E18, E22, E25, E27-E28, E37, E43, E49, E52, E58, E64, E73; K3=E3, E7
					7402	K3/55	K2/25	K4/15	FI; K3=E1-E20, E22-E26, E45, E51, E66; K2=E4, E10, E16, E22, E27-E28, E31, E36-E37, E43, E49, E52, E58, E64, E67, E69, E73, E79; K4=E13, E15, E51, E54, E62, E69, E113-E114
					7403	K3/95	K4/5	.../..	FI; K3=E1-E19, E22, E24-E26; K4=E29
					7417	K4/80	K2/20	.../..	FI; K4=E59-E60, E84, E93; K2=E31, E69
					7420	K4/99	.../..	.../..	FI; K4=E1, E61, E66, E68, E70, E74-E77, E83, E85, E93-E94, E103-E104, E113-E114
					7426	K2/99	.../..	.../..	FI; K2=E43
					7432	K2/99	.../..	.../..	FI; K2=E4, E16, E22, E27-E28, E37, E43, E52, E58, E67, E79
					7433	K4/99	.../..	.../..	FI; K4=E54, E60, E66-E68, E70, E75-E76, E83, E103
					7434	K7/30	K6/30	K2/30	K3/15 FI; K7=E41, E66; K6=E78, E90; K2=E18, E75; K3=E7
322)	6573	TEST361A	8827	7434	SR=(052525), D(C)="1"; SR/XMUX=(100125) (FLTPT)				
					7400	K4/96	K3/5	.../..	FI; K4=E27, E29, E37, E44, E46, E54, E69, E78; K3=E92
					7402	K4/99	.../..	.../..	FI; K4=E46
					7417	K4/99	.../..	.../..	FI; K4=E27

[Continued]

Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/% #2/% #3/%			Test summary - Print reference - Chip information
	6573	TEST361A	[Continued]			SR=(052525), D(C)="1"; SR/XMUX=(100125) (FLTPT)			
						7420 K4/99 ... .. FI; K4=E37,E44			
						7425 K4/99 ... .. FI; K4=E18			
						7426 K4/99 ... .. FI; K4=E37			
323)	6574	TEST7100	16265	7402	DATO, I/O PAGE, 16. BIT I/O PAGE MODE; SERVICE=(003740)				
						7400 K7/50 K6/30 K5/20 FI; K7=E44,E68,E71; K6=E51,E53-E54,E59,E105; K5=E7,E27,E30,E95			
324)	6575	TEST710E	16294	7402	CLEAR JAM ERRORS/ODD ADDR; JAM=(001000)				
						7401 K7/99 ... .. FI; K7=E25,E59			
325)	6576	TEST51202	13212	7405	KT-BSPHI(DF)=(02), NEGATED, -DR6				
						7412 K3/99 ... .. FI; K3=E47,E79,E93			
						74?? K3/99 ... .. K3I2=KT-SP-ADDR-DECODE			
326)	6577	TEST512E1	13234	7400	KT-ASPFI(DF)=(16), ASSERTED, F1*DR6*-DMO*PS15				
						7405 K3/99 ... .. FI; K3=E45			
						7412 K3/99 ... .. FI; K3=E47			
						74?? K3/99 ... .. K3I2=KT-SP-ADDR-DECODE			
327)	6600	TEST5100	12438	7406	BUTM(MASK-PS(T)), MASK-PS(T)=1*0="0"				
						7407 K2/99 ... .. FI; K2=E72,E101,E105			
328)	6601	TEST5030A	11359	7406	BUTM(FLTPT-INSTR), IR=(125122), -FLTPT				
						7407 K2/99 ... .. FI; K2=E53,E112			
329)	6603	TEST510A	12363	7407	BUTR(BG-SERVICE-L), NEGATED WHEN PS PRIO<7:5>=(7)				
						7403 K7/80 K3/15 K2/5 FI; K7=E3,E5,E28,E76; K3=E50,E74; K2=E100			
330)	6604	TEST510E	12497	7403	BUTR(SERVICE-H), ASSERTED="1" WHEN FLAG<7>H="1"				
						7402 K2/99 ... .. FI; K2=E78,E100			
331)	6605	TEST506E	12099	7403	BUTR(INTR-HIGH), INTR-HIGH-H="1"; INTERNAL SERVICE CLEAR				
						4747 K6/99 ... .. FI; K6=E78			
						7401 K7/85 K3/10 K2/5 FI; K7=E37,E43,E48,E50,E57-E58,E66-E67,E76-E77,E84; K3=E44,E50; K2=E72,E100			
332)	6607	TEST511A	12755	NONE	(NUA SEQUENCING LOGIC ERROR)				
						???? K1/99 ... .. INTERNAL DCS ERROR			
333)	6610	TEST510F	12524	7402	BUTR(SERVICE-H), NEGATED="0", NO INPUTS ACTIVE				
						7403 K7/80 K2/15 K3/5 FI; K7=E3,E7-E8,E11,E16,E27; K2=E38,E78,E100; K3=E64			
334)	6611	TEST507C	12221	7406	BUTM(PS03), PS<03>H="0"				
						7407 K2/99 ... .. FI; K2=E112			
						740? K2/80 K3/20 ... .. K2I0=MULTIPLE-BUT; K306=PS(CC)			

-----  
Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS  
-----

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/% #2/% #3/%	Test summary - Print reference - Chip information
335)	6613	TEST507A	12144	7402		LOAD/READ PS=(140125) PS-PORT PROC-MUX; UCON FUNCTION 7400 K2/50 K3/45 K4/5 K7/5 FI; K2=E102-E103,E105,E111,E113-E114,E116, E118; K3=E35,E71-E73,E83; K4=E63,E72; K7=E3 7401 K3/45 K2/45 K4/10 FI; K3=E28,E47,E54,E63; K2=E108,E118; K4=E72	
336)	6615	TEST504A	11578	4777		NVA SEQUENCING ERROR, JAMUPP ERROR; BUTA(DIAGNOSE) TO B.M. 'LOADNZW4/LOADNZW5' 4744 K2/99 ..../.. ..../.. FI; K2=E54 4757 K2/99 ..../.. ..../.. FI; K2=E54	
337)	6616	TEST510C	12418	7403		BUTR(SERVICE-H), ASSERTED="1" WHEN INTR-HIGH-H="0" 7345 K3/99 ..../.. ..../.. FI; K3=E76 7402 K2/75 K3/25 ..../.. FI; K2=E100; K3=E64	
338)	6617	TEST5100A	12475	7407		BUTM(D00), D=(040401) 7406 K2/99 ..../.. ..../.. K210=MULTIPLE-BUT	
339)	6621	TEST506A	11952	7402		LOAD/READ PS=(030252) PS/PROC-MUX; BUTA(CLR-...) SETS BUSDIN=EMIT; UCON FCN 7400 K2/50 K3/40 K7/10 K5/5 FI; K2=E61-E62,E70-E71,E82-E83,E98 E102-E103,E107-E108,E111-E114,E116-E118; K3=E47, E52,E57,E62-E63,E72-E73,E83,E89; K7=E3,E27,E42, E44,E48; K5=E13,E54,E84 7401 K2/90 K3/10 ..../.. FI; K2=E62,E82,E117; K3=E57 7403 K2/99 ..../.. ..../.. FI; K2=E108	
340)	6623	TEST505A	11847	7402		SET ALL FLAGS, CHECK BUTA(CLR-FLAG-RES-UCON) CLEARS SHORT TERM FLAGS 7400 K2/99 ..../.. ..../.. FI; K2=E101,E105,E119	
341)	6625	TEST533A	13528	7402		SHIFTER, AMUX/D[HI]#D[LO]=(052652), D=(0)(052652) 7400 K4/99 ..../.. ..../.. FI; K4=E29,E51,E54,E60-E62,E66-E69,E77,E83-E85, E93-E94,E103-E104,E113-E114	
342)	6627	TEST520A	13322	7403		INSTR-BRANCH-L="H", IR=(101004)=BHI, PS(NZVC)=(04) 7402 K3/99 ..../.. ..../.. FI; K3=E64,E83 7407 K3/99 ..../.. ..../.. K2=PS(CC)-BRANCH(E64,72-73,83)	
343)	6630	TEST503B	11318	7406		BUTM(FLAG7), FLAG<7>H="0" 7407 K2/85 K3/15 ..../.. FI; K2=E112; K3=E54	
344)	6631	TEST003	2673	5525		NVA SEQUENCING, PAGE (6) -> (5), UBF=(35) 4525 K2/80 K4/10 K3/10 FI; K2=E35,E47,E58,E62,E64,E67,E72-E73,E78,E91; K4=E3; K3=E111 5425 K3/50 K2/50 ..../.. FI; K3=E59,E91; K2=E1,E7 5520 K2/99 ..../.. ..../.. FI; K2=E76 5544 K2/99 ..../.. ..../.. FI; K2=E90 5546 K2/99 ..../.. ..../.. FI; K2=E71	

[Continued]

Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/%% #2/%% #3/%%	Test summary - Print reference - Chip information
	6631	TEST003	[Continued]				NUA SEQUENCING, PAGE (6) -> (5), UBF=(35) 6525 K2/99 ..../.. ..../.. FI; K2=E46,E59,E65,E68,E74,E80 7525 K2/99 ..../.. ..../.. FI; K2=E35,E47
345)	6632	TEST503C	11338	7402			BUTR(FLTPT-PROC-H), FLTPT-PROC-H="0" 4402 K2/99 ..../.. ..../.. FI; K2=E34 7403 K3/50 K2/50 ..../.. FI; K3=E64; K2=E119 7406 K2/60 K3/40 ..../.. FI; K2=E2,E8,E66,E69,E75,E81; K3=E21,E81
346)	6633	TEST533B	13569	7402			SHIFTER, AMUX/D(HI)#D(L0)=(125125), D=(0)(125125) 6367 K3/99 ..../.. ..../.. FI; K3=E38 7400 K4/99 ..../.. ..../.. FI; K4=E51,E54,E59-E62,E66-E69,E74-E76,E83-E85, E93-E94,E103-E104,E113 7401 K4/99 ..../.. ..../.. FI; K4=E75-E76 7403 K4/99 ..../.. ..../.. FI; K4=E51,E54,E62,E69,E84,E103,E114
347)	6634	TEST534B	13650	7402			SHIFTER, AMUX/D(L0)#D(HI)=(052652), D=(0)(125125) 7400 K4/99 ..../.. ..../.. FI; K4=E93-E94,E103-E104,E113 7403 K4/99 ..../.. ..../.. FI; K4=E114
348)	6635	TEST534D	13725	7402			SHIFTER, AMUX/8#D(C)#D(HI)=(000377), D=(0)(177777) 7400 K4/75 K2/20 K3/10 FI; K4=E59,E93,E104,E113-E114; K2=E65,E68; K3=E75 7403 K4/99 ..../.. ..../.. FI; K4=E114
349)	6636	TEST534F	13805	7402			SHIFTER#COUNTER, AMUX/COUNT#D(HI)=(052652), CTR=(125), D=(0)(125000) 7400 K4/99 ..../.. ..../.. FI; K4=E87,E90-E91,E93,E104,E113 7401 K4/99 ..../.. ..../.. FI; K4=E114 7403 K4/99 ..../.. ..../.. FI; K4=E8,E91,E114
350)	6637	TEST535B	13926	7402			SHIFTER, BMUX/4#D(C)#AMUX=(002645), D=(0)(055132) 7400 K4/99 ..../.. ..../.. FI; K4=E69,E83-E85 7403 K4/99 ..../.. ..../.. FI; K4=E69
351)	6640	TEST536B	14019	7402			SHIFTER, CMUX/2#D(C)#BMUX=(034343), D=(0)(161616) 7400 K4/99 ..../.. ..../.. FI; K4=E54,E60-E61,E67,E75 7401 K5/99 ..../.. ..../.. FI; K5=E105 7403 K4/99 ..../.. ..../.. FI; K4=E54
352)	6641	TEST536D	14103	7402			SHIFTER, CMUX/D(C)#BMUX=(063146), D=(0)(146314) 7400 K4/90 K3/15 ..../.. FI; K4=E60-E62,E67-E68,E75-E76; K3=E42 7403 K4/99 ..../.. ..../.. FI; K4=E54
353)	6642	TEST536F	14196	7402			SHIFTER, CMUX/BMUX#SENMUX=(063146), D/SR=(0)(031463)(077777) 7400 K4/99 ..../.. ..../.. FI; K4=E60-E62,E67-E68,E75-E77 7403 K4/99 ..../.. ..../.. FI; K4=E54
354)	6643	NEWCTR537A	14291	7370			(NUA SEQUENCING LOGIC ERROR) ???? K1/99 ..../.. ..../.. INTERNAL DCS ERROR

Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###)	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/% #2/% #3/%			Test summary - Print reference - Chip information
355)	6644	TEST710B	16212	7402	DATO,	000	ADD	ERR;	JAM=(101004)
					7400	K7/45	K5/45	K6/10	FI; K7=E61, E65-E66, E80, E82, E84; K5=E13, E43-E45, E64; K6=E76
					7401	K6/80	K7/20	.../..	FI; K6=E71, E80-E81, E99; K7=E80, E82
356)	6645	TEST710C	16236	7402	DATO,	I/O	PAGE,	18.	BIT I/O PAGE MODE; SERVICE=(002340)
					7400	K7/65	K5/15	K6/15	K4/5 K3/5 FI; K7=E14, E27, E32, E44, E64-E65, E68, E70, E73, E76-E77, E79-E82, E84, E88; K5=E13-E14, E39, E67, E93; K6=E43, E46-E47, E53-E54, E59, E81; K4=E20, E65; K3=E34
					7401	K2/65	K7/15	K6/15	FI; K2=E1, E82; K7=E80; K6=E43
					7403	K7/99	.../..	.../..	FI; K7=E80
357)	6646	TEST503K	11531	7407	BUTM(FLTP	-INSTR),	IR=(175252),	FLTP	
					7406	K2/50	K3/50	.../..	K210=MULTIPLE-BUT; K304=FP-DECODE
358)	6647	TEST507F	12286	7407	CHECK	PS(C)=PS(00)="1",	D(C) INPUTS	PS(C),	CIN/PS(C), CIN/D(C)=PS(C) ALL SET
					7400	K3/50	K2/50	.../..	FI; K3=E93; K2=E112
					7401	K4/99	.../..	.../..	FI; K4=E86
					7406	K3/99	.../..	.../..	FI; K3=E35, E46
359)	6653	TEST536E	14154	7402	SHIFTER,	CMUX/BMUX#	SENMUX=(114631),	D/SR=(0)(146314)(100000)	
					7400	K4/99	.../..	.../..	FI; K4=E54, E60-E62, E67-E68, E75-E77, E80
360)	6655	TEST536C	14067	7402	SHIFTER,	CMUX/D(C)#	BMUX=(114631),	D=(1)(031463)	
					7400	K4/95	K6/5	.../..	FI; K4=E54, E60-E62, E67-E68, E70, E75-E76; K6=E108
361)	6657	TEST536A	13973	7402	SHIFTER,	CMUX/2*D(C)#	BMUX=(143434),	D=(1)(016161)	
					7400	K4/99	.../..	.../..	FI; K4=E60-E62, E68, E76, E84
					7401	K4/99	.../..	.../..	FI; K4=E54, E60-E62, E67-E68, E70, E75-E76
362)	6661	TEST535A	13879	7402	SHIFTER,	BMUX/4*D(C)#	AMUX=(175132),	D=(1)(122645)	
					7400	K4/99	.../..	.../..	FI; K4=E3, E60, E69, E75, E77, E83-E85
363)	6663	TEST534E	13765	7402	SHIFTER#	COUNTER,	AMUX/COUNT#	D(L0)=(125125),	CTR=(252), D=(0)(000125)
					7400	K4/96	K2/5	.../..	FI; K4=E87, E90-E91, E93, E104, E113-E114; K2=E55
					7401	K4/80	K6/15	K2/5	FI; K4=E87, E90-E91; K6=E103; K2=E55
					7403	K4/99	.../..	.../..	FI; K4=E91, E114
364)	6665	TEST534C	13691	7402	SHIFTER,	AMUX/8*D(C)#	D(HI)=(177400),	D=(1)(000000)	
					7400	K4/99	.../..	.../..	FI; K4=E67-E68
					7401	K4/99	.../..	.../..	FI; K4=E28-E29, E93, E104, E113-E114
365)	6667	TEST534A	13615	7402	SHIFTER,	AMUX/D(L0)#	D(HI)=(125125),	D=(0)(052652)	
			[Continued]		6356	K2/65	K3/35	.../..	FI; K2=E2, E8-E9, E21, E27, E60, E66, E69, E75, E81;

-----  
 Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS  
 -----

ERROR ###) code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/%% #2/%% #3/%%			Test summary - Print reference - Chip information
6667	TEST534A	[Continued]						SHIFTER, AMUX/DILO) #D(HI)=(125125), D=(0)(052652) 6356 K2/65 K3/35 ... K3=E21-E22,E44,E56,E77 7341 K2/99 ... FI; K2=E34 7343 K3/99 ... FI; K3=E37-E39,E44,E64 7400 K4/90 K3/5 K2/5 FI; K4=E28-E29,E77,E93-E94,E103-E104,E113-E114; K3=E3; K2=E74 7401 K4/99 ... FI; K4=E28-E29,E93,E104,E113-E114
366)	6670 TEST5030	11379	7476					BUT(INSTR1), IR=(175252), CLASS-A(FLTPT), FLAG(4:5)H="10" 7474 K2/85 K7/15 ... FI; K2=E62,E100; K7=E44 7477 K3/99 ... K305=FLAGS/FP-INSTR1
367)	6671 TEST0500	11008	6377					(NUA SEQUENCING LOGIC ERROR) 6173 K2/70 K3/30 ... FI; K2=E2-E3,E6-E7,E12,E15,E18,E24,E30,E36; K3=E23, E58,E70 7376 K3/99 ... FI; K3=E58 7777 K1/95 ... INTERNAL DCS ERROR
368)	6672 TEST504F	11742	7407					BUTM(EXFLAG2), EXFLAG(2)H="1" 7406 K2/99 ... K210=FLAGS
369)	6673 TEST500A	11070	7434					SP REWRITE FUNCTION WR(A,HI,A) DOESN'T WRITE B-SIDE 7477 K4/99 ... K405=SP-REWRITE-CNTL,K406/7=A/B-SPADS
370)	6674 TEST507E	12265	7401					BUTR(INTR-HIGH), INTR-HIGH-H="0": INTERNAL SERVICE SET 7403 K2/65 K3/35 ... FI; K2=E72; K3=E44
371)	6675 TEST510B	12396	7407					BUTR(BG+FP/SERVICE), NEGATED="0" WHEN PS PRIO(7:5)=(7) 7277 K2/99 ... FI; K2=E43 7377 K2/65 K3/35 ... FI; K2=E16-E18,E44-E45,E67-E69; K3=E3,E7-E8,E25 7417 K3/99 ... FI; K3=E50,E76
372)	6677 TEST503A	11238	4332					NUA SEQUENCING ERROR; BUTA(DIAGNOSE) TO B.M. 'LOADNZW4', 'LOADNZW5' 0000 K2/99 ... FI; K2=E11,E17,E23,E29,E32,E38,E44,E50,E53-E54,E57, E59,E63,E65,E68,E72,E74,E78,E80,E92-E93 0127 K2/75 K3/25 ... FI; K2=E35,E47,E72,E78; K3=E111 0332 K2/99 ... FI; K2=E63,E80 0732 K2/99 ... FI; K2=E5,E11,E15,E17,E23,E29,E32,E38,E44,E50,E53, E59,E65,E68,E74,E80 4032 K2/99 ... FI; K2=E30 4126 K2/99 ... FI; K2=E6,E12,E18,E24,E30,E33,E39,E45,E51,E54,E60, E66,E69,E75,E81 4322 K2/99 ... FI; K2=E81 4330 K2/99 ... FI; K2=E81 4331 K2/99 ... FI; K2=E81 4335 K2/99 ... FI; K2=E81 4336 K2/99 ... FI; K2=E80 [Continued] 4337 K2/99 ... FI; K2=E57,E78

Module codes: K1/DCS K2/WORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

ERROR	code	Symbolic label	Line number	ENUA	TNUA	->Module sequence->			Test summary - Print reference - Chip information
						#1/%	#2/%	#3/%	
	6677	TEST503A	[Continued]						MUA SEQUENCING ERROR; BUTA(DIAGNOSE) TO B.M. 'LOADNZW4', 'LOADNZW5'
					4352	K2/99	.../...	.../...	FI; K2=E30
					4372	K2/99	.../...	.../...	FI; K2=E30
					4375	K2/99	.../...	.../...	FI; K2=E92
					4732	K2/99	.../...	.../...	FI; K2=E5, E11, E17, E21, E23, E29, E32, E38, E44, E50, E53, E59, E65, E68, E74, E80
					4747	K3/99	.../...	.../...	FI; K3=E1-E20, E24-E26
					7005	K2/99	.../...	.../...	FI; K2=E5, E11, E15, E17, E29, E32, E50, E53-E54, E63, E65, E68, E74, E80
373)	6700	TEST503E	11407	7402					EXFLAGS(2:1)H="01", READ THRU CUA-PORT, PROC-MUX
					7400	K2/85	K3/15	.../...	FI; K2=E92, E101, E103, E106, E108, E111-E112, E119; K3=E48
					7403	K2/99	.../...	.../...	FI; K2=E108
374)	6701	TEST372B	9658	5555					BUTA(SUBR-A)/BUTA(RETURN) SEQUENCE: D(14:03) -> RETURN; "1011 0110 1101"=(5555
					4555	K2/99	.../...	.../...	FI; K2=E77, E89, E92, E95, E102, E106
					5155	K2/99	.../...	.../...	FI; K2=E6, E12, E18, E24, E30, E33, E39, E42, E45, E51, E54, E60, E66, E69, E75, E77, E81, E89, E92-E93, E96, E106-E107
					5355	K2/99	.../...	.../...	FI; K2=E92, E107
					5455	K2/99	.../...	.../...	FI; K2=E3, E6, E12, E18, E24-E25, E30, E33, E39, E45, E51, E54, E60, E66, E69, E75, E81, E89, E92-E93, E96, E98, E101
					5515	K2/99	.../...	.../...	FI; K2=E3, E6, E12, E18, E24-E25, E30, E33, E39, E45, E51, E54, E60, E66, E69, E75, E81, E86-E87, E89, E96, E98, E101
					5557	K2/99	.../...	.../...	FI; K2=E9, E26, E86, E114
					7555	K2/99	.../...	.../...	FI; K2=E72, E77, E92, E102
375)	6702	TEST504G	11762	7406					BUTM(EXFLAG1), EXFLAG(1)H="0"
					7407	K2/99	.../...	.../...	FI; K2=E112
376)	6704	TEST410B	10814	7434					IR PATTERN LOOP: D(14:00)=ZERO NEGATED CORRECT # TIMES
					7400	K4/65	K3/35	.../...	FI; K4=E24, E33; K3=E84
377)	6705	TEST410E	10911	7402					IR PATTERN LOOP: "BYTE-CONSTANT/SECOND-1-OR-2" ASSERTED CORRECT # TIMES
					7400	K4/99	.../...	.../...	FI; K4=E33
					7401	K3/99	.../...	.../...	FI; K3=E49, E60
378)	6706	TEST504H	11783	7406					BUTR(FPS05), FPS(05)="0"
					7407	K3/99	.../...	.../...	FI; K3=E54
379)	6707	TEST410C	10844	7402					IR PATTERN LOOP: "BYTE-H" ASSERTED/NEGATED CORRECT # TIMES
					7400	K3/65	K5/35	.../...	FI; K3=E85; K5=E63
					7401	K6/85	K7/15	.../...	FI; K6=E60, E78-E80; K7=E10
380)	6710	TEST503F	11433	7406					BUTM(EXFLAG2), EXFLAG(2)H="0"
					7407	K2/99	.../...	.../...	FI; K2=E112

-----  
 Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS



###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/% #2/% #3/%			Test summary - Print reference - Chip information
381)	6711	TEST410A	10790	7422					IR PATTERN LOOP: D<14:00>=ZERO ONLY ASSERTED 2. TIMES 7420 K2/99 ..../.. ..../.. FI; K2=E40,E88-E89 7421 K4/65 K3/35 ..../.. FI; K4=E99; K3=E66 7424 K2/99 ..../.. ..../.. FI; K2=E77,E88-E89,E95,E99,E101,E106 7436 K2/99 ..../.. ..../.. FI; K2=E88-E89,E106
382)	6712	TEST505B	11891	7407					EXFLAG<2> NOT CLEARED BY BUTA(CLR-FLAG-RES-UCON) 7406 K2/99 ..../.. ..../.. K207=ACTIVE-BUT,K210=FLAGS
383)	6713	TEST371B	9538	7401					SR/NOP HAS NO EFFECT ON GUARD="0100" 74?? K4/99 ..../.. ..../.. K408=SR/RES/GUARD-LOGIC
384)	6714	TESTA503A	11292	7402					LOAD/READ FLAGS#FPS=(125252) FLAG-PORT PROC-MUX; UCON FUNCTION 4777 K3/99 ..../.. ..../.. FI; K3=E41,E52,E111 7400 K3/60 K2/40 K7/5 K5/5 FI; K3=E1-E20,E22-E27,E29,E33,E35,E41,E47, E51,E54,E94; K2=E4,E10,E16,E22,E28,E31,E36-E37, E43,E49,E52,E58,E62,E64,E67,E73,E79,E82,E98-E99, E101-E103,E106-E108,E110-E111,E113-E114,E117, E119; K7=E42,E44; K5=E84 7401 K2/55 K7/20 K5/15 K3/10 FI; K2=E62,E71,E82-E83,E100-E101,E106,E108, E117; K7=E42,E44,E51,E79; K5=E15,E59,E67,E70; K3=E57,E61 7403 K2/65 K3/25 K4/10 FI; K2=E95,E106,E108,E118; K3=E41; K4=E33
385)	6715	TEST371A	9490	7434					SR/NOP HAS NO EFFECT ON SR=(052525) 7400 K4/99 ..../.. ..../.. FI; K4=E36,E80 7433 K4/99 ..../.. ..../.. FI; K4=E35
386)	6716	TEST504I	11804	7403					BUTM(FLTPT-FD-H), FLTPT-FD-H="1" 7402 K3/50 K2/50 ..../.. FI; K3=E64,E86; K2=E119
387)	6717	TEST3700	9467	7401					SR/LEFT-GUARD/ENB SHIFTS GUARD LEFT, GD<3:0>="0100" AFTER 7400 K4/99 ..../.. ..../.. FI; K4=E53
388)	6720	TEST503G	11453	7407					BUTM(EXFLAG1), EXFLAG<1>H="1" 4756 K3/99 ..../.. ..../.. FI; K3=E61 7406 K2/65 K4/15 K3/15 FI; K2=E112; K4=E86; K3=E54
389)	6721	TEST370C	9429	7434					SR/LEFT-GUARD/ENB, GD<3>="1"; SR=(052525) AFTER 74?? K4/99 ..../.. ..../.. K408=SR/RES/GUARD-LOGIC
390)	6722	TEST505C	11912	7406					EXFLAG<1> CLEARED BY BUTA(CLR-FLAG-RES-UCON) 7406 K2/99 ..../.. ..../.. K207=ACTIVE-BUT,K210=FLAGS

Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/% #2/% #3/%	Test summary - Print reference - Chip information
391)	6723	TEST370B	9405	7402		SR/LEFT-GUARD/ENB SHIFTS GUARD LEFT, GD<3:0>="1010" AFTER 7400 K4/99 ..../.. ..../.. FI; K4=E30 7401 K2/99 ..../.. ..../.. FI; K2=E83	
392)	6724	TEST507B	12201	7403		BUTR(PS15), PS<15>H="1" 7402 K5/99 ..../.. ..../.. FI; K5=E54	
393)	6725	TEST370A	9360	7434		SR LEFT 1, SR=(125252) AFTER; GUARD/ENB="1010" AFTER 7400 K2/85 K4/10 K7/5 FI; K2=E46,E55,E61,E83,E105; K4=E80,E87; K7=E28 7421 K4/99 ..../.. ..../.. FI; K4=E80,E86	
394)	6726	TEST500C	11100	7402		IR PATTERN LOOP: 'OVERLAP-L' ASSERTED TO ASPLO CORRECT # TIMES 7400 K3/80 K4/10 K7/5 K2/5 FI; K3=E46,E59,E80,E84,E87,E89,E116-E117; K4=E25,E96,E99; K7=E69; K2=E55,E105 7401 K3/99 ..../.. ..../.. FI; K3=E46,E57,E59,E80,E84,E87,E89,E121	
395)	6727	TEST41J0	10879	7402		IR PATTERN LOOP: "BYTE-CONSTANT/FIRST-1-GA-2" ASSERTED CORRECT # TIMES 7400 K3/90 K5/5 K4/5 FI; K3=E50,E59-E60,E79,E89,E116-E117; K5=E63,E66; K4=E15 7401 K3/96 K4/5 ..../.. FI; K3=E32,E36,E46,E49-E50,E59-E60,E79,E89; K4=E38	
396)	6730	TEST504B	11647	7407		BUTM(FLAG7), FLAG<7>H="1" 7406 K2/99 ..../.. ..../.. FI; K2=E106	
397)	6731	TEST367R	9317	7401		SR/LOAD-GUARD/DISABLED SETUP BY DUTA(CLR-RES), GUARD NOT ALTERED 74?? K4/99 ..../.. ..../.. K406=SR/RES/GUARD-LOGIC	
398)	6732	TEST503H	11474	7407		BUTR(FPS05), FPS<05>H="1" 7406 K2/99 ..../.. ..../.. FI; K2=E7,E13,E25,E110,E117	
399)	6733	TEST366C	9278	7434		SR LEFT 1, D[C]="1"; SR=(025251) AFTER; (GUARD/DIS) 7400 K4/99 ..../.. ..../.. FI; K4=E36 7402 K4/99 ..../.. ..../.. FI; K4=E45 7420 K4/99 ..../.. ..../.. FI; K4=E35 7421 K4/99 ..../.. ..../.. FI; K4=E52,E71,E86	
400)	6734	TEST506D	12076	7406		BUTM(MASK-PS[T]), MASK-PS[T]=0*1="0" 7407 K2/99 ..../.. ..../.. FI; K2=E72,E78,E105,E112	
401)	6735	TEST366B	9256	7401		CHECK GUARD NOT ALTERED ON SR/LEFT-GUARD/DISABLED SHIFT 7402 K4/99 ..../.. ..../.. FI; K4=E30,E80,E86	
402)	6736	TEST500E	11164	7402		IR PATTERN LOOP: BUTR(PREFETCH[0]H) ASSERTED CORRECT # OF TIMES 7400 K3/99 ..../.. ..../.. FI; K3=E32,E34,E38,E44,E48,E80 7401 K3/90 K2/10 ..../.. FI; K3=E32,E38,E44,E48,E80; K2=E105	

-----  
Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS  
-----

###	ERROR code	Symbolic label	Line number	EMUA	TNUA	->Module sequence-> #1/% #2/% #3/%			Test summary - Print reference - Chip information
403)	6737	TEST366A	9209	7434	SR LEFT 1, D<0>="0"; SR=(112524) AFTER; (GUARD/DIS)	7005	K4/99	.../...	FI; K4=E80
					7400	K4/99	.../...	FI; K4=E45	
					7421	K4/99	.../...	FI; K4=E52,E71,E86	
404)	6740	TEST504C	11667	7403	BUTR(FLTPT-PROC-H), FLTPT-PROC-H="1"	7402	K2/65	K3/35	.../... FI; K2=E101,E106,E119; K3=E64
405)	6741	TEST365B	9187	7401	CHECK GUARD NOT ALTERED ON SR/RIGHT-GUARD/DISABLED SHIFT	7402	K4/99	.../...	FI; K4=E30,E53,E80
406)	6742	TEST503I	11495	7402	BUTR(FLTPT-FD-H), FLTPT-FD-H="0"	7403	K3/99	.../...	FI; K3=E64
					7756	K2/99	.../...	FI; K2=E40	
407)	6743	TEST365A	9146	7434	SR RIGHT 1, D<00>="1"; SR=(145252) AFTER	7400	K4/99	.../...	FI; K4=E10,E17,E36,E45
					7420	K4/99	.../...	FI; K4=E52	
408)	6744	TEST506C	12054	7407	BUTM(PS03), PS<03>H="1"	7406	K2/80	K3/20	.../... K210=MULTIPLE-BUT; K306=PS[CC]
409)	6745	TEST364B	9119	7401	BUT(G03-2), GD<3:0>="0101" AFTER SR/RIGHT, SR<00>="0"	74??	K4/99	.../...	K408=SR/RES/GUARD-LOGIC
410)	6746	TEST5000	11136	7402	IR PATTERN LOOP: 'OVERLAP-L' ASSERTED SAME # TIMES TO ASPLO AND BSPLO	7401	K4/99	.../...	FI; K4=E99
411)	6747	TEST364A	9091	7402	BUT(G03-2), GD<3:0>="1010" AFTER SR/RIGHT, SR<00>="1"	74??	K4/99	.../...	K408=SR/RES/GUARD-LOGIC
412)	6750	TEST504D	11688	7475	BUT(INSTR1), IR=(172525), CLASS-A(FLTPT), FLAG<4:5>H="01"	747?	K3/99	.../...	K305=FLAGS/FP-INSTR1
413)	6751	TEST363B	9069	7401	BUT(G03-2), GD<3:0>="0100" AFTER SR/RIGHT, SR<00>="0"	7400	K3/60	K4/40	.../... FI; K3=E54; K4=E53
					7403	K4/99	.../...	FI; K4=E53	
414)	6752	TEST534G	13844	7402	B.M. COUNTER CLEARED TO ZEROS BY BUTA(LAST)	7400	K4/65	K2/35	.../... FI; K4=E87,E90-E91; K2=E55
					7637	K2/70	K3/30	.../...	FI; K2=E22-E24,E49-E51,E73-E75; K3=E6,E9,E13,E19
415)	6753	TEST363A	9033	7434	SR RIGHT 1, D<00>="0"; SR=(052525) AFTER	7400	K4/99	.../...	FI; K4=E10,E17,E24,E45,E74,E85,E94
			[Continued]		7410	K4/99	.../...	FI; K4=E47	

-----  
Module codes: K1/DCS K2/UWORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

###	ERROR code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/%% #2/%% #3/%%			Test summary - Print reference - Chip information
	6753	TEST363A	[Continued]			SR RIGHT 1, D<00>="0"; SR=(052525) AFTER			
						7417 K4/99 .....			FI; K4=E36
						7420 K4/99 .....			FI; K4=E52
						7426 K4/99 .....			FI; K4=E35
416)	6754	TEST506B	12034	7402	BUTR(PS15), PS<15>H="0"				
						7403 K3/99 .....			FI; K3=E54
417)	6755	TEST362E	9010	7402	BUT(GD3-2), GD<3:0>="1000" AFTER SR/RIGHT, SR<00>="1"				
						7400 K4/80 K3/15 K2/5			FI; K4=E26, E28-E30, E47, E53, E55, E71, E80, E86; K3=E1-E3, E12-E13, E56; K2=E39, E54
418)	6756	TEST503J	11513	7406	BUTM(D00), D=(000000)				
						7407 K2/99 .....			FI; K2=E112
419)	6757	TEST362D	8989	7405	BUTR(SR1-0#COUNT), SR<1:0>="10"				
						4747 K2/99 .....			FI; K2=E46, E55
						74?? K4/99 .....			K408=SR/RES/GUARD-LOGIC
420)	6760	TEST504E	11716	7402	EXFLAGS<2:1>H="10", READ THRU CUA-PORT, PROC-MUX				
						7400 K2/99 .....			FI; K2=E103, E106, E111-E112
421)	6761	TEST702B	16099	7402	LOAD BA<17:16>="10", 18. BIT MODE, READ THRU STATUS-MUX(SERVICE)<9:8>				
						7400 K6/55 K4/25 K7/10 K5/10			FI; K6=E11, E51-E54; K4=E65; K7=E56; K5=E26, E60
422)	6762	TEST520D	13417	7402	INSTR-BRANCH-L="L", IR=(101401)=BLOS, PS(NZVC)=(01)				
						740? K3/99 .....			K2=PS(CC)-BRANCH(E64, 72, 73, 83)
423)	6763	TEST362B	8962	7434	SR=(125252), D(C)="0"; SR/XMUX=(000052) (FLTPT)				
						7400 K4/99 .....			FI; K4=E46, E55, E78
						7420 K4/99 .....			FI; K4=E37
						7421 K4/99 .....			FI; K4=E44
						7424 K4/99 .....			FI; K4=E44
424)	6764	TEST520B	13360	7402	INSTR-BRANCH-L="L", IR=(003000)=BGT, PS(NZVC)=(00)				
						7403 K3/99 .....			FI; K3=E64, E83
						740? K3/99 .....			K2=PS(CC)-BRANCH(E64, 72-73, 83)
425)	6765	TEST362A	8917	7434	SR RIGHT 1, D<00>="1"; SR=(125252) AFTER				
						7400 K4/99 .....			FI; K4=E3, E13, E22, E24, E29, E35-E36, E45, E52, E55, E74, E76, E85, E94, E104
426)	6766	TESTA504A	11621	7402	LOAD/READ FLAG#FPS=(052525) FLAG-PORT PROC-MUX: UCON FUNCTION				
						7400 K2/90 K3/10 .....			FI, K2=E78, E98-E99, E101-E103, E105-E108, E110-E114, E119; K3=E27, E47, E94, E96
						7403 K2/99 .....			FI; K2=E106, E108

-----  
Module codes: K1/DCS K2/UMORD K3/IRDECODE K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

ERROR ###) code	Symbolic label	Line number	ENUA	TNUA	->Module sequence-> #1/%% #2/%% #3/%%	Test summary - Print reference - Chip information	
427) 6767	TEST361E	8894	7400	BUT(GD3-2), GD<3:0>="0000"	AFTER ENABLED, SR/LOAD	7401 K4/80 K2/15 K3/5 FI; K4=E17, E28, E30, E53, E80, E86-E87; K2=E33, E45. E51: K3=E54	
				7402 K4/90 K3/10 ..../..	FI; K4=E26, E30, E53, E87; K3=E56		
				7404 K3/99 ..../..	FI; K3=E74		
428) 6770	TEST701C	15981	7402	LOAD BA<15:00>=(125252), READ THRU BA/KT-ALU/PBA/STATUS-MUX(PBA)		7400 K5/50 K4/15 K6/15 K7/15 FI; K5=E7, E9-E11, E27, E37-E38, E43, E45-E47, E50, E59, E69-E70, E72, E79-E82, E94; K4=E39, E48, E57; K6=E19, E25, E29, E33, E49, E59; K7=E56, E64-E65, E73. E80-E82, E88	
				7401 K7/50 K4/50 ..../..	FI; K7=E88; K4=E29		
				7403 K5/65 K6/20 K7/10	FI; K5=E7, E26-E27, E46-E47; K6=E25, E59; K7=E80		
429) 6771	TEST361D	8873	7403	BUTR(SR1-0#COUNT), SR<1:0>="01"		6377 K2/65 K3/35 ..../..	FI; K2=E4-E5, E58-E59; K3=E1, E16
				7407 K3/99 ..../..	FI; K3=E74		
				74?? K4/99 ..../..	K408=SR/RES/GUARD-LOGIC		
430) 6772	TEST520C	13388	7403	INSTR-BRANCH-L="H", IR=(002416)=BLT, PS(NZVC)=(16)		7402 K3/99 ..../..	FI; K3=E72, E83
				740? K3/99 ..../..	K2=PS[CC]-BRANCH(E64,72-73,83)		
431) 6773	TESTD410	10707	7361	(NUA SEQUENCING LOGIC ERROR)		6777 K2/99 ..../..	FI; K2=E48
				???	K1/99 ..../..	INTERNAL DCS ERROR	
432) 6775	TEST520E	13445	7403	INSTR-BRANCH-L="H", IR=(103406)=BL0, PS(NZVC)=(06)		7402 K3/99 ..../..	FI; K3=E83
				740? K3/99 ..../..	K2=PS[CC]-BRANCH(E64,72-73,83)		

-----  
KD11 Module code: ZERO K1/DCS K2/DCS K3/DCS K4/DATAPATH K5/KTCACHE K6/TIMING K7/STATUS

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DQKUB-80

10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100

PRODUCT CODE: MAINDEC-11-DQKUB-80

PRODUCT NAME: KD11-K Microdiagnostic

MAINTAINER: Diagnostic Engineering

F07

AUTHOR: Don North

DATE CREATED: 18-January-1977

LAST REVISION: 15-June-1977, Version 1.01/  
COPYRIGHT (C) 1976, 1977; DIGITAL EQUIPMENT CORPORATION  
146 MAIN STREET  
MAYNARD, MASSACHUSETTS, USA  
01754 617-897-5111

THIS SOFTWARE IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM, AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE, AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT NOT SUPPLIED BY DIGITAL.

## TABLE OF CONTENTS

59	--	IDENTIFICATION
94	--	REVISION HISTORY
106	--	MICROWORD FIELD DEFINITIONS
129	--	MICROWORD BIT LAYOUT
187	--	A & B & C SCRATCHPAD LAYOUT, DCS SPECIFIC
243	--	MICROWORD FIELD SPECIFICATION
248	--	MICROWORD FIELD FORMAT
260	--	NULL FIELD/MICRO SPECIFICATION
266	--	ALU AND INTERNAL DATA BUS CONTROL
270	--	<ALU>-ALU FUNCTION CONTROL BITS
295	--	<BEN>-B-BUS DATA SOURCE
305	--	<BSEL>-B-BUS SOURCE SELECTION CONTROL
345	--	<REN>-A-BUS DATA SOURCE
355	--	<ASEL>-A-BUS SOURCE SELECTION CONTROL
398	--	<RIF>-ASP, BSP REGISTER IMMEDIATE FIELD
435	--	<COU>-CARRY OUT BIT MUX SELECTION
451	--	CLOCKS
455	--	<WHEN>-D/SR WHEN TO CLOCK
463	--	<CLKD>-ENABLE D-REGISTER CLOCKING
471	--	<CLKSR>-ENABLE SR-REGISTER CLOCKING
479	--	<CLKBA>-ENABLE CLOCKING OF BA-REGISTER
487	--	<SCC>-ENABLE SETTING OF PS CONDITION CODES
498	--	BUS/UCON & CSP-ADDRESS & SHIFT-TREE CONTROL
502	--	BUS/UCON CONTROL
505	--	<BEGIN>-BEGIN BUS/UCON OPERATION
513	--	<SELECT>-SELECT BUS OR UCON
521	--	BUS CONTROL
524	--	<BUSCODE>-BUS CODE ACTION FIELD
540	--	UCON CONTROL
544	--	<FLPGO>-START HOT FLOATING POINT
552	--	<UCON-XFER>-UCON OPERATION
560	--	<UCON-LOAD>-LOAD UCON REGISTER
568	--	CSP ADDRESS SPECIFICATION
571	--	<CSPADDR>-CSP IMMEDIATE ADDRESS
597	--	SHIFT CONTROL
600	--	<SMUX>-SECOND LEVEL OF SHIFT TREE
608	--	<AMUX>-FIRST LEVEL OF SHIFT TREE
625	--	SP REWRITE & REGISTER CLOCKS
629	--	<WRCSF>-WRITE TO CSP
637	--	<MOD>-MODE CONTROL OF FOLLOWING BITS
645	--	SP REWRITE (A,B) CONTROL
649	--	<HILO>-SP HI/LO SELECT
659	--	<WRSEL>-REWRITE ADDRESS SELECT
669	--	<WRSP>-REWRITE A/B SELECT
688	--	REGISTER LOADING
692	--	<LOADRES>-LOAD RESIDUAL CONTROL REGISTER
701	--	<LOADCOUNT>-LOAD COUNTER
711	--	SEQUENCING FIELD
715	--	<UBF>-BUT MICROBRANCH FIELD
719	--	NO BUT

## TABLE OF CONTENTS

722	--	ACTIVE ONLY
739	--	INACTIVE ONLY
810	--	BOTH ACTIVE AND INACTIVE
831	--	<UPF>-MICRO POINTER FIELD
874	--	MISCELLANEOUS FIELDS
878	--	<NEXT-PAGE>-NEW PAGE ADDRESS LOADED DURING BUT(SUBROUTINE)
884	--	<MULTIPLE>-SELECT CODE FOR BUT(MULTIPLE)
898	--	EMIT FIELD - IMMEDIATE DATA FROM MICROWORD
925	--	RETURN ADDRESS - FOR MICROSUBROUTINE CALLS
931	--	UCON SELECTION AND CONTROL FIELDS
934	--	SELECTION
957	--	CONTROL
979	--	BASE MACHINE EXTENSION BITS
1052	--	SPECIAL DCS FIELDS
1056	--	FIELDS USED IN PAGES 4, 5, OR 6 OF DCS
1059	--	<LOAD-DCS-CTR>-LOAD DIAGNOSTIC COUNTER FROM EMITH
1072	--	<CTR>-4 BIT DCS COUNTER VALUE FROM EMIT
1096	--	<LOAD-ENUA-ERRCOD>-LOAD THE ENUA AND ERRCOD REGISTERS
1106	--	<ENUA>-ENUA VALUE FROM EMIT
1113	--	<VERIFY>-VERIFY BIT FOR SELF CHECK TEST
1125	--	FIELDS USED IN PAGE 7 OF DCS EXTENSION
1128	--	<EOP>-SIGNAL SUCCESSFUL END OF PASS
1137	--	<DAD>-DCS CONTROL OF BASE MACHINE EXTENSION DAD BITS
1149	--	FIELDS USED IN ALL PAGES OF DCS EXTENSION
1152	--	<SCOPE>-SCOPE ON ERROR, DIAGNOSTIC BUT
1170	--	MACRO DEFINITIONS
1173	--	PRIMITIVE OPERATIONS
1176	--	TIMING
1210	--	WRITING THE A AND B SCRATCH PADS
1234	--	ASP AND BSP PHYSICAL REGISTER ADDRESSES
1266	--	ASP AND BSP BASE MACHINE FUNCTIONAL REGISTER ADDRESSES
1339	--	ASP AND BSP INDIRECT REGISTER ADDRESSES
1364	--	ASP, BSP INDIRECT ADDRESSING
1377	--	ASP AND BSP DCS SPECIFIC FUNCTIONAL REGISTER ADDRESSES
1401	--	WRITING THE C SCRATCH PAD
1410	--	CSP IMPLIED ADDRESSING
1422	--	CSP DIRECT ADDRESSING
1436	--	SHIFT TREE SPECIFICATION
1440	--	ENABLED ONTO BUS A
1471	--	FIRST TWO LEVELS ONLY (AMUX, BMUX)
1479	--	ALU FUNCTIONS
1500	--	COUT GENERATION
1515	--	CLOCKS
1519	--	BASIC REGISTER CLOCKS (D, SR, BA, CC)
1528	--	REDEFINED FROM SP REWRITE FIELD (RES, COUNTER)
1534	--	RES REGISTER CONTROL VALUES (FROM EMIT)
1548	--	CC CONTROL (FROM EMIT)
1557	--	BUS CONTROL MACROS
1571	--	KT/KJ CONTROL FUNCTIONS
1589	--	UCON CONTROL MACROS



## TABLE OF CONTENTS

1595	--	PROCESSOR UCON CONTROL SETUP
1613	--	DCS/WCS/ECS CONTROL
1621	--	CACHE/KT UCON CONTROL
1641	--	I/O UCON CONTROL
1646	--	BUS CONTROL
1666	--	CONSOLE I-O
1682	--	REMOTE CONSOLE INTERFACE
1692	--	DCS ROM EXTENSION MACROS
1694	--	GENERAL FUNCTIONS
1705	--	DAD<1:0> BIT FUNCTIONS
1714	--	DIAGNOSTIC MODE BUT ENABLES
1732	--	MICROBRANCH FIELD MACROS
1748	--	MISCELLANEOUS
1750	--	OTHER SOURCES ENABLED FOR A-BUS
1756	--	PAGING, RETURN REGISTER
1771	--	ADVANCED OPERATIONS
1775	--	DATA INTO CSP, AT P3 ONLY
1821	--	MISC CONSTANTS INTO ASP, BSP, AT P2-T * P3
1849	--	DATA INTO ASP, BSP, AT P2-T * P3
2055	--	D AND SR (- (BUS-A FCN BUS-B), AT P2-T OR P3-T
2098	--	DIC) GETS SET
2116	--	D-REGISTER (- (BBUS = ABUS), BITWISE, AT P2-T OR P3-T
2153	--	D-REGISTER (- D-REGISTER THRU SHIFT-TREE
2187	--	D (- WHATEVER'S LEFT, AT P2-T OR P3-T
2234	--	SR (- DATA, AT P2 T OR P3 T
2269	--	RES-REG OPERATION MACROS
2278	--	BASE MACHINE COUNTER
2286	--	ENABLE ON BUS-A/B ONLY
2312	--	LOADING BA REGISTER
2325	--	D AND SR TOGETHER
2333	--	UCON FUNCTIONS
2337	--	PROCESSOR UCON FUNCTIONS
2364	--	CACHE/KT UCON FUNCTIONS
2390	--	I-O UCON FUNCTIONS
2406	--	DCS UCON FUNCTIONS
2413	--	CONSOLE UCON FUNCTIONS
2428	--	DBUF UCON FUNCTIONS
2437	--	MULTIPLE UCON FUNCTIONS
2452	--	SPECIFIC MACROS FOR PREFETCH/OVERLAP/SP-INHIBIT TESTS
2479	--	SPECIFIC MACROS FOR BYTE/BYTE CONSTANT/D=ZERO TESTS
2498	--	SUBROUTINE CALL MACROS
2568	--	JAM UPP LOG MACROS
2585	--	----- MICRODIAGNOSTIC CODE -----
2602	--	TEST001-007: NUA SEQUENCING
2775	--	TEST010-011: MICROSUBROUTINE OPERATION
2906	--	INIT REGISTERS, CONSOLE DEFAULT ERROR DISPLAY, REVISION CODE
2996	--	TEST012-050: IR DECODE (INSTRI, INSTR5, FLTPT, RELATED "BUTS")
5139	--	TEST101: D -> DBUF -> IR PATH
5188	--	TEST102-104: TESTING CSP ADDRESS/READ/WRITE FUNCTIONS
5573	--	TEST105: SR CAN LOAD/STORE AS A REGISTER

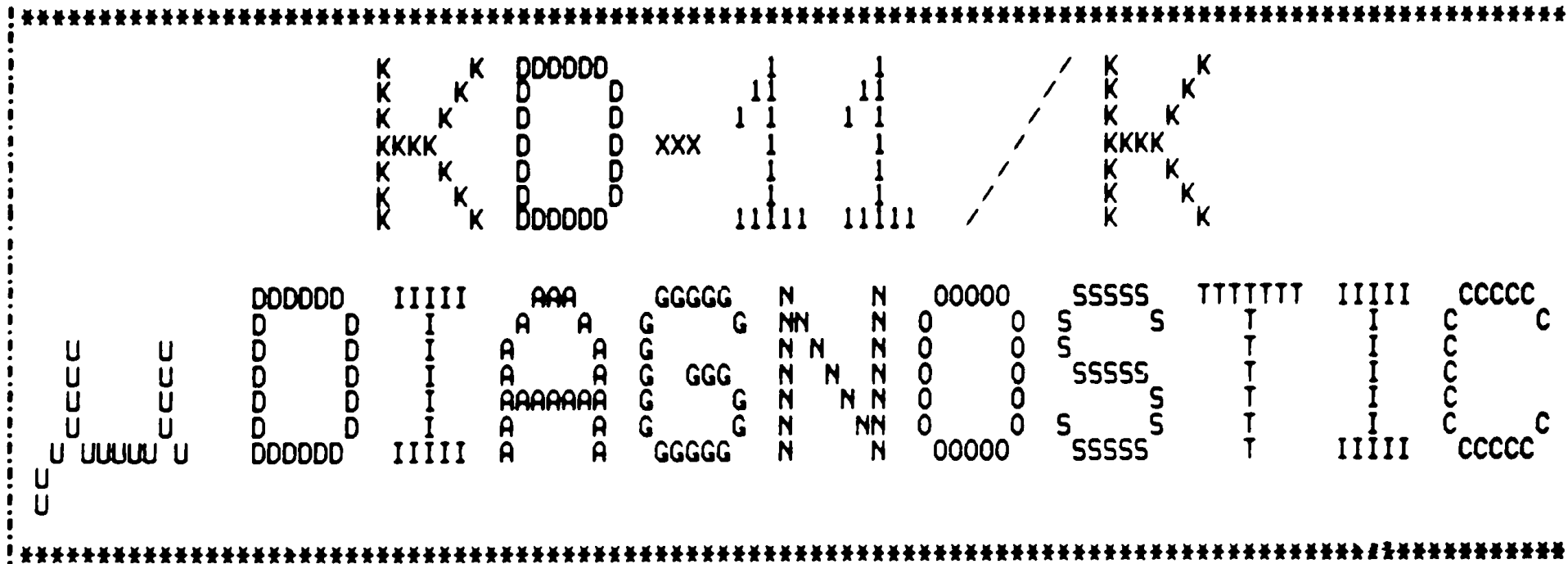
## TABLE OF CONTENTS

5772	--	TEST114-121: ALU LOGIC TESTS / D(C) TESTS
7236	--	TEST130-136: ALU ARITHMETIC FUNCTION/CARRY LOOKAHEAD TESTS
8078	--	TEST320: D(C) SELECTION / COUT07-DOUT07 / D<14:00>=ZERO BIT<00>
8286	--	TEST350-352: ASP/BSP RI/LO ADDRESSING MODES, DATA VALIDITY
8800	--	TEST361-372: TESTING SR, GUARD, RES, AND XMUX
9585	--	TEST372A-372B: TESTING CUA, PROCESSOR MUX, AND BUTA(SUBR A)
9712	--	TEST373: CHECK JAMUPP W/ BUTA(DIAGNOSE), BM EXT BIT FLPADR
9840	--	TEST374: A/B SP REWRITE MODES VERIFICATION
10488	--	TEST375-376: BYTE WRITE TO ASP/BSP LO, SP ADDRS R-OR-1/FLTPT-INHIBIT
10645	--	TEST410: BYTE/BYTE CONSTANT/D=ZERO
10954	--	TEST500: PREFETCH/OVERLAP/SP DEFEAT
11206	--	TEST503-510: PROCESSOR UCON TESTS (FLAGS, FPS, PS, BUTM) & ASSOC LOGIC
12636	--	TEST511: MESS LOGIC TESTS
12868	--	TEST512: KT SRC/DST ADDRESSING LOGIC TESTS
13288	--	TEST520A-520E: TESTING THE "INSTR BRANCH" ROM
13506	--	TEST533-537: SHIFT TREE
14448	--	TEST551: BASE MACHINE DATAPATH COUNTER CAN COUNT
14702	--	TEST610: CONDITION CODE LOGIC
15234	--	TEST620-624: TESTING UBREAK AND JAMUPP
15901	--	TEST701-702: LOAD/READ THE BA, FULL 18. BITS
16132	--	TEST710-722: BUS FUNCTION DECODE, BUS ERROR CONDITIONS
17056	--	TEST730-731: BUS CYCLES TO/FROM MEMORY
17523	--	TEST740: BUS CYCLE MODIFICATION - PREFETCH ALTERATION, OVERLAP YANK
17687	--	TEST761-763: TESTING UNIBUS INTERRUPT SERVICE WITH DL11-W LINE CLOCK
18191	--	END OF PASS CODE
18294	--	VERIFY MODE CODE
18352	--	DCS MICROCODE REVISION NUMBER
18394	--	COMMON SUBROUTINES
18396	--	CONSOLE DISPLAY SUBROUTINE
18445	--	CLEAR I-O / BUS CONTROL / SERVICE AREA STATUS LATCHES SUBR
18491	--	SUBR FOR PUTTING SELECTED PORTIONS OF D[15:00] INTO IR
18570	--	UCON SUBROUTINES (FLAGS, PS, FPS, CUA, SERVICE, JAM, PBA)
18691	--	SUBR FOR LOADING FPS<3:0> (VIA BUTA(DIAGNOSE))
18738	--	SUBR TO COPY D-REGISTER TO DBUF TO IR
18810	--	JAM UPP SERVICE SUBROUTINE
18894	--	MICROBRANCH [BUT] TAKEOFF WORDS
19233	--	MICROBRANCH [BUT] TARGET WORDS
19766	--	END OF KD11-K MICRODIAGNOSTIC CODE

55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108

!.PAGE=====

.TOC \* IDENTIFICATION  
:TITLE KD11-K MICRODIAGNOSTIC



!.PAGE=====

.TOC \* REVISION HISTORY  
.IDENT /VIC1A0/  
REV-NUMBER ::= 000101 !NO BIT15 DURING EXECUTION  
REV-NUMBER-AND-B15 ::= 100101 !BUT SET W/REV. NUMBER AT EOP ONLY

!.PAGE=====

.TOC \* MICROWORD FIELD DEFINITIONS  
! NOTE: THE FOLLOWING ARE THE ASSIGNED RANGES OF THE

109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162

MICROWORD FIELD BIT DEFINITIONS USED IN THIS SOURCE LISTING:

BITS(NUMBER)	WHERE HELD
[47:00] 48/48	MAIN MACHINE ROM, DCS MAIN ROM
[59:48] 12/12 *	MAIN MACHINE ROM EXTENSION, USES 12/12 BITS
[54:48] 7/12 *	DCS 4-BIT ROM EXTENSION, USES 7/12 BITS, [54:48]

\* = NOTE OVERLAP OF BM EXTENSION/DCS EXTENSION BITS. THESE BITS ARE MUTUALLY EXCLUSIVE.

! . PAGE -----

TOC	* MICROWORD BIT LAYOUT	1-EMIT	4-UCON-DATA	6-RETURN	9-RES-BITS	DCS EXTENSION	UCON PROCESSOR CONTROL
	BASE	2-SHIFT-TREE	5-CSP-ADDRESS	7-PAGING	10-MULTIPLE		
	MACHINE CONTROL	3-RESIDUAL-CTL		8-UCON-CONTROL			
U47	ALU3	1-EMITH15	4-UCONL10			DCS-CTR3	PS<3:0>-CLK
U46	ALU2	1-EMITH14	4-UCON-I/O-SEL	6-RETR11	9-HISMUXSELL	DCS-CTR2	
U45	ALU1	1-EMITH13	4-UCON-WCS-SEL	6-RETR10	9-SRS1-L	DCS-CTR1	
U44	ALU0	1-EMITH12	4-UCON-KT-SEL	6-RETR09	9-SRS0-L	DCS-CTRO	
U43	BEN1						UBREAK-CLK
U42	BEN0		4-UCONL09				<NU>
U41	BSEL1	1-EMITM11	4-UCONL08	6-RETR08	9-GUARD-EN-H	ENUA11	<NU>
U40	BSEL0	1-EMITM10	4-UCONL07	6-RETR07		ENUA10	
U39	REN1	1-EMITM09	4-UCONL06	6-RETR06		ENUA09	SEL-HBMUX1L
U38	REN0	1-EMITM08	4-UCONL05	6-RETR05		ENUA08	SEL-HBMUX0L
U37	ASEL1	1-EMITL07		6-RETR04		ENUA07	
U36	ASEL0	1-EMITL06	4-UCON-PROC-SEL	6-RETR03		ENUA06	
U35	RIF2	1-EMITL05	4-UCONM12	6-RETR02		ENUA05	FPS<7:4>-CLK
U34	RIF1	1-EMITL04	4-UCONM11	6-RETR01		ENUA04	PS<7:4>-CLK
U33	RIFO	1-EMITL03	4-UCON-FP-SEL	6-RETR00		ENUA03	
U32	COUT2	1-EMITL02	4-UCONH15	7-NEXT-PAGE2	10-MULT-SEL2	ENUA02	IR-CLOCK
U31	COUT1	1-EMITL01	4-UCONH14	7-NEXT-PAGE1	10-MULT-SEL1	ENUA01	PS<15:12>-CLK
U30	COUT0	1-EMITL00	4-UCONH13	7-NEXT-PAGE0	10-MULT-SELO	ENUA00	FLAG<8:0>-CLK
U29	* WHEN						
U28	* CLK-D						
U27	* CLK-SR						
U26	* CLK-BA						
U25	* SET-CC						
U24	* BEGIN						
U23	SELECT(=0)	2-BMUX	5-CSPADR3	8-SELECT(=1)			
U22	BUSCOD2	2-AMUX2	5-CSPADR2	8-FLPG0			
U21	BUSCOD1	2-AMUX1	5-CSPADR1	8-UCON-XFER			
U20	BUSCOD0	2-AMUX0	5-CSPADR0	8-UCON-LOAD			
U19	* WRCSP						

```

163 !U18 HI/LO 3-LOAD-RES
164 !U17 WRSEL
165 !U16 WRB 3-LOAD-COUNT
166 !U15 WRA
167 !U14 * MOD(=0) 3-MOD(=1)
168 !U13 * UBF4
169 !U12 * UBF3
170 !U11 * UBF2
171 !U10 * UBF1
172 !U09 * UBF0
173 !U08 * UPF8
174 !U07 * UPF7
175 !U06 * UPF6
176 !U05 * UPF5
177 !U04 * UPF4
178 !U03 * UPF3
179 !U02 * UPF2
180 !U01 * UPF1
181 !U00 * UPF0
    
```

( \* = DEDICATED TO THE CORRESPONDING SINGLE FUNCTION )

! .PAGE=====

TOC \* A & B & C SCRATCHPAD LAYOUT, DCS SPECIFIC

THE USE OF THE A, B, & C SCRATCHPADS AS TEMPORARY STORAGE AREAS IS OUTLINED BELOW. NOTE THAT IN MOST CASES, THE REGISTERS EMPLOYED HAVE NO RESEMBLANCE TO ANY SIMILAR NAMED REGISTER IN THE KD11-K BASE MACHINE MICROCODE, EITHER LIVING OR DEAD.

THE BELOW DEFINITIONS ARE -DCS SPECIFIC- ONLY.

OCTAL ADDRESS	ASPHI	ASPLO	BSPHI	BSPLO	CSP
00	. . .	. . .	. . .	. . .	<RETURN>
01	(000000)	. . .	(000000)	. . .	TEMP/(000152)
02	. . .	. . .	. . .	. . .	TEMP/(000125)
03	(177777)	. . .	(177777)	. . .	TEMP
04	. . .	. . .	. . .	. . .	TEMP/(125200)
05	(125252)	. . .	(125252)	. . .	TEMP/(052522)/(170360)
06	. . .	. . .	. . .	. . .	TEMP/(053433)/(007417)
07	(052525)	. . .	(052525)	. . .	TEMP
10	. . .	. . .	. . .	. . .	(052525)

217	:					
218	:	11	(000001)	. . .	(000001)	. . . (125252)
219	:					
220	:	12	. . .	. . .	. . .	(177777)
221	:					
222	:	13	(100000)	. . .	(100000)	. . . (000000)
223	:					
224	:	14	. . .	. . .	. . .	TEMP/(000100)/(057077)
225	:					
226	:	15	(000200)	. . .	(000200)	. . . TEMP/(000077)/(177067)
227	:					
228	:	16	. . .	. . .	. . .	TEMP/(170000)
229	:					
230	:	17	TEMPAHI	TEMPALO	TEMPBHI	TEMPBLO TEMP/(007700)
231	:					
232	:					
233	:					
234	:					
235	:					
236	:					
237	:					
238	:					
239	:					
240	:					
241	:					
242	:					
243	:					
244	:					
245	:					
246	:					
247	:					
248	:					
249	:					
250	:					
251	:					
252	:					
253	:					
254	:					
255	:					
256	:					
257	:					
258	:					
259	:					
260	:					
261	:					
262	:					
263	:					
264	:					
265	:					
266	:					
267	:					
268	:					
269	:					
270	:					

NOTES: (XXXXXX) ::= A REGISTER W/ A CONSTANT VALUE, NAMED:  
 CXXXXXX-A, IF ON THE A-SIDE, OR  
 CXXXXXX-B, IF ON THE B-SIDE

!.PAGE=====

.TOC \* MICROWORD FIELD SPECIFICATION

!.-----  
 .TOC \* MICROWORD FIELD FORMAT

.RADIX 8 ! ALL NUMBERS ARE OCTAL, UNLESS OTHERWISE NOTED

.WIDTH 64R ! MICROWORD IS 64\10 BITS WIDE, BIT <00> IS RIGHTMOST BIT

.BOUNDS [4000:7777] ! ADDRESSES ARE 12 BITS, ON PAGES 4:7

!.OBJECT <15:00>'<31:16>'<47:32>'<63:48> ! OUTPUT FORMAT (DEFAULT SPEC)

!.-----  
 .TOC \* NULL FIELD/MACRO SPECIFICATION

.FIELD N::=<63>  
 .MACRO NULLb::=N/O

.TOC \* ALU AND INTERNAL DATA BUS CONTROL

.TOC \* <ALU>-ALU FUNCTION CONTROL BITS

```

271 !SPECIFIES ALU FUNCTION CODE AND CINDEX SELECT. ALWAYS IN EFFECT.
272 .FIELD ALU::=(47:44)
273 !
274     ---FUNCTION--- LOGIC/ARITH ALUS(3:0) H CINDEX L
275     NOT-A::=00 !COMPLEMENT A, L 0000 -1
276     A-PLUS-B-PLUS-PS(C)::=01 !ADD, L A 1001 -PS(C)
277     NOT-A-AND-B::=02 !AND, L L 0010 -PS(C)
278     ZERO::=03 !ZERO, L L 0011 -PS(C)
279     A-PLUS-B-PLUS-D(C)::=04 !PLUS, L A 1001 -D(C)
280     A-PLUS-NOT-B-PLUS-D(C)::=05 !PLUS, L A 0110 -D(C)
281     A-XOR-B::=06 !XOR, L L 0110 -D(C)
282     A-AND-NOT-B::=07 !AND, L L 0111 -D(C)
283     DIVIDE::=10 !DIVIDE STEP
284     SUB, IF D(C)H=1 A 0110 -D(C)=-1
285     ADD, IF D(C)H=0 A A 1001 -D(C)=-0
286     A-PLUS-B::=11 !PLUS, L A 1001 -0
287     B::=12 !SELECT B, L L 1010 -0
288     A-AND-B::=13 !AND, L L 1011 -0
289     A-PLUS-B-PLUS-1::=14 !PLUS, L A 1001 -1
290     A-MINUS-B::=15 !MINUS, L A 0110 -1
291     A-IOR-B::=16 !IOR, L L 1110 -1
292     A::=17 !SELECT A, L L 1111 -1

```

```

295 .TOC * (BEN)-B-BUS DATA SOURCE
296 !SPECIFIES GATING OF DATA ONTO B-BUS. ALWAYS IN EFFECT.
297 .FIELD BEN::=(43:42)
298     BSPLO::=0 !DIRECT BSP LOCATIONS 00-17
299     BSPHI::=1 !DIRECT BSP LOCATIONS 20-37
300     CSP::=2 !USE (CSPADDR) (SIC) AS ADDRESS (4 BIT)
301     BASCON::=3 !1 OF 4 BASE CONSTANTS IN CSP17 TO CSP14 (2 BIT)

```

```

305 .TOC * (BSEL)-B-BUS SOURCE SELECTION CONTROL
306 !SPECIFIES CONTROL OF INDIVIDUAL B-BUS SOURCES. ALWAYS IN EFFECT.
307 .FIELD BSEL::=(41:40)
308 !NOT USED WHEN BEN/CSP
309 !CSP17 TO CSP14 IMMEDIATE ADDRESS WHEN BEN/BASCON
310     B17::=0
311     B16::=1
312     B15::=2
313     B14::=3
314 !USED IN CONJUNCTION WITH (RIF) FOR SP ADDRESS WHEN BEN/BSPLO OR BEN/BSPHI
315     DF::=0 !DESTINATION FIELD
316     SF::=1 !SOURCE FIELD
317     IMMEDO::=2 !DIRECT ADDRESS, LOW BIT=0
318     R00::=2 !FOR JOINT USE W/ RIF FIELD
319     R02::=2
320     R04::=2
321     R06::=2
322     R10::=2
323     R12::=2
324     R14::=2

```

```

325      R16::=2
326      IMMEDI::=3
327      R01::=3
328      R03::=3
329      R05::=3
330      R07::=3
331      R11::=3
332      R13::=3
333      R15::=3
334      R17::=3
335      C000000::=3
336      C177777::=3
337      C125252::=3
338      C052525::=3
339      C000001::=3
340      C100000::=3
341      C000200::=3
342
343
344
345      .TOC *      <AEN>-A-BUS DATA SOURCE
346      !SPECIFIES GATING OF DATA ONTO A-BUS. ALWAYS IN EFFECT.
347      .FIELD AEN::=<39:38>
348      XMUX::=0
349      CMUX::=1
350      ASPLO::=2
351      ASPHI::=3
352
353
354
355      .TOC *      <ASEL>-A-BUS SOURCE SELECTION CONTROL
356      !SPECIFIES CONTROL OF INDIVIDUAL A-BUS SOURCES. ALWAYS IN EFFECT.
357      .FIELD ASELO::=<36>
358      !XMUX CONTROL WHEN AEN/XMUX [USES ASELO ONLY]
359      SR::=0
360      FLTPT::=1
361      .FIELD ASEL::=<37:36>
362      !CMUX CONTROL WHEN AEN/CMUX. SHIFTS CMUX INPUT APPROPRIATE AMOUNT
363      LEFT-1::=0
364      DIRECT::=1
365      RIGHT-1::=2
366      RIGHT-2::=3
367      !USED IN CONJUNCTION WITH <RIF> FOR SP ADDRESS WHEN AEN/ASPLO OR AEN/ASPHI
368      IMMEDO::=0
369      R00::=0
370      R02::=0
371      R04::=0
372      R06::=0
373      R10::=0
374      R12::=0
375      R14::=0
376      R16::=0
377      IMMEDI::=1
378      R01::=1

```

```

!DIRECT ADDRESS, LOW BIT=1
!FOR JOINT USE W/ RIF FIELD

```

```

!ASPHI/BSPHI CONSTANTS

```

```

!XMUX=SR OR FLTPT ASSEMBLE
!SHIFT TREE
!DIRECT ASP LOCATIONS 00-17
!DIRECT ASP LOCATIONS 20-37

```

```

!SR OUTPUT ONTO BUS-A
!FLTPT-ASSEMBLE ONTO BUS-A

```

```

!LOW BIT GETS SENOMUX OUTPUT
!OUTPUT=INPUT
!HIGH BIT GETS D(C)
!HIGH BITS BOTH GET D(C)

```

```

!DIRECT ADDRESS, LOW BIT=1
!FOR JOINT USE W/ RIF FIELD

```



```

379      R03::=1
380      R05::=1
381      R07::=1
382      R11::=1
383      R13::=1
384      R15::=1
385      R17::=1
386      C000000::=1      :ASPBI/BSPBI CONSTANTS
387      C177777::=1
388      C125252::=1
389      C052525::=1
390      C000001::=1
391      C100000::=1
392      C000200::=1
393      DF::=2           :DESTINATION FIELD
394      SF::=3           :SOURCE FIELD
395
396
397

```

```

398      .TOC *      (RIF)-ASP, BSP REGISTER IMMEDIATE FIELD
399      !SPECIFIES ADDRESSES WITH ASP, BSP ALONG WITH AEN, ASEL & BEN, BSEL
400      .FIELD RIF::=<35:33>

```

```

401      R00-OR-01::=4      :LOW BIT IS 0/1, SPECIFIED BY
402      R00::=4
403      R01::=4
404      R02-OR-03::=5      :USING EITHER IMMEDO/IMMEDI MODES
405      R02::=5
406      R03::=5
407      R04-OR-05::=6
408      R04::=6
409      R05::=6
410      R06-OR-07::=7
411      R06::=7
412      R07::=7
413      R10-OR-11::=0
414      R10::=0
415      R11::=0
416      R12-OR-13::=1      :ADDR<3:0>H = -RIF<2>H # RIF<1:0>H # A/BSEL<0>H
417      R12::=1
418      R13::=1
419      R14-OR-15::=2
420      R14::=2
421      R15::=2
422      R16-OR-17::=3
423      R16::=3
424      R17::=3
425      C000000::=4      :ASPBI/BSPBI CONSTANTS
426      C177777::=5
427      C125252::=6
428      C052525::=7
429      C000001::=0
430      C100000::=1
431      C000200::=2
432

```

KD11-K

MICRO V00A-1 00:00:03 12-MAR-77

```

433
434
435 .TOC * <COUT>-CARRY OUT BIT MUX SELECTION
436 !SPECIFY INPUT TO D[C] REGISTER, LOADED WHEN D REGISTER LOADED. ALWAYS IN EFFECT.
437 .FIELD COUT::=<32:30>
438     CIN::=0           !OUTPUT OF CINMUX [SIC]
439     PS[C]::=1        !PS C-BIT
440     ALU00::=2        !ALU OUTPUT BIT 00
441     ALU07::=3        !ALU OUTPUT BIT 07
442     ALU15::=4        !ALU OUTPUT BIT 15
443     COUT07::=5       !BYTE CARRY BIT
444     COUT15::=6       !WORD CARRY BIT
445     D[C]::=7         !PROPOGATE [SAVE] LAST D[C]
446
447 !-----
448
449
450
451 .TOC * CLOCKS
452
453
454
455 .TOC * <WHEN>-D/SR WHEN TO CLOCK
456 !SPECIFY CLOCK D/SR REGISTERS AT P2 T OR P3 T. ALWAYS IN EFFECT.
457 .FIELD WHEN::=<29>,0
458     AT-P2-T::=0      !CLOCK D AND/OR SR AT P2 T[100 NS].
459     AT-P3-T::=1      !CLOCK D AND/OR SR AT P3 T[150 NS].
460
461
462
463 .TOC * <CLKD>-ENABLE D-REGISTER CLOCKING
464 !ENABLES CLOCKING OF D-REGISTER. ALWAYS IN EFFECT.
465 .FIELD CLKD::=<28>,0
466     NO::=0           !NOP
467     YES::=1          !CLOCK D[C], D-REGISTER AT <WHEN>
468
469
470
471 .TOC * <CLKSR>-ENABLE SR-REGISTER CLOCKING
472 !ENABLES CLOCKING OF SR-REGISTER. ALWAYS IN EFFECT.
473 .FIELD CLKSR::=<27>,0
474     NO::=0           !NOP
475     YES::=1          !CLOCK SR-REGISTER AT <WHEN>
476
477
478
479 .TOC * <CLKBA>-ENABLE CLOCKING OF BA-REGISTER
480 !ENABLES CLOCKING OF BA-REGISTER AT P1 T[60 NS]. ALWAYS IN EFFECT.
481 .FIELD CLKBA::=<26>,0
482     NO::=0           !NOP
483     YES::=1          !CLOCK BA-REGISTER AT P1 T[60 NS].
484
485
486

```

```

487 .TOC * <SCC>-ENABLE SETTING OF PS CONDITION CODES
488 !ENABLE CLOCKING OF PS CONDITION CODES AT P2 T(100 NS) OF NEXT UWORD. 0 MUST
489 !BE CLOCKED AT P2 T OR EARLIER OF PREVIOUS MICROWORD. ALWAYS IN EFFECT.
490 .FIELD SCC::=<25>,0
491 NO::=0 !NOP
492 YES::=1 !ENABLE CLOCKING IN NEXT UWORD
493
494 !-----
495
496
497 .TOC * BUS/UCON & CSP-ADDRESS & SHIFT-TREE CONTROL
498
499
500
501 .TOC * BUS/UCON CONTROL
502
503
504
505 .TOC * <BEGIN>-BEGIN BUS/UCON OPERATION
506 !INITIATE BUS XOR UCON OPERATION. ALWAYS IN EFFECT.
507 .FIELD BEGIN::=<24>,0
508 NO::=0 !NOP FOR BUS AND UCON OPERATIONS
509 YES::=1 !BEGIN OPERATION SPECIFIED
510
511
512
513 .TOC * <SELECT>-SELECT BUS OR UCON
514 !SELECT BUS XOR UCON. ONLY USED IF BEGIN/YES.
515 .FIELD SELECT::=<23>
516 BUS::=0 !SELECT BUS
517 UCON::=1 !SELECT UCON
518
519
520
521 .TOC * BUS CONTROL
522
523
524 .TOC * <BUSCODE>-BUS CODE ACTION FIELD
525 !BUS ACTION CODES. ONLY USED IF BEGIN/YES & SELECT/BUS.
526 .FIELD BUSCODE::=<22:20>
527 DATI-CLKIR::=0 !DATA IN, LOAD IR
528 DATI-NOINT::=1 !DATA IN, NO INTERNAL ADDRESS
529 DATO::=2 !DATA OUT
530 DATIB::=3 !DATA IN, ALLOW: ODD ADDRESS
531 DATIB(P)::=3 !DATA IN, ALLOW: ODD ADDRESS, FORCE TO PAUSE
532 DATIP::=4 !DATA IN, NO CACHE, LOCK BUS
533 DATOB::=5 !DATA OUT, ALLOW: ODD ADDRESS
534 DATI::=6 !DATA IN
535 DATI(P)::=6 !DATA IN, ALLOW: FORCE TO PAUSE
536 INVALIDATE::=7 !INVALIDATE CACHE LOCATION FUNCTION
537
538
539
540 .TOC * UCON CONTROL

```

```

541
542
543
544 .TOC      ,      <FLPGO>-START HOT FLOATING POINT
545 !INITIATES HOT FLOATING POINT FUNCTION. ONLY USED IF BEGIN/YES & SELECT/UCON.
546 .FIELD FLPGO::=<22>
547         NO::=0          !NOP
548         YES::=1         !YELL GO
549
550
551
552 .TOC      *      <UCON-XFER>-UCON OPERATION
553 !EXECUTE A UCON FUNCTION. ONLY USED IF BEGIN/YES & SELECT/UCON.
554 .FIELD UCON-XFER::=<21>
555         NO::=1          !NOP
556         YES::=1         !START UCON OPERATION
557
558
559
560 .TOC      *      <UCON-LOAD>-LOAD UCON REGISTER
561 !LOAD UCON CONTROL REGISTER. ONLY USED IF BEGIN/YES & SELECT/UCON.
562 .FIELD UCON-LOAD::=<20>
563         NO::=0          !NOP
564         YES::=1         !LOAD UCON CONTROL REGISTER
565
566
567
568 .TOC      *      CSP ADDRESS SPECIFICATION
569
570
571 .TOC      *      <CSPADDR>-CSP IMMEDIATE ADDRESS
572 !SPECIFY CSP 4 BIT ADDRESS. ONLY USED IF BEN/CSP.
573 .FIELD CSPADDR::=<23:20>
574         D00::=17        !NOTE INVERSION
575         D01::=16
576         D02::=15
577         D03::=14
578         D04::=13
579         D05::=12
580         D06::=11
581         D07::=10
582         D10::=07
583         D11::=06
584         D12::=05
585         D13::=04
586         D14::=03
587         D15::=02
588         D16::=01
589         D17::=00
590         C052525::=07    !CSP/DCS CONSTANTS
591         C125252::=06
592         C177777::=05
593         C000000::=04
594

```

KD11-K

MICRO

VOOA-1 00:00:03 12-MAR-77

595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648

.TOC \* SHIF\* CONTROL

.TOC \* <BMUX>-SECOND LEVEL OF SHIFT TREE  
 !BMUX CONTROLS SHIFT RIGHT OF 0 OR 4. ALWAYS IN EFFECT.  
 .FIELD BMUX::=<23>  
 DIRECT::=0 !AMUX<15:00>  
 RIGHT-4::=1 !4\*D[C] # AMUX <15:04>

.TOC \* <AMUX>-FIRST LEVEL OF SHIFT TREE  
 !AMUX CONTROLS INPUT OF D-REG/COUNTER TO TREE. ALWAYS IN EFFECT.  
 .FIELD AMUX::=<22:20>  
 DIRECT::=0 !D<HI> # D<LO>  
 D[LO]#D[LO]::=1 !D<LO> # D<LO>  
 SIGNEXT::=2 !8\*D[C] # D<LO>  
 COUNTER#D[LO]::=3 !COUNTER # D<LO>  
 COUNTER::=3 !SAME  
 D[HI]#D[HI]::=4 !D<HI> # D<HI>  
 SWAB::=5 !D<LO> # D<HI>  
 RIGHT-8::=6 !8\*D[C] # D<HI>  
 COUNTER#D[HI]::=7 !COUNTER # D<HI>

-----  
 .TOC \* SP REWRITE & REGISTER CLOCKS

.TOC \* <WRCSP>-WRITE TO CSP  
 !WRITE CSP FROM DMUX (BUSDIN/CACHE). ALWAYS IN EFFECT.  
 .FIELD WRCSP::=<19>,0  
 NO::=0 !NOP  
 YES::=1 !ON P3, 120-150 NS.

.TOC \* <MOD>-MODE CONTROL OF FOLLOWING BITS  
 !CONTROLS REDEFINITION OF SP REWRITE/REGISTER CLOCK BITS. ALWAYS IN EFFECT.  
 .FIELD MOD::=<14>,0  
 CLKSP::=0 !CONTROL ASP/BSP CLOCKING  
 LOADREG::=1 !CONTROL RES-REG/COUNTER LOADING

.TOC \* SP REWRITE (A,B) CONTROL  
 !WHEN MOD/CLKSP

```

649 .TOC * <HILO>-SP HI/LO SELECT
650 !WHICH HALF OF SP'S TO REWRITE. ONLY IF MOD/CLKSP.
651 .FIELD HILO::=<18>
652 LO::=0 !REWRITE ENABLE A/B SP LO [00-17]
653 L::=0 !
654 HI::=1 !REWRITE ENABLE A/B SP HI [20-37]
655 H::=1 !
656
657
658 .TOC * <WRSEL>-REWRITE ADDRESS SELECT
659 !WHICH WRITE ADDRESS TO USE ON REWRITE. ONLY IF MOD/CLKSP.
660 .FIELD WRSEL::=<17>
661 A-ADDR::=0 !USE A ADDRESS ON REWRITE
662 A::=0 !
663 B-ADDR::=1 !USE B ADDRESS ON REWRITE
664 B::=1 !
665
666
667
668 .TOC * <WRSP>-REWRITE A/B SELECT
669 !ENABLE REWRITE OF SPECIFIC SP'S. ONLY IF MOD/CLKSP.
670 .FIELD WRSP::=<16:15>
671 NOP::=0 !NO ASP/BSP REWRITE
672 WR-A::=1 !WRITE ASP ONLY, ON P3 120-150 NS.
673 A::=1 !
674 ASP::=1 !
675 WR-B::=2 !WRITE BSP ONLY, ON P3 120-150 NS.
676 B::=2 !
677 BSP::=2 !
678 WR-A-AND-B::=3 !WRITE BOTH ON P3
679 AB::=3 !
680 BA::=3 !
681 ABSP::=3 !
682 BASP::=3 !
683 BOTH::=3 !
684
685
686
687
688 .TOC * REGISTER LOADING
689 !WHEN MOD/LOADREG
690
691
692 .TOC * <LOADRES>-LOAD RESIDUAL CONTROL REGISTER
693 !ENABLE LOAD OF RESIDUAL CONTROL REGISTER FROM B-BUS. ONLY IF MOD/LOADREG.
694 .FIELD LOADRES::=<18>
695 NO::=0 !NOP
696 YES::=1 !LOAD RES WITH B-BUS<14:11>
697 !AT P2 T[100 NS], B-BUS<14> COMPLEMENTED
698
699
700
701 .TOC * <LOADCOUNT>-LOAD COUNTER
702 !ENABLE LOAD OF COUNTER FROM B-BUS <7:0>. ONLY IF MOD/LOADREG.

```

```

703 .FIELD LOADCOUNT: := <16>
704     NO: := 0                !NOP
705     YES: := 1              !LOAD COUNTER AT P2 T(100 NS).
706
707 !-----
708
709
710
711 .TOC * SEQUENCING FIELD
712
713
714
715 .TOC * <UBF>-BUT MICROBRANCH FIELD
716 !SPECIFIES CONDITIONS TO MODIFY <UPF /<J> FIELD DURING BRANCH. ALWAYS IN EFFECT.
717 .FIELD UBF: := <13:9>, 30
718
719 .TOC * NO BUT
720     NULL: := 30                !SPECIFY NO MODIFICATION - DEFAULT
721
722 .TOC * ACTIVE ONLY
723 !PURELY ACTIVE BUTS GENERATE SIDE AFFECTS; THEY DO NOT MODIFY THE <UPF> FIELD
724 !BY THE "OR-ING"-IN-OF-CONDITIONS METHOD. THEY MAY MODIFY EXPLICITLY THE ENTIRE <UPF> FIELD.
725 !AS IN BUT(RETURN)
726     R-OR-1: := 22              !FORM R(SF)-IOR-"001
727     CUA-TRACK: := 31          !RESUME/RESTART CUA TRACKING
728     CLR-FLAG-RES-UCON: := 32 !CLEAR FLAGS<2:0>, EX-FLAG<1>, RES-REGISTER, UCON-REGISTER
729     DIAGNOSE: := 33          !SPECIAL DIAGNOSTIC BUT
730     SUBR8: := 34             !RETURN (- EMIT<14:03>, PAGE (- EMIT<02:00>
731     SUBR-B: := 34           !SYNONYMS ARE:
732     GOTO: := 34
733     GO-TO: := 34
734     SUBRA: := 35            !RETURN (- D<14:03>, PAGE (- EMIT<02:00>
735     SUBR-A: := 35          !SYNONYM
736     B#36: := 36            !TBD
737     RETURN: := 37          !PAGE (- RETURN<11:09>, NUA (- RETURN<08:00>
738
739 .TOC * INACTIVE ONLY
740 !INACTIVE BUTS ONLY CAUSE MODIFICATION OF THE <UPF> FIELD BY THE "OR-ING"-
741 !IN-OF-CONDITIONS METHOD.
742
743
744     SR3-0: := 00            !----UPF MASK-----
745     CASE: := 00            !876 543 210 OCTAL *=NOT AFFECTED
746     SR03: := 00           !*** **0 000 (000)
747     SR02: := 00           !*** **1 111 (007)
748     SR01: := 00           !*** **1 011 (013)
749     SR00: := 00           !*** **1 101 (015)
750     IR15-12: := 01        !*** **1 110 (016)
751     DOP: := 01            !*** **0 000 (000)
752     INSTRS: := 02         !*** *00 000 (000)
753     INSTR-5: := 02        !*** *00 000 (000)
754     IR11#FLTPT3-0: := 03 !*** *01 111 (017)
755     IR11-A: := 03        !*** **0 000 (000)
756     IR9-6: := 04

```

757	SOP::=04	!*** **0 000 (000)
758	MOV-DR7#IRS-3::=05	!*** **0 111 (007)
759	MOV-DR7::=05	!*** **1 000 (010)
760	IRS-3::=05	!*** **0 000 (000)
761	BGSEV-FPSERV#D(C)#FPRET::=07	!*** **0 111 (007)
762	BGSEV-FPSERV::=07	!*** **1 011 (013)
763	D(C)-C::=07	!*** **1 100 (014)
764	FPRET1-0::=07	!*** *** 000 (000)
765	COUT07#00OUT07#FPSOS::=10	!*** *** 011 (003)
766	COUT07::=10	!*** *** 101 (005)
767	DOUT07::=10	!*** *** 001 (001)
768	COUT07#DOUT07::=10	!*** *** 110 (006)
769	FPSOS::=10	!*** *** 000 (000)
770	DMD#SMD#BYTE::=11	!*** *** 011 (003)
771	DMD::=11	!*** *** 101 (005)
772	SMD::=11	!*** *** 110 (006)
773	BYTE::=11	!*** *** *00 (000)
774	GD3-2::=12	!*** *** 000 (000)
775	BG-SERVICE-L#MFSS#MULTIPLE::=14	!*** *** 011 (003)
776	BG-SERVICE-L::=14	!*** *** 101 (005)
777	MFSS::=14	!*** *** 110 (006)
778	MULTIPLE::=14	!*** *** *00 (000)
779	MASKED-PS(T)::=14	
780	DOO::=14	
781	PS(N)::=14	
782	FLAG7::=14	
783	EXFLAG1::=14	
784	FLTPTS::=14	
785	EXFLAG2::=14	
786	INIT-JAM::=14	
787	D14-00E00#015::=15	!*** *** *00 (000)
788	D14-00-E0-0#015::=15	!*** *** *01 (001)
789	D14-00-E0-0::=15	!*** *** *10 (002)
790	D15::=15	!*** *** *00 (000)
791	IRI1#PS15::=16	!*** *** *01 (001)
792	IF11-8::=16	!*** *** *10 (002)
793	PS15::=16	!*** *** *00 (000)
794	VECTOR-LOAD#DR6-7L::=21	!*** *** *01 (001)
795	VECTOR-LOAD::=21	!*** *** *10 (002)
796	DR6-7L::=21	!*** *** *00 (000)
797	D(C)#BA00::=23	!*** *** *01 (001)
798	D(C)-8::=23	!*** *** *10 (002)
799	BA00::=23	!*** *** *00 (000)
800	OTHER-JAM#FP-PROC::=24	!*** *** *01 (001)
801	OTHER-JAM::=24	!*** *** *10 (002)
802	FP-PROC::=24	!*** *** *00 (000)
803	INTR-HIGH#INSTR-BRANCH-L::=26	!*** *** *01 (001)
804	INTR-HIGH::=26	!*** *** *10 (002)
805	INSTR-BRANCH-L::=26	!*** *** *00 (000)
806	PREFETCH-JAM#FP-FD::=27	!*** *** *01 (001)
807	PREFETCH-JAM::=27	!*** *** *10 (002)
808	FP-FD::=27	
809		
810	.TOC * BOTH ACTIVE AND INACTIVE	



```

811 !THESE BUTS HAVE BOTH ACTIVE AND INACTIVE EFFECTS
812
813 :-----UPF MASK-----
814 INSTR1::=06 :876 543 210 OCTAL *=NOT AFFECTED
815 INSTR-1::=06 :*00 000 000 (000) BUS CONTROL, SP REWRITE DEFEAT
816 SRI-0#COUNT-IS-377::=13 :*** ** 000 (000) BUMP COUNTER
817 SRI-0::=13 :*** ** 001 (001) STILL BUMP COUNTER
818 COUNT-IS-377-A::=13 :*** ** 110 (006) BUMP COUNTER
819 COUNT-IS-377#D(C)::=17 :*** ** *00 (000) BUMP COUNTER
820 COUNT-IS-377-B::=17 :*** ** *01 (001) BUMP COUNTER
821 D(C)-A::=17 :*** ** *10 (002) STILL BUMP COUNTER
822 COUNT-IS-377::=25 :*** ** *0 (000) BUMP COUNTER
823 PREFETCH-L#SERVICE::=20 :*** ** *00 (000) TIMING
824 PREFETCH-L::=20 :*** ** *01 (001) TIMING
825 SERVICE::=20 :*** ** *10 (002) TIMING
826 LAST::=20 :*** ** *11 (003) TIMING
827
828
829
830

```

```

831 .TOC * <UPF>-MICRO POINTER FIELD
832 !SPECIFIES EITHER NEXT MICROINSTRUCTION ADDRESS OR BASE TARGET
833 !ADDRESS TO BE USED "UNDER" THE BUT-CODE IN <UBF>.
834 .FIELD UPF::=<8:0>,000 !ACTUAL MICROWORD POINTER FIELD
835 .ADDRESS J::=<8:0> !THIS FIELD ALSO HAS MICROADDRESS QUALITIES
836
837
838

```

!BASE MACHINE MICROCODE ENTRY POINTS:

```

839 !THESE ENTRY POINTS HAVE BEEN FIXED AS OF 31-AUGUST-1976.
840 INIT01 ::= 3412 !INITIALIZATION SUBROUTINE
841 CON99 ::= 1040 !FORCE "CONSOLE-MODE HALT"
842 FET01 ::= 0702 !INSTR FETCH, NO OVERLAP
843 FET03 ::= 0700 !INSTR FETCH, OVERLAP
844 SER01 ::= 0701 !SERVICE ENTRY, OVERLAP
845 SER02 ::= 0703 !SERVICE ENTRY, NO OVERLAP
846

```

!ENTRY POINTS INTO BASE MACHINE FOR "BUTA(DIAGNOSE)":

```

848 !THESE ENTRY POINTS FIXED AS OF 26-OCT-76:
849 MED23 ::= 3200 !FOR FLPAOR: D+ASPLO(DF)-TOP
850 MED25 ::= 3020 !FOR FLPAOR: D+ASPHI(DF)-TOP
851 MED27 ::= 3210 !FOR FLPAOR: ASPLO(DF)-TOP+D
852 MED29 ::= 3214 !FOR FLPAOR: ASPHI(DF)-TOP+D
853 MED31 ::= 3044 !FOR FLPAOR: D+BSPLO(DF)-TOP
854 MED33 ::= 3230 !FOR FLPAOR: D+BSPHI(DF)-TOP
855 MED35 ::= 3234 !FOR FLPAOR: BSPLO(DF)-TOP+D
856 MED37 ::= 3064 !FOR FLPAOR: BSPHI(DF)-TOP+D
857

```

```

858 BYTE01 ::= 0032 !FOR KJENAB: DAT08*KJENAB
859
860 RTS02 ::= 4034 !FOR SPBYKT: DAT1, BA+SP-A, SP+SP+2
861
862 DST01 ::= 0511 !FOR ALTER: DATIB(P)
863 DST02 ::= 0512
864 DST20 ::= 0527

```

```

865      DST22 ::= 0525      !
866
867      LOADNZW4 ::= 4330      !FOR FPSCC#CLKFPSCC: D+CSP(MD)
868      LOADNZW5 ::= 4332      !                      FPS<3:0>+D<3:0>
869
870      !-----
871
872
873
874      .TOC * MISCELLANEOUS FIELDS
875
876
877
878      .TOC * <NEXT-PAGE>-NEW PAGE ADDRESS LOADED DURING BUT(SUBROUTINE)
879      !THESE 3 BITS ARE CLOKED INTO PAGE REGISTER DURING A BUT(SUBRA) OR
880      !BUT(SUBRB). ONLY USED WHEN UBF/BUT(SUBRA) OR UBF/BUT(SUBRB).
881      .FIELD NEXT-PAGE::=<32:30>
882
883
884      .TOC * <MULTIPLE>-SELECT CODE FOR BUT(MULTIPLE)
885      !MUST BE SET IN BOTH PREVIOUS AND CURRENT MICROWORDS WHEN BUT(MULTIPLE) IS TO BE EMPLOYED.
886      .FIELD MULTIPLE::=<32:30>
887      MASKED-PS(I)::=0      !
888      DOO::=1              !
889      PS(N)::=2            !
890      FLAG7::=3            !
891      EXFLAG1::=4          !
892      FLTPTS::=5           !
893      EXFLAG2::=6          !
894      INIT-JAM::=7        !
895
896
897
898      .TOC * EMIT FIELD - IMMEDIATE DATA FROM MICROWORD
899      !USED WHENEVER LOADING IMMEDIATE DATA FROM MICROWORD
900      .FIELD EMIT::=<47:44>'<41:30>
901      .FIELD EMITH::=<47:44>
902      .FIELD EMITM::=<41:38>
903      .FIELD EMITL::=<37:30>
904      .FIELD EMITML::=<41:30>
905      .FIELD EMIT9-6::=<39:36>
906      .FIELD EMIT15::=<47>
907      .FIELD EMIT14::=<46>
908      .FIELD EMIT13::=<45>
909      .FIELD EMIT12::=<44>
910      .FIELD EMIT11::=<41>
911      .FIELD EMIT10::=<40>
912      .FIELD EMIT09::=<39>
913      .FIELD EMIT08::=<38>
914      .FIELD EMIT07::=<37>
915      .FIELD EMIT06::=<36>
916      .FIELD EMIT05::=<35>
917      .FIELD EMIT04::=<34>
918      .FIELD EMIT03::=<33>

```

```

919 .FIELD EMIT02::=<32>
920 .FIELD EMIT01::=<31>
921 .FIELD EMIT00::=<30>
922
923
924
925 .TOC * RETURN ADDRESS - FOR MICROSUBROUTINE CALLS
926 !USED WITH BUT(SUBRB) AND BUT(SUBRA)
927 .FIELD RETURN::=<46:44>'<41:33> !PAGE # D.I.P.
928
929
930
931 .TOC * UCON SELECTION AND CONTROL FIELDS
932
933
934 .TOC * SELECTION
935 !SELECT PARTICULAR UCON. ONLY USED IF BEGIN/YES & SELECT/UCON.
936 .FIELD UCON-SEL-EMIT::=<43> !SELECT EMIT CAN ONLY BE DONE BY USING
937 ! NO::=0 ! BUTA(CLR-FLAG-RES-UCON) TO ASSERT UCON-SEL-EMIT-L
938 ! YES::=1
939 .FIELD UCON-SEL-I 0::=<46> !SELECT I-0 [BUS] CONTROL
940 ! NO::=0
941 ! YES::=1
942 .FIELD UCON-SEL-WCS::=<45> !SELECT WCS/ECS/DCS
943 ! NO::=0
944 ! YES::=1
945 .FIELD UCON-SEL-CACHEKT::=<44> !SELECT CACHE/KT
946 ! NO::=0
947 ! YES::=1
948 .FIELD UCON-SEL-PROG::=<36> !SELECT PROCESSOR CONTROL
949 ! NO::=0
950 ! YES::=1
951 .FIELD UCON-SEL-FLTPT::=<33> !SELECT HOT FLOATING POINT
952 ! NO::=0
953 ! YES::=1
954
955
956
957 .TOC * CONTROL
958 !AFTER UCON(S) SELECTED FROM ABOVE, CONTROL COMES FROM HERE.
959 .FIELD UCON::=<32:30>'<35:34>'<47>'<42:38>
960 .FIELD UCONH::=<32:30>
961 .FIELD UCONM::=<35:34>
962 .FIELD UCONL::=<47>'<42:38>
963 .FIELD UCON15::=<32>
964 .FIELD UCON14::=<31>
965 .FIELD UCON13::=<30>
966 .FIELD UCON12::=<35>
967 .FIELD UCON11::=<34>
968 .FIELD UCON10::=<47>
969 .FIELD UCON09::=<42>
970 .FIELD UCON08::=<41>
971 .FIELD UCON07::=<40>
972 .FIELD UCON06::=<39>

```

973 .FIELD UCON05::=(38)

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025

1026

TOC \* BASE MACHINE EXTENSION BITS  
LAYOUT IN BASE MACHINE (NOT DCS) ADDRESS SPACE:

NAME	0	1	2	3	4
ROMEX 00 H	X				X
ROMEX 01 H			X	X	
ROMEX 03 H				X	
FPSEL L	X		X		X
SFCC L	X		X		X
FLPADR L	X		X	X	X
SPBYKT L	X				X
UDAD01 L	X				X
UDAD00 L	X				X
UKJCONT L	X				X
UKT01 H	X				X
UKT00 H	X				X
UALTER L	X				X
UKTEN L	X				X

NULL BIT DEFINITIONS:

.FIELD ROMEX00::=(60) :ACTIVE HIGH  
 ZERO::=0  
 ONE::=1

.FIELD ROMEX01::=(61) :ACTIVE HIGH  
 ZERO::=0  
 ONE::=1

.FIELD ROMEX03::=(62) :ACTIVE HIGH  
 ZERO::=0  
 ONE::=1

.FIELD FPSEL::=(57) :ACTIVE LOW  
 ZERO::=1  
 ONE::=0

.FIELD SFCC::=(58) :ACTIVE LOW  
 ZERO::=1  
 ONE::=0

.FIELD FLPADR::=(59) :ACTIVE LOW  
 ZERO::=1  
 ONE::=0

.FIELD SPBYKT::=(55) :ACTIVE LOW  
 ZERO::=1  
 ONE::=0

.FIELD UDAD01::=(53) :ACTIVE LOW

```

1027     ZERO::=1
1028     ONE::=0
1029 .FIELD UDAD00::=<52>      ACTIVE LOW
1030     ZERO::=1
1031     ONE::=0
1032 .FIELD UKJCONT::=<54>    ACTIVE LOW
1033     ZERO::=1
1034     ONE::=0
1035 .FIELD UKTO1::=<49>     ACTIVE HIGH
1036     ZERO::=0
1037     ONE::=1
1038 .FIELD UKTO0::=<48>    ACTIVE HIGH
1039     ZERO::=0
1040     ONE::=1
1041 .FIELD UALTER::=<51>   ACTIVE LOW
1042     ZERO::=1
1043     ONE::=0
1044 .FIELD UKTEN::=<50>    ACTIVE LOW
1045     ZERO::=1
1046     ONE::=0

```

```

1047
1048 !-----
1049

```

```

1050
1051
1052 .TOC * SPECIAL DCS FIELDS
1053

```

```

1054
1055
1056 .TOC * FIELDS USED IN PAGES 4, 5, OR 6 OF DCS
1057

```

```

1058
1059 .TOC * <LOAD-DCS-CTR>-LOAD DIAGNOSTIC COUNTER FROM EMITH
1060 !THIS CODE LOADS THE 4-BIT DCS COUNTER FROM THE CURRENT
1061 !MICROWORD'S EMITH FIELD. THIS COUNTER IS CLOCKED AT EVERY PO
1062 !FOLLOWING, UNTIL THE COUNTER REACHES ZERO. AT THIS POINT, THE
1063 !COMPARE IS ENABLED, CLOCKING THE RESULT OF THE CURRENT ENUA:TNUA
1064 !COMPARE INTO THE ERROR LATCH.
1065 !ONLY USED IN PAGES 4, 5, OR 6 OF DCS.
1066 .FIELD LOAD-DCS-CTR::=<51>,0
1067     NOP::=0           !NOP
1068     YES::=1          !LOAD, COUNT, AND ENABLE COMPARE.
1069

```

```

1070
1071
1072 .TOC * <CTR>-4 BIT DCS COUNTER VALUE FROM EMIT
1073 !THIS FOUR BIT VALUE IS LOADED INTO THE COUNTER (DIAGNOSTIC!),
1074 !WHEN LOAD COUNTDOWN/YES. COMPLEMENT OF ACTUAL VALUE IS USED, FOR COUNT DOWN.
1075 !LOADING COUNTER VALUE OF 17(8) CAUSES COMPARE AT END OF THIS UWORD.
1076 .FIELD CTR::=<47:44>
1077     C0::=17
1078     C1::=16
1079     C2::=15
1080     C3::=14

```

```

1081      C4.::=13
1082      C5.::=12
1083      C6.::=11
1084      C7.::=10
1085      C8.::=07
1086      C9.::=06
1087      C10.::=05
1088      C11.::=04
1089      C12.::=03
1090      C13.::=02
1091      C14.::=01
1092      C15.::=00

```

```

1096      .TOC *      <LOAD-ENUA-ERRCOD>-LOAD THE ENUA AND ERRCOD REGISTERS
1097      !THIS CODE LOADS THE 12-BIT ENUA REGISTER FROM THE <EMITM,EMITL> FIELD
1098      !OF THE CURRENT MICROWORD, AND LATCHES THE NUA INTO THE ERRCOD REGISTER.
1099      !ONLY USED IN PAGES 4, 5, OR 6 OF DCS.
1100      .FIELD LOAD-ENUA-ERRCOD::=<54>,0
1101      NOP::=0          !NOP
1102      YES::=1          !LOAD REGISTERS AT PO

```

```

1106      .TOC *      <ENUA>-ENUA VALUE FROM EMIT
1107      !THIS 12 BIT FIELD IS LOADED FROM <EMIT> TO THE ENUA REGISTER
1108      !WHEN LOAD ENUA-ERRCOD/YES.
1109      .FIELD ENUA::=<41:30>

```

```

1113      .TOC *      <VERIFY>-VERIFY BIT FOR SELF CHECK TEST
1114      !WHEN IN SELF TEST MODE OF DCS, SETTING THIS BIT CAUSES THE VERIFY COUNTER TO BE
1115      !BUMPED AT THE START OF THIS MICROWORD. THE VERIFY BIT IS IMPLICITLY SET FOR
1116      !ANY REFERENCE TO PAGE 7 (IE, THE COUNTER IS AUTOMATICALLY BUMPED ON A REFERENCE
1117      !TO PAGE 7).
1118      !ONLY [EXPLICITLY] USED IN PAGES 4, 5, OR 6 OF DCS, WHEN IN SELF TEST MODE.
1119      .FIELD VERIFY::=<48>,0
1120      NOP::=0          !NO ACTION
1121      BUMP::=1        !BUMP VERIFY COUNTER AT PO, WHEN IN SELF TEST MODE

```

```

1125      .TOC *      FIELDS USED IN PAGE 7 OF DCS EXTENSION

```

```

1128      .TOC *      <EOP>-SIGNAL SUCCESSFUL END OF PASS
1129      !THIS CODE SETS THE END OF PASS LATCH, LIGHTING THE EOP LED
1130      !ONLY USED IN PAGE 7 OF DCS.
1131      .FIELD EOP::=<49>,0
1132      NO::=0          !NO EOP
1133      SIGNAL::=1     !SIGNAL SUCCESSFUL EOP AT PO
1134

```

```

1135
1136
1137 .TOC * <DAD>-DCS CONTROL OF BASE MACHINE EXTENSION DAD BITS
1138 !THESE BITS ARE WIRE-AND'ED INTO THE BASE MACHINE DAD<1:0> BITS.
1139 !MUST BE SPECIFIED IN UWORD BEFORE CSP REFERENCE.
1140 !ONLY USED IN PAGE 7 OF DCS.
1141 .FIELD DAD::=<53:52>,0
1142     NO-DAD::=0
1143     FIRST-1-OR-2::=1      !SETUP BYTE-CONST ROM INPUT
1144     SECOND-1-OR-2::=2
1145     WRITE-BYTE::=3      !SETUP FOR BYTE WRITE TO ASH/BSP
1146
1147
1148
1149 .TOC * FIELDS USED IN ALL PAGES OF DCS EXTENSION
1150
1151
1152 .TOC * <SCOPE>-SCOPE ON ERROR, DIAGNOSTIC BUT
1153 !THIS CODE IS A SPECIAL BUT, THAT, WHEN ENABLED, CHECKS THE ERROR
1154 !LATCH TO SEE IF IT IS SET. IF IT IS, NUADD IS FORCED TO A ZERO!
1155 !ELSE IT IS LEFT UNCHANGED. USED TO IMPLEMENT FORCED SCOPE LOOP ON ERROR.
1156 !USED IN ALL PAGES OF DCS.
1157 .FIELD SCOPE::=<50>,0
1158     NOP::=0      !NOP
1159     ENABLED::=1  !ENABLE SCOPE LOOPING FACILITY
1160
1161
1162 !-----
1163 !END OF MICROWORD FIELD DEFINITIONS
1164 !-----
1165
1166
1167
1168 !.PAGE=====
1169
1170 .TOC * MACRO DEFINITIONS
1171
1172
1173 .TOC * PRIMITIVE OPERATIONS
1174
1175
1176 .TOC * TIMING
1177 .MACRO P0      ::= NULL      !0 NS., UP3 VIEWED AS THE START OF A MICROCYCLE
1178
1179 .MACRO P1      ::= NULL      !60 NS., AT P1
1180 .MACRO P1-L    ::= NULL      !30 NS., AT P1 LEADING EDGE
1181 .MACRO P1-T    ::= NULL      !60 NS., AT P1 TRAILING EDGE
1182
1183 .MACRO P2      ::= NULL      !100 NS., AT P2
1184 .MACRO P2-L    ::= NULL      !70 NS., AT P2 LEADING EDGE
1185 .MACRO P2-T    ::= WHEN/AT-P2-T !100 NS., AT P2 TRAILING EDGE
1186 .MACRO P2-U    ::= NULL      !UNSUPPRESSED P2, CLOCK CONTINUOUSLY
1187
1188 .MACRO P3      ::= NULL      !150 NS., 120-150 NS., AT P3

```





```

1243 .MACRO ASPHI(XX) ::= AEN/ASPHI, !SELECT
1244                   ASEL/0XX, !REGISTER &
1245                   RIF/0XX !ENABLE ON BUS-A
1246
1247 .MACRO ASP(XX) ::= ASEL/0XX, !SELECT REGISTER,
1248                   RIF/0XX !NO ENABLE
1249
1250
1251 .MACRO BSPL0(XX) ::= BEN/BSPL0, !SELECT
1252                   BSEL/0XX, !REGISTER &
1253                   RIF/0XX !ENABLE ON BUS-B
1254
1255 .MACRO BSPHI(XX) ::= BEN/BSPHI, !SELECT
1256                   BSEL/0XX, !REGISTER &
1257                   RIF/0XX !ENABLE ON BUS-B
1258
1259 .MACRO BSP(XX) ::= BSEL/0XX, !SELECT REGISTER,
1260                   RIF/0XX !NO ENABLE
1261
1262
1263
1264
1265

```

1266 .TOC \* ASP AND BSP BASE MACHINE FUNCTIONAL REGISTER ADDRESSES

1267  
1268 : ENABLE INPUT/OUTPUT (FOR READ AND/OR WRITE) OF THE APPROPRIATE SCRATCH PAD ONTO  
1269 : EITHER BUS-A "-A" OR BUS-B "-B" VIA FUNCTIONAL REGISTER DESIGNATION  
1270 :

```

1271 .MACRO R0-A ::= ASPL0(R00)
1272 .MACRO R0-B ::= BSPL0(R00)
1273 .MACRO R1-A ::= ASPL0(R01)
1274 .MACRO R1-B ::= BSPL0(R01)
1275 .MACRO R2-A ::= ASPL0(R02)
1276 .MACRO R2-B ::= BSPL0(R02)
1277 .MACRO R3-A ::= ASPL0(R03)
1278 .MACRO R3-B ::= BSPL0(R03)
1279 .MACRO R4-A ::= ASPL0(R04)
1280 .MACRO R4-B ::= BSPL0(R04)
1281 .MACRO R5-A ::= ASPL0(R05)
1282 .MACRO R5-B ::= BSPL0(R05)
1283 .MACRO SP-A ::= ASPL0(R06)
1284 .MACRO SP-B ::= BSPL0(R06)
1285 .MACRO PC-A ::= ASPL0(R07)
1286 .MACRO PC-B ::= BSPL0(R07)
1287 .MACRO FACA[0]-B ::= BSPHI(R10)
1288 .MACRO FACB[0]-A ::= ASPHI(R10)
1289 .MACRO FACC[0]-B ::= BSPL0(R10)
1290 .MACRO FADC[0]-A ::= ASPL0(R10)
1291 .MACRO FACA[1]-B ::= BSPHI(R11)
1292 .MACRO FACB[1]-A ::= ASPHI(R11)
1293 .MACRO FACC[1]-B ::= BSPL0(R11)
1294 .MACRO FADC[1]-A ::= ASPL0(R11)
1295 .MACRO FACA[2]-B ::= BSPHI(R12)
1296 .MACRO FACB[2]-A ::= ASPHI(R12)

```

```

1297 .MACRO FACD(2)-B      ::= BSPL0(R12)
1298 .MACRO FACD(2)-A      ::= ASPLO(R12)
1299 .MACRO FACA(3)-B      ::= BSPHI(R13)
1300 .MACRO FACB(3)-A      ::= ASPHI(R13)
1301 .MACRO FACD(3)-B      ::= BSPL0(R13)
1302 .MACRO FACD(3)-A      ::= ASPLO(R13)
1303 .MACRO FACA(4)-B      ::= BSPHI(R14)
1304 .MACRO FACB(4)-A      ::= ASPHI(R14)
1305 .MACRO FACD(4)-B      ::= BSPL0(R14)
1306 .MACRO FACD(4)-A      ::= ASPLO(R14)
1307 .MACRO FACA(5)-B      ::= BSPHI(R15)
1308 .MACRO FACB(5)-A      ::= ASPHI(R15)
1309 .MACRO FACD(5)-B      ::= BSPL0(R15)
1310 .MACRO FACD(5)-A      ::= ASPLO(R15)
1311 .MACRO FUSTA-B        ::= BSPHI(R17)
1312 .MACRO FUSTB-A        ::= ASPHI(R17)
1313 .MACRO FUSTC-B        ::= BSPL0(R17)
1314 .MACRO FUSTD-A        ::= ASPLO(R17)
1315 .MACRO FPSHI#FEC-A    ::= ASPHI(R16)
1316 .MACRO FEA-B          ::= BSPHI(R16)
1317 .MACRO USER-SP-A     ::= ASPLO(R16)
1318 .MACRO USER-SP-B     ::= BSPL0(R16)
1319 .MACRO MAMI-A        ::= ASPHI(R02)
1320 .MACRO R[ZERO] B     ::= BSPHI(R03)
1321 .MACRO R[IR]-A       ::= ASPHI(R17)
1322 .MACRO R[SRC]-B      ::= BSPHI(R04)
1323 .MACRO R[SRC]-A      ::= ASPHI(R04)
1324 .MACRO R[DST]-B      ::= BSPHI(R05)
1325 .MACRO R[DST]-A     ::= ASPHI(R05)
1326 .MACRO R[VECT]-B     ::= BSPHI(R02)
1327 .MACRO MCSB[0]-B     ::= BSPHI(R00)
1328 .MACRO MCSB[1]-B     ::= BSPHI(R01)
1329 .MACRO MCSA[0]-A     ::= ASPHI(R00)
1330 .MACRO MCSA[1]-A     ::= ASPHI(R01)
1331 .MACRO FPA-B         ::= BSPHI(R06)
1332 .MACRO CNSL-CNTL-B   ::= BSPHI(R07)
1333 .MACRO CNSL-CADR-A   ::= ASPHI(R07)
1334 .MACRO CNSL-SW-A     ::= ASPHI(R06)
1335 .MACRO CNSL-TMPSW-A  ::= ASPHI(R03)
1336
1337
1338
1339 .TOC *   ASP AND BSP INDIRECT REGISTER ADDRESSES
1340
1341 .ENABLE INPUT/OUTPUT [FOR READ AND/OR WRITE] OF THE APPROPRIATE SCRATCH PAD
1342 .ON BUS-A [A] OR BUS-B [B] USING INDIRECT ADDRESSING WITH THE IR,
1343 .WHERE:
1344
1345 .SF<3:0>H = [FLPADR H + KTSRCADR3 H] # [FLTPT L * IR8 H] # [IR7 H] # [IR6 H + ROR1 H]
1346
1347 .DF<3:0>H = [FLPADR H + KTDSTADR3 H] # [IR2 H] # [IR1 H] # [IRO H]
1348
1349 .MACRO R[SF]-LO-A      ::= PEN/ASPLO,ASEL/SF
1350 .MACRO R[SF]-LO-B      ::= BEN/BSPL0,BSEL/SF

```

```

1351 .MACRO R(SF)-HI-A      ::= AEN/ASPFI,ASEL/SF
1352 .MACRO R(SF)-HI-B      ::= BEN/BSPFI,BSEL/SF
1353 .MACRO R(DF)-LO-A      ::= AEN/ASPLO,ASEL/DF
1354 .MACRO R(DF)-LO-B      ::= BEN/BSPL0,BSEL/DF
1355 .MACRO R(DF)-HI-A      ::= AEN/ASPFI,ASEL/DF
1356 .MACRO R(DF)-HI-B      ::= BEN/BSPFI,BSEL/DF
1357 .MACRO R(SF)-A         ::= R(SF)-LO-A
1358 .MACRO R(SF)-B         ::= R(SF)-LO-B
1359 .MACRO R(DF)-A         ::= R(DF)-LO-A
1360 .MACRO R(DF)-B         ::= R(DF)-LO-B

```

```

1361
1362
1363
1364 .TOC *      ASP, BSP INDIRECT ADDRESSING
1365
1366 .:
1367 .:   THESE MACROS ONLY SELECT THE ADDRESS MODE FOR THE ASP AND BSP;
1368 .:   THE SELECTED SP IS NOT ENABLED ONTO THE BUS

```

```

1369 .MACRO ASP-ADDRS-R(DF) ::= ASEL/DF
1370 .MACRO ASP-ADDRS-R(SF) ::= ASEL/SF
1371 .MACRO BSP-ADDRS-R(DF) ::= BSEL/DF
1372 .MACRO BSP-ADDRS-R(SF) ::= BSEL/SF

```

```

1373
1374
1375
1376
1377 .TOC *      ASP AND BSP DCS SPECIFIC FUNCTIONAL REGISTER ADDRESSES
1378
1379 .:
1380 .:   ENABLE INPUT/OUTPUT (FOR READ AND/OR WRITE) OF THE APPROPRIATE SCRATCH PAD ONTO
1381 .:   EITHER BUS-A "-A" OR BUS-B "-B" VIA FUNCTIONAL REGISTER DESIGNATION

```

```

1382 .MACRO C000000-A        ::= ASPFI(C000000)    !IN R01
1383 .MACRO C000000-B        ::= BSPFI(C000000)    !IN R01
1384 .MACRO C177777-A        ::= ASPFI(C177777)    !IN R03
1385 .MACRO C177777-B        ::= BSPFI(C177777)    !IN R03
1386 .MACRO C125252-A        ::= ASPFI(C125252)    !IN R05
1387 .MACRO C125252-B        ::= BSPFI(C125252)    !IN R05
1388 .MACRO C052525-A        ::= ASPFI(C052525)    !IN R07
1389 .MACRO C052525-B        ::= BSPFI(C052525)    !IN R07
1390 .MACRO C000001-A        ::= ASPFI(C000001)    !IN R11
1391 .MACRO C000001-B        ::= BSPFI(C000001)    !IN R11
1392 .MACRO C100000-A        ::= ASPFI(C100000)    !IN R13
1393 .MACRO C100000-B        ::= BSPFI(C100000)    !IN R13
1394 .MACRO C000200-A        ::= ASPFI(C000200)    !IN R15
1395 .MACRO C000200-B        ::= BSPFI(C000200)    !IN R15

```

```

1396
1397
1398
1399 !.PAGE=====

```

```

1401 .TOC *      WRITING THE C SCRATCH PAD
1402
1403 .:
1404 .:   WRITE DATA ON BUSDIN (ACTUALLY DMUX OUTPUT) INTO ADDRESSED CSP LOCATION
1405 .:   [SEE BELOW] DURING P3

```

```

1405 !
1406 .MACRO WR-CSP          ::= WRCSP/YES
1407
1408
1409
1410 .TOC *      CSP IMPLIED ADDRESSING
1411
1412     ENABLE FOR INPUT/OUTPUT (READ AND/OR WRITE) ONTO BUS-B ONLY A SPECIFIC CSP LOCATION,
1413     WHERE THE ADDRESS IS DETERMINED AS FOLLOWS:
1414
1415         CSPADDR<3:0>H = -[ 0 * 0 * BSEL<1>H * BSEL<0>H ]
1416
1417 .MACRO CSPB(XX)        ::= BEN/BASCON,          !USE IMMEDIATE MODE
1418                         BSEL/@XX              !WHICH ONE
1419
1420
1421 .TOC *      CSP DIRECT ADDRESSING
1422
1423     ENABLE FOR INPUT/OUTPUT (READ AND/OR WRITE) ONTO BUS-B ONLY A SPECIFIC CSP LOCATION,
1424     WHERE THE ADDRESS IS DETERMINED AS FOLLOWS:
1425
1426         CSPADDR<3:0>H = -JWORD<23:20> H
1427
1428 .MACRO CSPD(XX)        ::= BEN/CSP,            !USE CSP-ADDR MODE
1429                         CSPADDR/@XX         !WHICH ONE
1430
1431
1432
1433
1434 !.PAGE=====
1435
1436 .TOC *      SHIFT TREE SPECIFICATION
1437 !N.B. MAY REQUIRE PRIOR SETUP OF RES-REGISTER FOR SHIFT END MUX SELECTION CONTROL
1438 ! (EG, WHEN ASEL/LEFT-1 IS USED).
1439
1440 .TOC *      ENABLED ONTO BUS A
1441 .MACRO D-RIGHT-14      ::= REN/CMUX, AMUX/RIGHT-8, BMUX/RIGHT-4, ASEL/RIGHT-2
1442 .MACRO D-RIGHT-13      ::= REN/CMUX, AMUX/RIGHT-8, BMUX/RIGHT-4, ASEL/RIGHT-1
1443 .MACRO D-RIGHT-12      ::= REN/CMUX, AMUX/RIGHT-8, BMUX/RIGHT-4, ASEL/DIRECT
1444 .MACRO D-RIGHT-11      ::= REN/CMUX, AMUX/RIGHT-8, BMUX/RIGHT-4, ASEL/LEFT-1      !SENDMUX SETUP
1445 .MACRO D-RIGHT-10      ::= REN/CMUX, AMUX/RIGHT-8, BMUX/DIRECT, ASEL/RIGHT-2
1446 .MACRO D-RIGHT-9       ::= REN/CMUX, AMUX/RIGHT-8, BMUX/DIRECT, ASEL/RIGHT-1
1447 .MACRO D-RIGHT-8       ::= REN/CMUX, AMUX/RIGHT-8, BMUX/DIRECT, ASEL/DIRECT
1448 .MACRO D-RIGHT-7       ::= REN/CMUX, AMUX/RIGHT-8, BMUX/DIRECT, ASEL/LEFT-1      !SENDMUX SETUP
1449 .MACRO D-RIGHT-6       ::= REN/CMUX, AMUX/DIRECT, BMUX/RIGHT-4, ASEL/RIGHT-2
1450 .MACRO D-RIGHT-5       ::= REN/CMUX, AMUX/DIRECT, BMUX/RIGHT-4, ASEL/RIGHT-1
1451 .MACRO D-RIGHT-4       ::= REN/CMUX, AMUX/DIRECT, BMUX/RIGHT-4, ASEL/DIRECT
1452 .MACRO D-RIGHT-3       ::= REN/CMUX, AMUX/DIRECT, BMUX/RIGHT-4, ASEL/LEFT-1      !SENDMUX SETUP
1453 .MACRO D-RIGHT-2       ::= REN/CMUX, AMUX/DIRECT, BMUX/DIRECT, ASEL/RIGHT-2
1454 .MACRO D-RIGHT-1       ::= REN/CMUX, AMUX/DIRECT, BMUX/DIRECT, ASEL/RIGHT-1
1455 .MACRO D-NO-SHIFT      ::= REN/CMUX, AMUX/DIRECT, BMUX/DIRECT, ASEL/DIRECT
1456 .MACRO D-DIRECT        ::= D-NO-SHIFT
1457 .MACRO D-LEFT-1        ::= REN/CMUX, AMUX/DIRECT, BMUX/DIRECT, ASEL/LEFT-1      !SENDMUX SETUP
1458 .MACRO D-SWAB          ::= REN/CMUX, AMUX/SWAB, BMUX/DIRECT, ASEL/DIRECT

```

```

1459 .MACRO D-SWAB-RIGHT-3      ::= REN/CMUX, AMUX/SWAB, BMUX/RIGHT-4, ASEL/LEFT-1      !SENDMUX SETUP
1460 .MACRO D-SWAB-LEFT-1     ::= REN/CMUX, AMUX/SWAB, BMUX/DIRECT, ASEL/LEFT-1      !SENDMUX SETUP
1461 .MACRO D-SIGNEXT          ::= REN/CMUX, AMUX/SIGNEXT, BMUX/DIRECT, ASEL/DIRECT
1462 .MACRO D-SIGNEXT-RIGHT-1 ::= REN/CMUX, AMUX/SIGNEXT, BMUX/DIRECT, ASEL/RIGHT-1
1463 .MACRO D-SIGNEXT-LEFT-1  ::= REN/CMUX, AMUX/SIGNEXT, BMUX/DIRECT, ASEL/LEFT-1      !SENDMUX SETUP
1464 .MACRO NO-SHIFT           ::= REN/CMUX, BMUX/DIRECT, ASEL/DIRECT
1465 .MACRO DIRECT             ::= NO-SHIFT
1466 .MACRO COUNT#D[HI]        ::= REN/CMUX, AMUX/COUNTER#D[HI], BMUX/DIRECT, ASEL/DIRECT
1467 .MACRO COUNT#D[LO]        ::= REN/CMUX, AMUX/COUNTER#D[LO], BMUX/DIRECT, ASEL/DIRECT

```

```

1468
1469
1470

```

```

1471 .TOC * FIRST TWO LEVELS ONLY [AMUX, BMUX]
1472 !N.B.: FOR USE WHEN SHIFTING SR RIGHT, SR<15> <- BMUX<00>
1473 .MACRO D-DIRECT [BMUX]    ::= AMUX/DIRECT, BMUX/DIRECT
1474
1475

```

```

1476

```

```

1477 !.PAGE=====

```

```

1478
1479

```

```

1480 .TOC * ALU FUNCTIONS
1481 ! [SEE FIELD DESCRIPTION OF "ALU" FOR FULL DESCRIPTION]
1482 .MACRO ZERO                ::= ALU/ZERO
1483 .MACRO A-XOR-B             ::= ALU/A-XOR-B
1484 .MACRO B                   ::= ALU/B
1485 .MACRO A-AND-B             ::= ALU/A-AND-B
1486 .MACRO A-IOR-B             ::= ALU/A-IOR-B
1487 .MACRO A                   ::= ALU/A
1488 .MACRO NOT-A               ::= ALU/NOT-A
1489 .MACRO NOT-A-AND-B         ::= ALU/NOT-A-AND-B
1490 .MACRO A-AND-NOT-B         ::= ALU/A-AND-NOT-B
1491 .MACRO DIVIDE              ::= ALU/DIVIDE
1492 .MACRO A-PLUS-B            ::= ALU/A-PLUS-B
1493 .MACRO A-MINUS-B           ::= ALU/A-MINUS-B
1494 .MACRO A-PLUS-B-PLUS-PS[C] ::= ALU/A-PLUS-B-PLUS-PS[C]
1495 .MACRO A-PLUS-B-PLUS-D[C]  ::= ALU/A-PLUS-B-PLUS-D[C]
1496 .MACRO A-PLUS-NOT-B-PLUS-D[C] ::= ALU/A-PLUS-NOT-B-PLUS-D[C]
1497 .MACRO A-PLUS-B-PLUS-1     ::= ALU/A-PLUS-B-PLUS-1

```

```

1498
1499

```

```

1500 .TOC * COUT GENERATION
1501 ! [SEE FIELD DESCRIPTION OF "COUT" FOR FULL DESCRIPTION]
1502 .MACRO COUT+CIN            ::= COUT/CIN
1503 .MACRO COUT+PS[C]         ::= COUT/PS[C]
1504 .MACRO COUT+ALU00         ::= COUT/ALU00
1505 .MACRO COUT+ALU07         ::= COUT/ALU07
1506 .MACRO COUT+ALU15         ::= COUT/ALU15
1507 .MACRO COUT+COUT07        ::= COUT/COUT07
1508 .MACRO COUT+COUT15        ::= COUT/COUT15
1509 .MACRO COUT+D[C]          ::= COUT/D[C]

```

```

1510
1511
1512

```

1513 !.PAGE=====

1514  
1515 .TOC \* CLOCKS  
1516

1517  
1518  
1519 .TOC \* BASIC REGISTER CLOCKS (D, SR, BA, CC)  
1520 .MACRO CLK-D ::= CLKD/YES !MUST SPECIFY P2 T OR P3 T  
1521 .MACRO CLK-SR ::= CLKSr/YES !MUST SPECIFY P2 T OR P3 T  
1522 .MACRO CLK-BA ::= CLKBA/YES !AT P1 T ONLY  
1523 .MACRO SET-CC ::= SCC/YES !SETUP HERE, CLOCKED AT P2 T \*\*OF NEXT UWORD\*\* ONLY  
1524 .MACRO CLK-CC ::= NULL !IN NEXT UWORD, FOR DOCUMENTATION  
1525  
1526

1527  
1528 .TOC \* REDEFINED FROM SP REWRITE FIELD (RES, COUNTER)  
1529 .MACRO LOAD-RES ::= MOD/LOADREG,LOADRES/YES !AT P2 T ONLY, FROM B-BUS<14:11>  
1530 .MACRO LOAD-COUNTER ::= MOD/LOADREG,LOADCOUNT/YES !DURING ENTIRE UWORD, FROM B-BUS<7:0>  
1531

1532  
1533  
1534 .TOC \* RES REGISTER CONTROL VALUES (FROM EMIT)  
1535 !LOADED VIA: EMIT<14:11> -> CSP[XX]<14:11> -> B-BUS<14:11> -> RES<3:0>  
1536 .MACRO SENDMUX-0123-SEL ::= EMIT14/1 !FOR SHIFT TREE  
1537 .MACRO SENDMUX-4567-SEL ::= EMIT14/0 !FOR SHIFT TREE  
1538 .MACRO SR-LOAD ::= EMIT13/0,EMIT12/0 !FOR SR/GUARD  
1539 .MACRO SR-LEFT ::= EMIT13/0,EMIT12/1 !FOR SR/GUARD  
1540 .MACRO SR-RIGHT ::= EMIT13/1,EMIT12/0 !FOR SR/GUARD  
1541 .MACRO SR-NOP ::= EMIT13/1,EMIT12/1 !FOR SR/GUARD  
1542 .MACRO GUARD-EN ::= EMIT11/1 !FOR SR/GUARD  
1543 .MACRO GUARD-DIS ::= EMIT11/0 !FOR SR/GUARD  
1544  
1545

1546  
1547  
1548 .TOC \* CC CONTROL (FROM EMIT)  
1549 !USED VIA: BUS-U37-H -> EMIT07-H -> MODIFY-V(1)-H  
1550 .MACRO MODIFY-VBIT ::= EMIT07/1  
1551 .MACRO NOT-MODIFY-VBIT ::= EMIT07/0  
1552  
1553

1554  
1555 !.PAGE=====

1556  
1557 .TOC \* BUS CONTROL MACROS  
1558 .MACRO DATI-CLKIR ::= BEGIN/YES,SELECT/BUS,BUSCODE/DATI-CLKIR  
1559 .MACRO DATI-NOINT ::= BEGIN/YES,SELECT/BUS,BUSCODE/DATI-NOINT  
1560 .MACRO DATI ::= BEGIN/YES,SELECT/BUS,BUSCODE/DATI  
1561 .MACRO DATI[P] ::= BEGIN/YES,SELECT/BUS,BUSCODE/DATI[P] !WITH ALTER/ALLOWED  
1562 .MACRO DATO ::= BEGIN/YES,SELECT/BUS,BUSCODE/DATO  
1563 .MACRO DATIB ::= BEGIN/YES,SELECT/BUS,BUSCODE/DATIB  
1564 .MACRO DATIB[P] ::= BEGIN/YES,SELECT/BUS,BUSCODE/DATIB[P] !WITH ALTER/ALLOWED  
1565 .MACRO DATIP ::= BEGIN/YES,SELECT/BUS,BUSCODE/DATIP  
1566 .MACRO DATOB ::= BEGIN/YES,SELECT/BUS,BUSCODE/DATOB

```

1567 .MACRO INVALIDATE ::= BEGIN/YES,SELECT/BUS,BUSCODE/INVALIDATE
1568
1569
1570
1571 .TOC * KT/KJ CONTROL FUNCTIONS
1572
1573 !THESE BITS ACTUALLY ARISE OUT OF THE BASE MACHINE EXTENSION ROMS,
1574 !AND AS SUCH ARENT DIRECTLY ACCESSIBLE FROM THE DCS. THEY ARE
1575 !INCLUDED HERE ONLY FOR DOCUMENTATION PURPOSES.
1576
1577 .MACRO KJ-ENABLE ::= KJ/ONE
1578
1579 .MACRO MAINTENANCE ::= UKTEN/ONE
1580 .MACRO CURRENT-MODE ::= UKT01/ONE,UKT00/ZERO
1581 .MACRO KERNAL-MODE ::= UKT01/ONE,UKT00/ONE
1582 .MACRO MT-MODE ::= UKT01/ZERO,UKT00/ONE
1583 .MACRO MF-MODE ::= UKT01/ZERO,UKT00/ZERO
1584
1585
1586
1587 !.PAGE=====
1588
1589 .TOC * UCON CONTROL MACROS
1590 .MACRO SET-UCON-CONTROL ::= BEGIN/YES,SELECT/UCON,UCON-LOAD/YES !LOAD UCON CONTROL REGISTER AT PO
1591 .MACRO UCON-OPERATION ::= BEGIN/YES,SELECT/UCON,UCON-XFER/YES !PERFORM UCON OPERATION
1592
1593
1594
1595 .TOC * PROCESSOR UCON CONTROL SETUP
1596 .MACRO UCON-PROC ::= UCON-SEL-PROC/YES !SELECT PROCESSOR
1597 .MACRO EN-CLK-IR[15-00] ::= UCON15/1 !ENABLE OPERATIONS
1598 .MACRO EN-CLK-PS[15-12] ::= UCON14/1
1599 .MACRO EN-CLK-FLAG[8-0] ::= UCON13/1
1600 .MACRO EN-CLK-FPS[7-4] ::= UCON12/1
1601 .MACRO EN-CLK-PS[7-4] ::= UCON11/1
1602 .MACRO EN-CLK-PS[3-0] ::= UCON10/1
1603 .MACRO EN-CLK-UBREAK[11-00] ::= UCON09/1
1604 !UCON<8:7> ARE NOT USED IN PROCESSOR CONTROL
1605 .MACRO BUSDIN+EMIT[15-00] ::= UCON06/0,UCON05/0 !HBMUX SELECT
1606 .MACRO BUSDIN+CUA[14-03] ::= UCON06/0,UCON05/1
1607 .MACRO BUSDIN+PS[15-00] ::= UCON06/1,UCON05/0
1608 .MACRO BUSDIN+FLAG[8-0]#FPS[7-0] ::= UCON06/1,UCON05/1
1609
1610
1611
1612
1613 .TOC * DCS/WCS/ECS CONTROL
1614 .MACRO UCON-DCS ::= UCON-SEL-WCS/YES !SELECT DCS
1615 .MACRO BUSDIN+TNUA[11-00] ::= UCON14/0 !DCS BUSDIN MUX SEL
1616 .MACRO BUSDIN+ERR#EOP#ERRCOD[11-00] ::= UCON14/1
1617 .MACRO START-DCS ::= UCON15/1
1618
1619
1620

```

```

1621 .TOC *          CACHE/KT UCON CONTROL
1622 .MACRO UCON-CACHE-KT          ::= UCON-SEL-CACHEKT/YES          !SELECT CACHE / KT UCON FUNCTION
1623 !UCON<15> NOT USED HERE
1624 .MACRO EN-KT-NO-RELOCATE      ::= UCON14/1          !INHIBIT KT FROM ANY RELOCATION OF BA -> PBA
1625 .MACRO BUSDIN+BUS-INTERNAL-ADDR[15-00] ::= UCON13/0,UCON12/1 !FROM INTERNAL ADDR RCM
1626 .MACRO BUSDIN+CPU-INTERNAL-ADDR[15-00] ::= UCON13/1,UCON12/1 !DITTO ...
1627 .MACRO BUSDIN+MMR2[15-00]      ::= UCON11/1,UCON09/0 !VIRTUAL PC
1628 .MACRO BUSDIN+CACHE-STATUS[15-00] ::= UCON11/1,UCON09/1 !CACHE INFO
1629 .MACRO BUSDIN+KT-SEL          ::= UCON10/1          !FOR PAR-S, PDR-S ETC
1630 .MACRO KT-WRITE-HIGH         ::= UCON08/1          !WRITE REGISTER <15:08>
1631 .MACRO KT-WRITE-LOW          ::= UCON07/1          !WRITE REGISTER <07:00>
1632 .MACRO KT-WRITE              ::= UCON08/1,UCON07/1 !WRITE REGISTER <15:00>
1633 .MACRO KT-SEL-SLR#CCR        ::= UCON06/0,UCON05/0 !SELECT KT-MUX OUTPUT
1634 .MACRO KT-SEL-MMR0          ::= UCON06/0,UCON05/1
1635 .MACRO KT-SEL-PDR           ::= UCON06/1,UCON05/0
1636 .MACRO KT-SEL-PAR           ::= UCON06/1,UCON05/1
1637
1638
1639
1640
1641 .TOC *          I/O UCON CONTROL
1642 .MACRO UCON-I-0              ::= UCCN-SEL-I-0/YES          !SELECT I-0 CONTROL
1643
1644
1645
1646 .TOC *          BUS CONTROL
1647 .MACRO EN-LOAD-DBUF[15-00]    ::= UCON15/1          !EN LOAD DBUF AT P3
1648 .MACRO BUSDIN+DBUF[15-00]    ::= UCON15/1          !DBUF ON BUSDIN
1649 .MACRO EN-STATUS-MUX        ::= UCON15/0          !STATUS-MUX ENABLE ON BUSDIN
1650 !UCON<14:11> ARE NOT USED IN UCON BUS CONTROL
1651 .MACRO BUSDIN+SERVICE[15-00] ::= UCON10/0,UCON09/1
1652 .MACRO BUSDIN+JAM[15-00]     ::= UCON10/1,UCON09/0
1653 .MACRO BUSDIN+PBA[15-00]     ::= UCON10/1,UCON09/1
1654 .MACRO DMUX+CACHEDATA[15-00] ::= UCON08/1
1655 .MACRO EN-BC-FCN-0          ::= UCON07/0,UCON06/0,UCON05/0 !SELECT BUS CONTROL FUNCTION
1656 .MACRO EN-START-DELAY       ::= UCON07/0,UCON06/0,UCON05/1
1657 .MACRO EN-CLR-JAM-ERRORS    ::= UCON07/0,UCON06/1,UCON05/0
1658 .MACRO EN-CLR-NPR-TIMEOUT   ::= UCON07/0,UCON06/1,UCON05/1
1659 .MACRO EN-CLR-PWR-FAIL     ::= UCON07/1,UCON06/0,UCON05/0
1660 .MACRO EN-CLR-YELLOW-ZONE   ::= UCON07/1,UCON06/0,UCON05/1
1661 .MACRO EN-ALLOW-BG[1]H     ::= UCON07/1,UCON06/1,UCON05/0
1662 .MACRO EN-BUS-INIT-UCON     ::= UCON07/1,UCON06/1,UCON05/1
1663
1664
1665
1666 .TOC *          CONSOLE I-0
1667 .MACRO EN-CONSOLE-COMMAND    ::= UCON15/0,UCON14/0          !SETS UP UCON I-0 BITS FOR CONSOLE COMMANDS
1668 !ALSO SELECTS STATUS-MUX ON BUSDIN
1669 .MACRO EN-CNSL-NOP          ::= UCON13/0,UCON12/0,UCON11/0 !ENABLE CONSOLE NO OPERATION
1670 .MACRO EN-CLR-COUNTR        ::= UCON13/0,UCON12/0,UCON11/1 !ENABLE CLEAR DIGIT PAIR COUNTER
1671 .MACRO EN-INCR-COUNTR       ::= UCON13/0,UCON12/1,UCON11/0 !ENABLE BUMP TO NEXT DIGIT PAIR
1672 .MACRO EN-CLR-CNSL-SRVC     ::= UCON13/0,UCON12/1,UCON11/1 !ENABLE CLEAR CONSOLE SERVICE ROST FLOP
1673 .MACRO EN-STRB-DISP         ::= UCON13/1,UCON12/0,UCON11/0 !ENABLE WRITE DIGIT PAIR TO DISPLAY LATCH
1674 .MACRO EN-CLR-CNSL         ::= UCON13/1,UCON12/0,UCON11/1 !ENABLE CLEAR CONSOLE LED

```



```

1675 .MACRO EN-SET-CNSL          ::= UCON13/1,UCON12/1,UCON11/0      !ENABLE SET CONSOLE LED
1676 .MACRO EN-SET-DP          ::= UCON13/1,UCON12/1,UCON11/1      !ENABLE SET ALL DP LEDS
1677 .MACRO BUSDIN<CONSOLE(06-00) ::= UCON10/0,UCON09/0          !STATUS-MUX SELECT
1678 !UCON<8:5> ARE NOT USED IN UCON CONSOLE CONTROL
1679
1680
1681
1682 .TOC *          REMOTE CONSOLE INTERFACE
1683 !N.B.: "EN CONSOLE COMMAND" DOES NOT APPLY TO REMOTE CONSOLE
1684 .MACRO EN-REMSTRB          ::= UCON14/1          !EN REMOTE CONSOLE STROBE
1685 .MACRO EN-REMCODE1        ::= UCON12/1          !EN SPECIAL CODE 1
1686 .MACRO EN-REMCODE0        ::= UCON11/1          !EN SPECIAL CODE 0
1687
1688
1689
1690 !.PAGE=====
1691
1692 .TOC *          DCS ROM EXTENSION MACROS
1693
1694 .TOC *          GENERAL FUNCTIONS
1695 .MACRO LOAD-ENUA(MICROADDR) ::= LOAD-ENUA-ERRCOD/YES,          !SPECIFY LOAD
1696                               ENUA/2MICROADDR                !AND VALUE
1697 .MACRO LOAD-ERROR(MICROADDR) ::= LOAD-ENUA-ERRCOD/YES          !OR EFFECT ONLY - ALWAYS ACCOMPANIES ABOVE
1698 .MACRO BUMP-VERIFY          ::= VERIFY/BUMP                    !BUMP VERIFY COUNTER WHEN IN SELF TEST MODE
1699 .MACRO SIGNAL-EOP          ::= EOP/SIGNAL                      !SIGNAL END OF PASS
1700 .MACRO DCS-CTR(XX)         ::= _LOAD-DCS-CTR/YES,            !SPECIFY LOAD COUNTER (DCS!)
1701                               CTR/2XX                          !AND VALUE
1702
1703
1704
1705 .TOC *          DAD<1:0> BIT FUNCTIONS
1706 .MACRO NO-DAD              ::= DAD/NO-DAD                    !DON'T ASSERT DAD BITS
1707 .MACRO FIRST-1-OR-2        ::= DAD/FIRST-1-OR-2             !SELECT CSP CONSTANT 1/2, FIRST USE
1708 .MACRO SECOND-1-OR-2       ::= DAD/SECOND-1-OR-2            !SELECT CSP CONSTANT 1/2, SECOND USE
1709 .MACRO WRITE-BYTE         ::= DAD/WRITE-BYTE                !BYTE WRITE ENABLE TO ASP/BSP
1710 .MACRO BYTE-WRITE         ::= DAD/WRITE-BYTE                !
1711
1712
1713
1714 .TOC *          DIAGNOSTIC MODE BUT ENABLES
1715
1716 .MACRO BUTD(SCOPE)         ::= SCOPE/ENABLED                !ENABLE SCOPE LOOPING CHECK
1717 .MACRO BUTD(ERROR)         ::= SCOPE/ENABLED                !FORCES NUAD0=0 IF ERROR[1]H SET
1718
1719 .MACRO BUTD(VERIFY-MODE)   ::= NULL                          !CHECK IMPLICITLY FOR VERIFY MODE
1720 !'SIGNAL EOP' MUST BE PRESENT IN SAME WORD
1721 !FORCES NUAD1=0 IF VERIFY SWITCH SET
1722
1723 .MACRO BUTD(EOP-OVERFLOW)  ::= NULL                          !AFTER 'SIGNAL EOP' GIVEN, ADDRESS IS FORCED
1724 !TO (4000) IF HARDWARE EOP/VERIFY COUNTER
1725 !HAS NOT YET OVERFLOWED, FORCING ANOTHER
1726 !PASS. ELSE ADDRESS IN UPF IS TAKEN UNMODIFIED.
1727
1728

```

```

1729
1730 !.PAGE=====
1731
1732 .TOC * MICROBRANCH FIELD MACROS
1733 ! [SEE <UBF> FIELD DESCRIPTION FOR FULL INFO]
1734
1735 .MACRO BUT(XX) ::= UBF/2XX ! INACTIVE, FULL WIDTH
1736 .MACRO BUTR(XX) ::= UBF/2XX ! INACTIVE, RESTRICTED WIDTH
1737
1738 .MACRO BUTA(XX) ::= UBF/2XX ! ACTIVE, FULL WIDTH
1739 .MACRO BUTRA(XX) ::= UBF/2XX ! ACTIVE, RESTRICTED WIDTH
1740
1741 .MACRO TEST(XX) ::= MULTIPLE/2XX ! FOR BUTR(MULTIPLE) SETUP
1742 .MACRO BUTM(XX) ::= MULTIPLE/2XX,UBF/2XX ! A MULTIPLE BUTR
1743
1744
1745
1746 !.PAGE=====
1747
1748 .TOC * MISCELLANEOUS
1749
1750 .TOC * OTHER SOURCES ENABLED FOR A-BUS
1751 .MACRO SR ::= AEN/XMUX,ASELO/SR
1752 .MACRO FLTPT ::= AEN/XMUX,ASELO/FLTPT
1753
1754
1755
1756 .TOC * PAGING, RETURN REGISTER
1757
1758 !PAGE FIELD ONLY:
1759 .MACRO PAGE(X) ::= NEXT-PAGE/2X
1760
1761 !PAGE FIELD AND BUT(SUBR B):
1762 .MACRO GOTO-PAGE(X) ::= NEXT-PAGE/2X,UBF/SUBR-B
1763
1764 !RETURN REGISTER <- D<14:03>, PAGE <- EMIT<02:00> ON BUTA(SUBR-A)
1765 .MACRO RETURN+D[14-03] ::= UBF/SUBR-A
1766
1767
1768
1769 !.PAGE=====
1770
1771 .TOC * ADVANCED OPERATIONS
1772
1773
1774
1775 .TOC * DATA INTO CSP, AT P3 ONLY
1776
1777 !N.B.: BUSDIN IS ANY BUT EMIT (OVERLAPS BSEL<1:0>)
1778 .MACRO CSPB[14]+BUSDIN ::= CSPB(B14),WR-CSP
1779 .MACRO CSPB[15]+BUSDIN ::= CSPB(B15),WR-CSP
1780 .MACRO CSPB[16]+BUSDIN ::= CSPB(B16),WR-CSP
1781 .MACRO CSPB[17]+BUSDIN ::= CSPB(B17),WR-CSP
1782

```

```

1783      !N.B.: SETS WHATEVER IS ON BUSDIN
1784      .MACRO  CSPD[00]←BUSDIN      ::=  CSPD(000),WR-CSP
1785      .MACRO  CSPD[01]←BUSDIN      ::=  CSPD(001),WR-CSP
1786      .MACRO  CSPD[02]←BUSDIN      ::=  CSPD(002),WR-CSP
1787      .MACRO  CSPD[03]←BUSDIN      ::=  CSPD(003),WR-CSP
1788      .MACRO  CSPD[04]←BUSDIN      ::=  CSPD(004),WR-CSP
1789      .MACRO  CSPD[05]←BUSDIN      ::=  CSPD(005),WR-CSP
1790      .MACRO  CSPD[06]←BUSDIN      ::=  CSPD(006),WR-CSP
1791      .MACRO  CSPD[07]←BUSDIN      ::=  CSPD(007),WR-CSP
1792      .MACRO  CSPD[10]←BUSDIN      ::=  CSPD(010),WR-CSP
1793      .MACRO  CSPD[11]←BUSDIN      ::=  CSPD(011),WR-CSP
1794      .MACRO  CSPD[12]←BUSDIN      ::=  CSPD(012),WR-CSP
1795      .MACRO  CSPD[13]←BUSDIN      ::=  CSPD(013),WR-CSP
1796      .MACRO  CSPD[14]←BUSDIN      ::=  CSPD(014),WR-CSP
1797      .MACRO  CSPD[15]←BUSDIN      ::=  CSPD(015),WR-CSP
1798      .MACRO  CSPD[16]←BUSDIN      ::=  CSPD(016),WR-CSP
1799      .MACRO  CSPD[17]←BUSDIN      ::=  CSPD(017),WR-CSP
1800
1801      !N.B.: REQUIRED THAT BUSDIN←EMIT[15-00] PREVIOUSLY SET JP
1802      .MACRO  CSPD[00]←EMIT        ::=  CSPD(000),WR-CSP
1803      .MACRO  CSPD[01]←EMIT        ::=  CSPD(001),WR-CSP
1804      .MACRO  CSPD[02]←EMIT        ::=  CSPD(002),WR-CSP
1805      .MACRO  CSPD[03]←EMIT        ::=  CSPD(003),WR-CSP
1806      .MACRO  CSPD[04]←EMIT        ::=  CSPD(004),WR-CSP
1807      .MACRO  CSPD[05]←EMIT        ::=  CSPD(005),WR-CSP
1808      .MACRO  CSPD[06]←EMIT        ::=  CSPD(006),WR-CSP
1809      .MACRO  CSPD[07]←EMIT        ::=  CSPD(007),WR-CSP
1810      .MACRO  CSPD[10]←EMIT        ::=  CSPD(010),WR-CSP
1811      .MACRO  CSPD[11]←EMIT        ::=  CSPD(011),WR-CSP
1812      .MACRO  CSPD[12]←EMIT        ::=  CSPD(012),WR-CSP
1813      .MACRO  CSPD[13]←EMIT        ::=  CSPD(013),WR-CSP
1814      .MACRO  CSPD[14]←EMIT        ::=  CSPD(014),WR-CSP
1815      .MACRO  CSPD[15]←EMIT        ::=  CSPD(015),WR-CSP
1816      .MACRO  CSPD[16]←EMIT        ::=  CSPD(016),WR-CSP
1817      .MACRO  CSPD[17]←EMIT        ::=  CSPD(017),WR-CSP
1818
1819
1820
1821      .TOC *      MISC CONSTANTS INTO ASP, BSP, AT P2-T * P3
1822
1823      .MACRO  A#BSPHI[C100000]←D    ::=  ASP(C100000),BSP(C100000),WR(AB,H,B)
1824      .MACRO  A#BSPHI[C000200]←D    ::=  ASP(C000200),BSP(C000200),WR(AB,H,B)
1825      .MACRO  A#BSPHI[C000000]←D    ::=  ASP(C000000),BSP(C000000),WR(AB,H,B)
1826      .MACRO  A#BSPHI[C177777]←D    ::=  ASP(C177777),BSP(C177777),WR(AB,H,B)
1827      .MACRO  A#BSPHI[C000001]←D    ::=  ASP(C000001),BSP(C000001),WR(AB,H,B)
1828      .MACRO  A#BSPHI[C052525]←D    ::=  ASP(C052525),BSP(C052525),WR(AB,H,B)
1829      .MACRO  A#BSPHI[C125252]←D    ::=  ASP(C125252),BSP(C125252),WR(AB,H,B)
1830
1831      .MACRO  A#BSPHI[C100000]←D-[B] ::=  BSP(C100000),WR(AB,H,B)
1832      .MACRO  A#BSPHI[C000200]←D-[B] ::=  BSP(C000200),WR(AB,H,B)
1833      .MACRO  A#BSPHI[C000000]←D-[B] ::=  BSP(C000000),WR(AB,H,B)
1834      .MACRO  A#BSPHI[C177777]←D-[B] ::=  BSP(C177777),WR(AB,H,B)
1835      .MACRO  A#BSPHI[C000001]←D-[B] ::=  BSP(C000001),WR(AB,H,B)
1836      .MACRO  A#BSPHI[C052525]←D-[B] ::=  BSP(C052525),WR(AB,H,B)

```

```

1837 .MACRO A#BSPHI(C125252)+D-(B) ::= BSP(C125252),WR(AB,H,B)
1838
1839 .MACRO A#BSPHI(C100000)+D-(A) ::= ASP(C100000),WR(AB,H,A)
1840 .MACRO A#BSPHI(C000200)+D-(A) ::= ASP(C000200),WR(AB,H,A)
1841 .MACRO A#BSPHI(C000000)+D-(A) ::= ASP(C000000),WR(AB,H,A)
1842 .MACRO A#BSPHI(C177777)+D-(A) ::= ASP(C177777),WR(AB,H,A)
1843 .MACRO A#BSPHI(C000001)+D-(A) ::= ASP(C000001),WR(AB,H,A)
1844 .MACRO A#BSPHI(C052525)+D-(A) ::= ASP(C052525),WR(AB,H,A)
1845 .MACRO A#BSPHI(C125252)+D-(A) ::= ASP(C125252),WR(AB,H,A)
1846
1847
1848
1849 .TOC * DATA INTO ASP, BSP, AT P2 T * P3
1850
1851 .MACRO ASPL0[17]+CSPB(XX) ::= B,ASPL0(R17),CSPB(XX),CLK-D,P2-T,WR(A,L,A)
1852 .MACRO ASPL0[17]+CSPD(XX) ::= B,ASPL0(R17),CSPD(XX),CLK-D,P2-T,WR(A,L,A)
1853 .MACRO PC+D ::= PC-A,WR(AB,L,A)
1854 .MACRO RS+D ::= RS-A,WR(AB,L,A)
1855
1856 .MACRO ASPL0[00]+D ::= ASP(R00),WR(A,L,A)
1857 .MACRO ASPL0[01]+D ::= ASP(R01),WR(A,L,A)
1858 .MACRO ASPL0[02]+D ::= ASP(R02),WR(A,L,A)
1859 .MACRO ASPL0[03]+D ::= ASP(R03),WR(A,L,A)
1860 .MACRO ASPL0[04]+D ::= ASP(R04),WR(A,L,A)
1861 .MACRO ASPL0[05]+D ::= ASP(R05),WR(A,L,A)
1862 .MACRO ASPL0[06]+D ::= ASP(R06),WR(A,L,A)
1863 .MACRO ASPL0[07]+D ::= ASP(R07),WR(A,L,A)
1864 .MACRO ASPL0[10]+D ::= ASP(R10),WR(A,L,A)
1865 .MACRO ASPL0[11]+D ::= ASP(R11),WR(A,L,A)
1866 .MACRO ASPL0[12]+D ::= ASP(R12),WR(A,L,A)
1867 .MACRO ASPL0[13]+D ::= ASP(R13),WR(A,L,A)
1868 .MACRO ASPL0[14]+D ::= ASP(R14),WR(A,L,A)
1869 .MACRO ASPL0[15]+D ::= ASP(R15),WR(A,L,A)
1870 .MACRO ASPL0[16]+D ::= ASP(R16),WR(A,L,A)
1871 .MACRO ASPL0[17]+D ::= ASP(R17),WR(A,L,A)
1872
1873 .MACRO ASPHI[00]+D ::= ASP(R00),WR(A,H,A)
1874 .MACRO ASPHI[01]+D ::= ASP(R01),WR(A,H,A)
1875 .MACRO ASPHI[02]+D ::= ASP(R02),WR(A,H,A)
1876 .MACRO ASPHI[03]+D ::= ASP(R03),WR(A,H,A)
1877 .MACRO ASPHI[04]+D ::= ASP(R04),WR(A,H,A)
1878 .MACRO ASPHI[05]+D ::= ASP(R05),WR(A,H,A)
1879 .MACRO ASPHI[06]+D ::= ASP(R06),WR(A,H,A)
1880 .MACRO ASPHI[07]+D ::= ASP(R07),WR(A,H,A)
1881 .MACRO ASPHI[10]+D ::= ASP(R10),WR(A,H,A)
1882 .MACRO ASPHI[11]+D ::= ASP(R11),WR(A,H,A)
1883 .MACRO ASPHI[12]+D ::= ASP(R12),WR(A,H,A)
1884 .MACRO ASPHI[13]+D ::= ASP(R13),WR(A,H,A)
1885 .MACRO ASPHI[14]+D ::= ASP(R14),WR(A,H,A)
1886 .MACRO ASPHI[15]+D ::= ASP(R15),WR(A,H,A)
1887 .MACRO ASPHI[16]+D ::= ASP(R16),WR(A,H,A)
1888 .MACRO ASPHI[17]+D ::= ASP(R17),WR(A,H,A)
1889
1890 .MACRO BSPL0[00]+D ::= BSP(R00),WR(B,L,B)

```

1891	.MACRO	BSPLO(01)+D	.....	BSP(R00),WR(00),L(00)
1892	.MACRO	BSPLO(02)+D	.....	BSP(R01),WR(00),L(00)
1893	.MACRO	BSPLO(03)+D	.....	BSP(R02),WR(00),L(00)
1894	.MACRO	BSPLO(04)+D	.....	BSP(R03),WR(00),L(00)
1895	.MACRO	BSPLO(05)+D	.....	BSP(R04),WR(00),L(00)
1896	.MACRO	BSPLO(06)+D	.....	BSP(R05),WR(00),L(00)
1897	.MACRO	BSPLO(07)+D	.....	BSP(R06),WR(00),L(00)
1898	.MACRO	BSPLO(10)+D	.....	BSP(R10),WR(00),L(00)
1899	.MACRO	BSPLO(11)+D	.....	BSP(R11),WR(00),L(00)
1900	.MACRO	BSPLO(12)+D	.....	BSP(R12),WR(00),L(00)
1901	.MACRO	BSPLO(13)+D	.....	BSP(R13),WR(00),L(00)
1902	.MACRO	BSPLO(14)+D	.....	BSP(R14),WR(00),L(00)
1903	.MACRO	BSPLO(15)+D	.....	BSP(R15),WR(00),L(00)
1904	.MACRO	BSPLO(16)+D	.....	BSP(R16),WR(00),L(00)
1905	.MACRO	BSPLO(17)+D	.....	BSP(R17),WR(00),L(00)
1906				
1907	.MACRO	BSPHI(00)+D	.....	BSP(R00),WR(00),H(00)
1908	.MACRO	BSPHI(01)+D	.....	BSP(R01),WR(00),H(00)
1909	.MACRO	BSPHI(02)+D	.....	BSP(R02),WR(00),H(00)
1910	.MACRO	BSPHI(03)+D	.....	BSP(R03),WR(00),H(00)
1911	.MACRO	BSPHI(04)+D	.....	BSP(R04),WR(00),H(00)
1912	.MACRO	BSPHI(05)+D	.....	BSP(R05),WR(00),H(00)
1913	.MACRO	BSPHI(06)+D	.....	BSP(R06),WR(00),H(00)
1914	.MACRO	BSPHI(07)+D	.....	BSP(R07),WR(00),H(00)
1915	.MACRO	BSPHI(10)+D	.....	BSP(R10),WR(00),H(00)
1916	.MACRO	BSPHI(11)+D	.....	BSP(R11),WR(00),H(00)
1917	.MACRO	BSPHI(12)+D	.....	BSP(R12),WR(00),H(00)
1918	.MACRO	BSPHI(13)+D	.....	BSP(R13),WR(00),H(00)
1919	.MACRO	BSPHI(14)+D	.....	BSP(R14),WR(00),H(00)
1920	.MACRO	BSPHI(15)+D	.....	BSP(R15),WR(00),H(00)
1921	.MACRO	BSPHI(16)+D	.....	BSP(R16),WR(00),H(00)
1922	.MACRO	BSPHI(17)+D	.....	BSP(R17),WR(00),H(00)
1923				
1924	.MACRO	A#BSPLO(00)+D	.....	ASP(R00),BSP(R00),WR(00),L(00)
1925	.MACRO	A#BSPLO(01)+D	.....	ASP(R01),BSP(R01),WR(00),L(00)
1926	.MACRO	A#BSPLO(02)+D	.....	ASP(R02),BSP(R02),WR(00),L(00)
1927	.MACRO	A#BSPLO(03)+D	.....	ASP(R03),BSP(R03),WR(00),L(00)
1928	.MACRO	A#BSPLO(04)+D	.....	ASP(R04),BSP(R04),WR(00),L(00)
1929	.MACRO	A#BSPLO(05)+D	.....	ASP(R05),BSP(R05),WR(00),L(00)
1930	.MACRO	A#BSPLO(06)+D	.....	ASP(R06),BSP(R06),WR(00),L(00)
1931	.MACRO	A#BSPLO(07)+D	.....	ASP(R07),BSP(R07),WR(00),L(00)
1932	.MACRO	A#BSPLO(10)+D	.....	ASP(R10),BSP(R10),WR(00),L(00)
1933	.MACRO	A#BSPLO(11)+D	.....	ASP(R11),BSP(R11),WR(00),L(00)
1934	.MACRO	A#BSPLO(12)+D	.....	ASP(R12),BSP(R12),WR(00),L(00)
1935	.MACRO	A#BSPLO(13)+D	.....	ASP(R13),BSP(R13),WR(00),L(00)
1936	.MACRO	A#BSPLO(14)+D	.....	ASP(R14),BSP(R14),WR(00),L(00)
1937	.MACRO	A#BSPLO(15)+D	.....	ASP(R15),BSP(R15),WR(00),L(00)
1938	.MACRO	A#BSPLO(16)+D	.....	ASP(R16),BSP(R16),WR(00),L(00)
1939	.MACRO	A#BSPLO(17)+D	.....	ASP(R17),BSP(R17),WR(00),L(00)
1940				
1941	.MACRO	A#BSPHI(00)+D	.....	ASP(R00),BSP(R00),WR(00),H(00)
1942	.MACRO	A#BSPHI(01)+D	.....	ASP(R01),BSP(R01),WR(00),H(00)
1943	.MACRO	A#BSPHI(02)+D	.....	ASP(R02),BSP(R02),WR(00),H(00)
1944	.MACRO	A#BSPHI(03)+D	.....	ASP(R03),BSP(R03),WR(00),H(00)

```

1945 .MACRO A#BSPHI[04] +D          ::: =  ASP(R04),BSP(R04),WR(R08,H,A)
1946 .MACRO A#BSPHI[05] +D          ::: =  ASP(R05),BSP(R05),WR(R08,H,A)
1947 .MACRO A#BSPHI[06] +D          ::: =  ASP(R06),BSP(R06),WR(R08,H,A)
1948 .MACRO A#BSPHI[07] +D          ::: =  ASP(R07),BSP(R07),WR(R08,H,A)
1949 .MACRO A#BSPHI[10] +D          ::: =  ASP(R10),BSP(R10),WR(R08,H,A)
1950 .MACRO A#BSPHI[11] +D          ::: =  ASP(R11),BSP(R11),WR(R08,H,A)
1951 .MACRO A#BSPHI[12] +D          ::: =  ASP(R12),BSP(R12),WR(R08,H,A)
1952 .MACRO A#BSPHI[13] +D          ::: =  ASP(R13),BSP(R13),WR(R08,H,A)
1953 .MACRO A#BSPHI[14] +D          ::: =  ASP(R14),BSP(R14),WR(R08,H,A)
1954 .MACRO A#BSPHI[15] +D          ::: =  ASP(R15),BSP(R15),WR(R08,H,A)
1955 .MACRO A#BSPHI[16] +D          ::: =  ASP(R16),BSP(R16),WR(R08,H,A)
1956 .MACRO A#BSPHI[17] +D          ::: =  ASP(R17),BSP(R17),WR(R08,H,A)
1957
1958 .MACRO A#BSPLO[00] +D-[A]       ::: =  ASP(R00),WR(R08,L,A)
1959 .MACRO A#BSPLO[01] +D-[A]       ::: =  ASP(R01),WR(R08,L,A)
1960 .MACRO A#BSPLO[02] +D-[A]       ::: =  ASP(R02),WR(R08,L,A)
1961 .MACRO A#BSPLO[03] +D-[A]       ::: =  ASP(R03),WR(R08,L,A)
1962 .MACRO A#BSPLO[04] +D-[A]       ::: =  ASP(R04),WR(R08,L,A)
1963 .MACRO A#BSPLO[05] +D-[A]       ::: =  ASP(R05),WR(R08,L,A)
1964 .MACRO A#BSPLO[06] +D-[A]       ::: =  ASP(R06),WR(R08,L,A)
1965 .MACRO A#BSPLO[07] +D-[A]       ::: =  ASP(R07),WR(R08,L,A)
1966 .MACRO A#BSPLO[10] +D-[A]       ::: =  ASP(R10),WR(R08,L,A)
1967 .MACRO A#BSPLO[11] +D-[A]       ::: =  ASP(R11),WR(R08,L,A)
1968 .MACRO A#BSPLO[12] +D-[A]       ::: =  ASP(R12),WR(R08,L,A)
1969 .MACRO A#BSPLO[13] +D-[A]       ::: =  ASP(R13),WR(R08,L,A)
1970 .MACRO A#BSPLO[14] +D-[A]       ::: =  ASP(R14),WR(R08,L,A)
1971 .MACRO A#BSPLO[15] +D-[A]       ::: =  ASP(R15),WR(R08,L,A)
1972 .MACRO A#BSPLO[16] +D-[A]       ::: =  ASP(R16),WR(R08,L,A)
1973 .MACRO A#BSPLO[17] +D-[A]       ::: =  ASP(R17),WR(R08,L,A)
1974
1975 .MACRO A#BSPHI[00] +D-[A]       ::: =  ASP(R00),WR(R08,H,A)
1976 .MACRO A#BSPHI[01] +D-[A]       ::: =  ASP(R01),WR(R08,H,A)
1977 .MACRO A#BSPHI[02] +D-[A]       ::: =  ASP(R02),WR(R08,H,A)
1978 .MACRO A#BSPHI[03] +D-[A]       ::: =  ASP(R03),WR(R08,H,A)
1979 .MACRO A#BSPHI[04] +D-[A]       ::: =  ASP(R04),WR(R08,H,A)
1980 .MACRO A#BSPHI[05] +D-[A]       ::: =  ASP(R05),WR(R08,H,A)
1981 .MACRO A#BSPHI[06] +D-[A]       ::: =  ASP(R06),WR(R08,H,A)
1982 .MACRO A#BSPHI[07] +D-[A]       ::: =  ASP(R07),WR(R08,H,A)
1983 .MACRO A#BSPHI[10] +D-[A]       ::: =  ASP(R10),WR(R08,H,A)
1984 .MACRO A#BSPHI[11] +D-[A]       ::: =  ASP(R11),WR(R08,H,A)
1985 .MACRO A#BSPHI[12] +D-[A]       ::: =  ASP(R12),WR(R08,H,A)
1986 .MACRO A#BSPHI[13] +D-[A]       ::: =  ASP(R13),WR(R08,H,A)
1987 .MACRO A#BSPHI[14] +D-[A]       ::: =  ASP(R14),WR(R08,H,A)
1988 .MACRO A#BSPHI[15] +D-[A]       ::: =  ASP(R15),WR(R08,H,A)
1989 .MACRO A#BSPHI[16] +D-[A]       ::: =  ASP(R16),WR(R08,H,A)
1990 .MACRO A#BSPHI[17] +D-[A]       ::: =  ASP(R17),WR(R08,H,A)
1991
1992 .MACRO A#BSPLO[00] +D-[B]       ::: =  BSP(R00),WR(R08,L,B)
1993 .MACRO A#BSPLO[01] +D-[B]       ::: =  BSP(R01),WR(R08,L,B)
1994 .MACRO A#BSPLO[02] +D-[B]       ::: =  BSP(R02),WR(R08,L,B)
1995 .MACRO A#BSPLO[03] +D-[B]       ::: =  BSP(R03),WR(R08,L,B)
1996 .MACRO A#BSPLO[04] +D-[B]       ::: =  BSP(R04),WR(R08,L,B)
1997 .MACRO A#BSPLO[05] +D-[B]       ::: =  BSP(R05),WR(R08,L,B)
1998 .MACRO A#BSPLO[06] +D-[B]       ::: =  BSP(R06),WR(R08,L,B)

```

```

1999 .MACRO A#BSPL0[07]+D-[B]      ::= BSP(R07),WR(AB,L,B)
2000 .MACRO A#BSPL0[10]+D-[B]     ::= BSP(R10),WR(AB,L,B)
2001 .MACRO A#BSPL0[11]+D-[B]     ::= BSP(R11),WR(AB,L,B)
2002 .MACRO A#BSPL0[12]+D-[B]     ::= BSP(R12),WR(AB,L,B)
2003 .MACRO A#BSPL0[13]+D-[B]     ::= BSP(R13),WR(AB,L,B)
2004 .MACRO A#BSPL0[14]+D-[B]     ::= BSP(R14),WR(AB,L,B)
2005 .MACRO A#BSPL0[15]+D-[B]     ::= BSP(R15),WR(AB,L,B)
2006 .MACRO A#BSPL0[16]+D-[B]     ::= BSP(R16),WR(AB,L,B)
2007 .MACRO A#BSPL0[17]+D-[B]     ::= BSP(R17),WR(AB,L,B)
2008
2009 .MACRO A#BSPHI[00]+D-[B]      ::= BSP(R00),WR(AB,H,B)
2010 .MACRO A#BSPHI[01]+D-[B]     ::= BSP(R01),WR(AB,H,B)
2011 .MACRO A#BSPHI[02]+D-[B]     ::= BSP(R02),WR(AB,H,B)
2012 .MACRO A#BSPHI[03]+D-[B]     ::= BSP(R03),WR(AB,H,B)
2013 .MACRO A#BSPHI[04]+D-[B]     ::= BSP(R04),WR(AB,H,B)
2014 .MACRO A#BSPHI[05]+D-[B]     ::= BSP(R05),WR(AB,H,B)
2015 .MACRO A#BSPHI[06]+D-[B]     ::= BSP(R06),WR(AB,H,B)
2016 .MACRO A#BSPHI[07]+D-[B]     ::= BSP(R07),WR(AB,H,B)
2017 .MACRO A#BSPHI[10]+D-[B]     ::= BSP(R10),WR(AB,H,B)
2018 .MACRO A#BSPHI[11]+D-[B]     ::= BSP(R11),WR(AB,H,B)
2019 .MACRO A#BSPHI[12]+D-[B]     ::= BSP(R12),WR(AB,H,B)
2020 .MACRO A#BSPHI[13]+D-[B]     ::= BSP(R13),WR(AB,H,B)
2021 .MACRO A#BSPHI[14]+D-[B]     ::= BSP(R14),WR(AB,H,B)
2022 .MACRO A#BSPHI[15]+D-[B]     ::= BSP(R15),WR(AB,H,B)
2023 .MACRO A#BSPHI[16]+D-[B]     ::= BSP(R16),WR(AB,H,B)
2024 .MACRO A#BSPHI[17]+D-[B]     ::= BSP(R17),WR(AB,H,B)
2025
2026 .MACRO ASPLO[DF]+D           ::= ASP-ADORS-R[DF],WR(A,L,A)
2027 .MACRO ASPHI[DF]+D          ::= ASP-ADORS-R[DF],WR(A,H,A)
2028 .MACRO BSPL0[DF]+D          ::= BSP-ADORS-R[DF],WR(B,L,B)
2029 .MACRO BSPHI[DF]+D          ::= BSP-ADORS-R[DF],WR(B,H,B)
2030
2031 .MACRO ASPLO[SF]+D           ::= ASP-ADORS-R[SF],WR(A,L,A)
2032 .MACRO ASPHI[SF]+D          ::= ASP-ADORS-R[SF],WR(A,H,A)
2033 .MACRO BSPL0[SF]+D          ::= BSP-ADORS-R[SF],WR(B,L,B)
2034 .MACRO BSPHI[SF]+D          ::= BSP-ADORS-R[SF],WR(B,H,B)
2035
2036 .MACRO A#BSPL0[DF]+D-[A]     ::= ASP-ADORS-R[DF],WR(AB,L,A)
2037 .MACRO A#BSPHI[DF]+D-[A]     ::= ASP-ADORS-R[DF],WR(AB,H,A)
2038 .MACRO A#BSPL0[DF]+D-[B]     ::= BSP-ADORS-R[DF],WR(AB,L,B)
2039 .MACRO A#BSPHI[DF]+D-[B]     ::= BSP-ADORS-R[DF],WR(AB,H,B)
2040
2041 .MACRO A#BSPL0[SF]+D-[A]     ::= ASP-ADORS-R[SF],WR(AB,L,A)
2042 .MACRO A#BSPHI[SF]+D-[A]     ::= ASP-ADORS-R[SF],WR(AB,H,A)
2043 .MACRO A#BSPL0[SF]+D-[B]     ::= BSP-ADORS-R[SF],WR(AB,L,B)
2044 .MACRO A#BSPHI[SF]+D-[B]     ::= BSP-ADORS-R[SF],WR(AB,H,B)
2045
2046 .MACRO A#BSPL0[SF]+D         ::= ASP-ADORS-R[SF],BSP-ADORS-R[SF],WR(AB,L,A)
2047 .MACRO A#BSPL0[DF]+D         ::= ASP-ADORS-R[DF],BSP-ADORS-R[DF],WR(AB,L,A)
2048 .MACRO A#BSPHI[SF]+D         ::= ASP-ADORS-R[SF],BSP-ADORS-R[SF],WR(AB,H,A)
2049 .MACRO A#BSPHI[DF]+D         ::= ASP-ADORS-R[DF],BSP-ADORS-R[DF],WR(AB,H,A)
2050
2051
2052

```

```

2053 !.PAGE=====
2054
2055 .TOC *      D AND SR (- (BUS-A FCN BUS-B), AT P2-T OR P3-T
2056
2057 !LOGIC FUNCTIONS:
2058 .MACRO SR+ZERO          ::= ZERO,CLK-SR
2059 .MACRO SR+A-XOR-B      ::= A-XOR-B,CLK-SR
2060 .MACRO SR+B            ::= B,CLK-SR
2061 .MACRO SR+A-AND-B     ::= A-AND-B,CLK-SR
2062 .MACRO SR+A-IOR-B     ::= A-IOR-B,CLK-SR
2063 .MACRO SR+A           ::= A,CLK-SR
2064 .MACRO SR+NOT-A       ::= NOT-A,CLK-SR
2065 .MACRO SR+NOT-A-AND-B ::= NOT-A-AND-B,CLK-SR
2066 .MACRO SR+A-AND-NOT-B ::= A-AND-NOT-B,CLK-SR
2067 .MACRO D+ZERO         ::= ZERO,CLK-D
2068 .MACRO D+A-XOR-B     ::= A-XOR-B,CLK-D
2069 .MACRO D+B           ::= B,CLK-D
2070 .MACRO D+A-AND-B     ::= A-AND-B,CLK-D
2071 .MACRO D+A-IOR-B     ::= A-IOR-B,CLK-D
2072 .MACRO D+A           ::= A,CLK-D
2073 .MACRO D+NOT-A       ::= NOT-A,CLK-D
2074 .MACRO D+NOT-A-AND-B ::= NOT-A-AND-B,CLK-D
2075 .MACRO D+A-AND-NOT-B ::= A-AND-NOT-B,CLK-D
2076
2077 !ARITH FUNCTIONS:
2078 .MACRO D+DIVIDE-STEP  ::= DIVIDE,CLK-D
2079 .MACRO D+A-PLUS-B     ::= A-PLUS-B,CLK-D
2080 .MACRO D+A-PLUS-B-PLUS-0 ::= A-PLUS-B,CLK-D
2081 .MACRO D+A-MINUS-B    ::= A-MINUS-B,CLK-D
2082 .MACRO D+A-PLUS-B-PLUS-PS(C) ::= A-PLUS-B-PLUS-PS(C),CLK-D
2083 .MACRO D+A-PLUS-B-PLUS-D(C) ::= A-PLUS-B-PLUS-D(C),CLK-D
2084 .MACRO D+A-PLUS-NOT-B-PLUS-D(C) ::= A-PLUS-NOT-B-PLUS-D(C),CLK-D
2085 .MACRO D+A-PLUS-B-PLUS-1 ::= A-PLUS-B-PLUS-1,CLK-D
2086 .MACRO SR+DIVIDE-STEP ::= DIVIDE,CLK-SR
2087 .MACRO SR+A-PLUS-B    ::= A-PLUS-B,CLK-SR
2088 .MACRO SR+A-PLUS-B-PLUS-0 ::= A-PLUS-B,CLK-SR
2089 .MACRO SR+A-MINUS-B   ::= A-MINUS-B,CLK-SR
2090 .MACRO SR+A-PLUS-B-PLUS-PS(C) ::= A-PLUS-B-PLUS-PS(C),CLK-SR
2091 .MACRO SR+A-PLUS-B-PLUS-D(C) ::= A-PLUS-B-PLUS-D(C),CLK-SR
2092 .MACRO SR+A-PLUS-NOT-B-PLUS-D(C) ::= A-PLUS-NOT-B-PLUS-D(C),CLK-SR
2093 .MACRO SR+A-PLUS-B-PLUS-1 ::= A-PLUS-B-PLUS-1,CLK-SR
2094
2095
2096
2097
2098 .TOC *      D(C) GETS SET
2099
2100 .MACRO D(C)+CINMUX    ::= CLK-D,COUT+CIN
2101 .MACRO D(C)+1        ::= CLK-D,COUT+CIN      !NEEDS SPECIFIC ALU/---
2102 .MACRO D(C)+0        ::= CLK-D,COUT+CIN      !NEEDS SPECIFIC ALU/---
2103 .MACRO D(C)+PS(C)    ::= CLK-D,COUT+PS(C)
2104 .MACRO D(C)+ALU00    ::= CLK-D,COUT+ALU00
2105 .MACRO D(C)+ALU07    ::= CLK-D,COUT+ALU07
2106 .MACRO D(C)+ALU15    ::= CLK-D,COUT+ALU15

```



```

2107 .MACRO D[C]+COUT07      ::= CLK-D,COUT+COUT07
2108 .MACRO D[C]+COUT15      ::= CLK-D,COUT+COUT15
2109 .MACRO D[C]+D[C]         ::= CLK-B;COUT+B[C]
2110 .MACRO SAVE-D[C]         ::= CLK-B;COUT+B[C]
2111
2112
2113
2114
2115
2116 .TOC *      D-REGISTER (- [BBUS = ABUS], BITWISE, AT P2-T OR P3-T
2117
2118     IN B:  SHIFT TREE ENABLED SEPARATELY
2119 .MACRO D+D-SHIFTED-XOR-CSPB(XX)  ::= A-XOR-B,CSPB(@XX),CLK-D
2120 .MACRO D+D-SHIFTED-XOR-BSPHI(XX) ::= A-XOR-B,BSPHI(@XX),CLK-D
2121
2122 .MACRO D+FLTPT-XOR-CSPB(XX)      ::= A-XOR-B,FLTPT,CSPB(@XX),CLK-D
2123 .MACRO D+FLTPT-XOR-CSPD(XX)      ::= A-XOR-B,FLTPT,CSPD(@XX),CLK-D
2124 .MACRO D+FLTPT-XOR-BSPHI(XX)     ::= A-XOR-B,FLTPT,BSPHI(@XX),CLK-D
2125
2126 .MACRO D+SR-XOR-CSPB(XX)          ::= A-XOR-B,SR,CSPB(@XX),CLK-D
2127 .MACRO D+SR-XOR-CSPD(XX)          ::= A-XOR-B,SR,CSPD(@XX),CLK-D
2128 .MACRO D+SR-XOR-BSPHI(XX)         ::= A-XOR-B,SR,BSPHI(@XX),CLK-D
2129
2130 .MACRO D+ASPLO[17]-XOR-CSPD(XX)    ::= A-XOR-B,ASPLO(R17),CSPD(@XX),CLK-D
2131 .MACRO D+ASPLO[07]-XOR-BSPHI(XX)  ::= A-XOR-B,ASPLO(R07),BSPHI(@XX),CLK-D
2132 .MACRO D+ASPLO[05]-XOR-BSPHI(XX)  ::= A-XOR-B,ASPLO(R05),BSPHI(@XX),CLK-D
2133
2134 .MACRO D+SR-XOR-ASPLO[SF]          ::= A-XOR-B,SR,R[SF]-LO-B,CLK-D
2135 .MACRO D+SR-XOR-BSPHI[DF]         ::= A-XOR-B,SR,R[DF]-HI-B,CLK-D
2136
2137 .MACRO D+ASPLO[DF]-XOR-BSPHI[SF]  ::= A-XOR-B,R[DF]-LO-A,R[SF]-HI-B,CLK-D
2138 .MACRO D+ASPLO[SF]-XOR-BSPLO[DF]  ::= A-XOR-B,R[SF]-HI-A,R[DF]-LO-B,CLK-D
2139
2140 .MACRO D+CSPD[05]-XOR-ASPLO(XX)    ::= A-XOR-B,CSPD(D05),ASPLO(@XX),CLK-D
2141 .MACRO D+CSPD[05]-XOR-ASPHI(XX)    ::= A-XOR-B,CSPD(D05),ASPHI(@XX),CLK-D
2142 .MACRO D+CSPD[06]-XOR-ASPLO(XX)    ::= A-XOR-B,CSPD(D06),ASPLO(@XX),CLK-D
2143 .MACRO D+CSPD[06]-XOR-ASPHI(XX)    ::= A-XOR-B,CSPD(D06),ASPHI(@XX),CLK-D
2144 .MACRO D+CSPD[17]-XOR-ASPHI(XX)    ::= A-XOR-B,CSPD(D17),ASPHI(@XX),CLK-D
2145
2146 .MACRO D+ASPLO[02]-XOR-BSPLO(XX)   ::= A-XOR-B,ASPLO(R02),BSPLO(@XX),CLK-D
2147 .MACRO D+ASPLO[03]-XOR-BSPLO(XX)   ::= A-XOR-B,ASPLO(R03),BSPLO(@XX),CLK-D
2148 .MACRO D+ASPLO[04]-XOR-BSPLO(XX)   ::= A-XOR-B,ASPLO(R04),BSPLO(@XX),CLK-D
2149 .MACRO D+ASPLO[05]-XOR-BSPLO(XX)   ::= A-XOR-B,ASPLO(R05),BSPLO(@XX),CLK-D
2150
2151
2152
2153 .TOC *      D-REGISTER (- D-REGISTER THRU SHIFT-TREE
2154
2155 .MACRO D+D-RIGHT-14      ::= A,D-RIGHT-14,CLK-D
2156 .MACRO D+D-RIGHT-13      ::= A,D-RIGHT-13,CLK-D
2157 .MACRO D+D-RIGHT-12      ::= A,D-RIGHT-12,CLK-D
2158 .MACRO D+D-RIGHT-11      ::= A,D-RIGHT-11,CLK-D
2159 .MACRO D+D-RIGHT-10      ::= A,D-RIGHT-10,CLK-D
2160 .MACRO D+D-RIGHT-9        ::= A,D-RIGHT-9,CLK-D

```

```

2161 .MACRO D+D-RIGHT-8      ::= A,D-D-RIGHT-8,CLK-D
2162 .MACRO D+D-RIGHT-7      ::= A,D-D-RIGHT-7,CLK-D
2163 .MACRO D+D-RIGHT-6      ::= A,D-D-RIGHT-6,CLK-D
2164 .MACRO D+D-RIGHT-5      ::= A,D-D-RIGHT-5,CLK-D
2165 .MACRO D+D-RIGHT-4      ::= A,D-D-RIGHT-4,CLK-D
2166 .MACRO D+D-RIGHT-3      ::= A,D-D-RIGHT-3,CLK-D
2167 .MACRO D+D-RIGHT-2      ::= A,D-D-RIGHT-2,CLK-D
2168 .MACRO D+D-RIGHT-1      ::= A,D-D-RIGHT-1,CLK-D
2169 .MACRO D+D-NO-SHIFT      ::= A,D-D-NO-SHIFT,CLK-D
2170 .MACRO D+D-DIRECT      ::= A,D-D-DIRECT,CLK-D
2171 .MACRO D+D              ::= A,D-D-DIRECT,CLK-D
2172 .MACRO SAVE-D          ::= A,D-D-DIRECT,CLK-D
2173 .MACRO D+D-LEFT-1       ::= A,D-D-LEFT-1,CLK-D
2174 .MACRO D+D-SWAB         ::= A,D-D-SWAB,CLK-D
2175 .MACRO D+D-SWAB-RIGHT-3 ::= A,D-D-SWAB-RIGHT-3,CLK-D
2176 .MACRO D+D-SWAB-LEFT-1  ::= A,D-D-SWAB-LEFT-1,CLK-D
2177 .MACRO D+D-SIGNEXT      ::= A,D-D-SIGNEXT,CLK-D
2178 .MACRO D+D-SIGNEXT-RIGHT-1 ::= A,D-D-SIGNEXT-RIGHT-1,CLK-D
2179 .MACRO D+D-SIGNEXT-LEFT-1 ::= A,D-D-SIGNEXT-LEFT-1,CLK-D
2180 .MACRO D+NO-SHIFT      ::= A,NO-SHIFT,CLK-D
2181 .MACRO D+DIRECT        ::= A,DIRECT,CLK-D
2182 .MACRO D+COUNT#D[HI]  ::= A,COUNT#D[HI],CLK-D
2183 .MACRO D+COUNT#D[LO]  ::= A,COUNT#D[LO],CLK-D
2184
2185
2186
2187 .TOC * D (- WHATEVER'S LEFT, AT P2-T OR P3-T)
2188
2189 .MACRO D+NOT-ASPHI(XX)   ::= NOT-A,ASPHI(@XX),CLK-D
2190 .MACRO D+NOT-ASPLO(XX)  ::= NOT-A,ASPLO(@XX),CLK-D
2191 .MACRO D+NOT-CSPB(XX)   ::= A-AND-NOT-B,C177777-A,CSPB(@XX),CLK-D
2192 .MACRO D+NOT-CSPD(XX)   ::= A-AND-NOT-B,C177777-A,CSPD(@XX),CLK-D
2193
2194 .MACRO D+CSPD(XX)        ::= B,CSPD(@XX),CLK-D
2195 .MACRO D+CSPB(XX)        ::= B,CSPB(@XX),CLK-D
2196 .MACRO D+CSPB[16]#D[C]+1 ::= A-IOR-B,C000000-A,CSPB(B16),CLK-D,D[C]+CINMUX
2197
2198 .MACRO D+BSPHI(XX)       ::= B,BSPHI(@XX),CLK-D
2199 .MACRO D+BSPLO(XX)       ::= B,BSPLO(@XX),CLK-D
2200 .MACRO D+ASPHI(XX)       ::= A,ASPHI(@XX),CLK-D
2201 .MACRO D+ASPLO(XX)       ::= A,ASPLO(@XX),CLK-D
2202
2203 .MACRO D+ASPLO[DF]       ::= A,R[DF]-LO-A,CLK-D
2204 .MACRO D+ASPHI[DF]       ::= A,R[DF]-HI-A,CLK-D
2205 .MACRO D+BSPLO[DF]       ::= B,R[DF]-LO-B,CLK-D
2206 .MACRO D+BSPHI[DF]       ::= B,R[DF]-HI-B,CLK-D
2207 .MACRO D+ASPLO[SF]       ::= A,R[SF]-LO-A,CLK-D
2208 .MACRO D+ASPHI[SF]       ::= A,R[SF]-HI-A,CLK-D
2209 .MACRO D+BSPLO[SF]       ::= B,R[SF]-LO-B,CLK-D
2210 .MACRO D+BSPHI[SF]       ::= B,R[SF]-HI-B,CLK-D
2211
2212 .MACRO D+CSPD[14]-AND-ASPHI(XX) ::= A-AND-B,CSPD(D14),ASPHI(@XX),CLK-D
2213 .MACRO D+CSPD[15]-AND-ASPHI(XX) ::= A-AND-B,CSPD(D15),ASPHI(@XX),CLK-D
2214

```

```

2215 .MACRO SR+ASPHI(17)-AND-007700 ::= A-AND-B, ASPHI(R17), CSPB(B17), CLK-SR
2216 .MACRO D+SR-IOR-170000 ::= A-IOR-B, SR, CSPB(B16), CLK-D
2217 .MACRO SR+ASPHI(17)-AND-000077 ::= A-AND-B, ASPHI(R17), CSPB(B15), CLK-SR
2218 .MACRO D+SR-IOR-000100 ::= A-IOR-B, SR, CSPB(B14), CLK-D
2219
2220 .MACRO D+ASPLO(17)-AND-CSPD(XX) ::= A-AND-B, ASPLO(R17), CSPD(XX), CLK-D
2221 .MACRO D+ASPHI(00)-IOR-CSPD(XX) ::= A-IOR-B, ASPHI(R00), CSPD(XX), CLK-D
2222 .MACRO D+ASPHI(00)-IOR-CSPB(XX) ::= A-IOR-B, ASPHI(R00), CSPB(XX), CLK-D
2223
2224 .MACRO D+SR ::= A, SR, CLK-D
2225 .MACRO D+ALL-ONES ::= A, C177777-A, CLK-D
2226 .MACRO D+D-PLUS-1 ::= A-PLUS-B, D-DIRECT, C000001-B, CLK-D
2227 .MACRO D+JUNK ::= ZERO, CLK-D
2228 .MACRO D+TWO ::= A-PLUS-B, C000001-A, C000001-B, CLK-D
2229
2230
2231

```

2232 !.PAGE=====

2233 .TOC \* SR (- DATA, AT P2 T OR P3 T

```

2236 !4.B.: THE PARTICULAR FUNCTION SELECTED REQUIRES THE RESIDUAL
2237 !: CONTROL REGISTER ("RES-REG") TO HAVE THE APPROPRIATE
2238 !: FUNCTION SETUP FOR THE SR OPERATION.
2239 !:

```

2240 !: POSSIBLE FUNCTIONS: LOAD, LEFT, RIGHT, NOP

```

2241
2242 .MACRO SR+ASPHI(XX) ::= A, ASPHI(XX), CLK-SR
2243 .MACRO SR+NOT-ASPHI(XX) ::= NOT-A, ASPHI(XX), CLK-SR
2244 .MACRO SR+CSPB(XX) ::= B, CSPB(XX), CLK-SR
2245 .MACRO SR+CSPD(XX) ::= B, CSPD(XX), CLK-SR
2246 .MACRO SR+NOT-BSPHI(XX) ::= A-AND-NOT-B, BSPHI(XX), C177777-A, CLK-SR
2247 .MACRO SR+BSPHI(XX) ::= B, BSPHI(XX), CLK-SR
2248 .MACRO SR+SR-PLUS-1 ::= A-PLUS-B, C000001-B, SR, CLK-SR
2249 .MACRO SR+ALL-ONES ::= A, C177777-A, CLK-SR
2250 .MACRO SR+NOT-CSPB(XX) ::= A-AND-NOT-B, C177777-A, CSPB(XX), CLK-SR
2251 .MACRO SR+NOT-CSPD(XX) ::= A-AND-NOT-B, C177777-A, CSPD(XX), CLK-SR
2252 .MACRO SR+SR-RIGHT-1 ::= D-DIRECT(BMUX), CLK-SR
2253 .MACRO SR+SR-LEFT-1 ::= CLK-SR
2254 .MACRO SR+JUNK ::= ZERO, CLK-SR
2255 .MACRO SR+D ::= A, D-DIRECT, CLK-SR
2256 .MACRO SR+ASPLO(DF) ::= A, R(DF)-LO-A, CLK-SR
2257 .MACRO SR+ASPHI(DF) ::= A, R(DF)-HI-A, CLK-SR
2258 .MACRO SR+BSPLO(DF) ::= B, R(DF)-LO-B, CLK-SR
2259 .MACRO SR+BSPHI(DF) ::= B, R(DF)-HI-B, CLK-SR
2260 .MACRO SR+ASPLO(SF) ::= A, R(SF)-LO-A, CLK-SR
2261 .MACRO SR+ASPHI(SF) ::= A, R(SF)-HI-A, CLK-SR
2262 .MACRO SR+BSPLO(SF) ::= B, R(SF)-LO-B, CLK-SR
2263 .MACRO SR+BSPHI(SF) ::= B, R(SF)-HI-B, CLK-SR
2264
2265
2266
2267
2268

```

```

2269 .TOC * RES-REG OPERATION MACROS
2270
2271 .MACRO RES+CSPD(XX) ::= CSPD(2XX),LOAD-RES
2272 .MACRO RES+CSPB(XX) ::= CSPB(2XX),LOAD-RES
2273
2274
2275
2276
2277
2278 .TOC * BASE MACHINE COUNTER
2279
2280 .MACRO COUNTER+CSPD(XX) ::= LOAD-COUNTER,CSPD(2XX)
2281 .MACRO COUNTER+BSPHI(XX) ::= LOAD-COUNTER,BSPHI(2XX)
2282
2283
2284
2285
2286 .TOC * ENABLE ON BUS-A/B ONLY
2287
2288 .MACRO BUS-A+ASPLO(SF) ::= R(SF)-LO-A
2289 .MACRO BUS-A+ASPLO(DF) ::= R(DF)-LO-A
2290 .MACRO BUS-A+ASPHI(SF) ::= R(SF)-HI-A
2291 .MACRO BUS-A+ASPHI(DF) ::= R(DF)-HI-A
2292 .MACRO BUS-A ::= NULL
2293 .MACRO BUS-A+ASPLO(XX) ::= ASPLO(2XX)
2294 .MACRO BUS-A+ASPHI(XX) ::= ASPHI(2XX)
2295 .MACRO BUS-A+SR ::= SR
2296 .MACRO BUS-A+FLTPT ::= FLTPT
2297
2298 .MACRO BUS-B+BSPLO(SF) ::= R(SF)-LO-B
2299 .MACRO BUS-B+BSPLO(DF) ::= R(DF)-LO-B
2300 .MACRO BUS-B+BSPHI(SF) ::= R(SF)-HI-B
2301 .MACRO BUS-B+BSPHI(DF) ::= R(DF)-HI-B
2302 .MACRO BUS-B ::= NULL
2303 .MACRO BUS-B+BSPLO(XX) ::= BSPLO(2XX)
2304 .MACRO BUS-B+BSPHI(XX) ::= BSPHI(2XX)
2305 .MACRO BUS-B+CSPD(XX) ::= CSPD(2XX)
2306 .MACRO BUS-B+CSPB(XX) ::= CSPB(2XX)
2307
2308
2309
2310
2311
2312 .TOC * LOADING BA REGISTER
2313 !LOADED AT P1-T ONLY, FROM BUS-B<01:00>#BUS-A<15:00> -> BA<17:00>
2314
2315 .MACRO BA+BSPLO(XX) ::= CLK-BA,BSPLO(2XX)
2316 .MACRO BA+BSPHI(XX) ::= CLK-BA,BSPHI(2XX)
2317 .MACRO BA+SR ::= CLK-BA,SR
2318 .MACRO BA+ASPLO(XX) ::= CLK-BA,ASPLO(2XX)
2319 .MACRO BA+ASPHI(XX) ::= CLK-BA,ASPHI(2XX)
2320
2321
2322

```

```

2323
2324
2325 .TOC *      D AND SR TOGETHER
2326
2327 .MACRO, SR#D+SR-PLUS-CSPD(XX) ::= A-PLUS-B, SR, CSPD(2XX), CLK-D, CLK-SR
2328
2329
2330
2331 !.PAGE=====
2332
2333 .TOC *      UCON FUNCTIONS
2334
2335
2336
2337 .TOC *      PROCESSOR UCON FUNCTIONS
2338
2339 !PREVIOUSLY SET UP [UCON-PROC, SET-UCON-CONTROL, EN "FUNCTION"]
2340 .MACRO IR+EMIT ::= UCON-OPERATION
2341 .MACRO PS[15-12]+D[15#13] ::= UCON-OPERATION
2342 .MACRO FLAG[8-0]+D[15-8] ::= UCON-OPERATION
2343 .MACRO FPS[7-4]+D[7-4] ::= UCON-OPERATION
2344 .MACRO PS[7-4]+D[7-4] ::= UCON-OPERATION
2345 .MACRO PS[3-0]+D[3-0] ::= UCON-OPERATION
2346 .MACRO PS+D ::= UCON-OPERATION
2347 .MACRO LBREAK+BUSDIN[11-00] ::= UCON-OPERATION
2348
2349 !SETUP UCON AND EXECUTE IN 1 MICROWORD:
2350 .MACRO PS[15-12]+D[15#13]-[I] ::= UCON-PROC, SET-UCON-CONTROL, UCON-OPERATION, EN-CLK-PS[15-12]
2351 .MACRO FLAG[8-0]+D[15-8]-[I] ::= UCON-PROC, SET-UCON-CONTROL, UCON-OPERATION, EN-CLK-FLAG[8-0]
2352 .MACRO FPS[7-4]+D[7-4]-[I] ::= UCON-PROC, SET-UCON-CONTROL, UCON-OPERATION, EN-CLK-FPS[7-4]
2353 .MACRO PS[7-4]+D[7-4]-[I] ::= UCON-PROC, SET-UCON-CONTROL, UCON-OPERATION, EN-CLK-PS[7-4]
2354 .MACRO PS[3-0]+D[3-0]-[I] ::= UCON-PROC, SET-UCON-CONTROL, UCON-OPERATION, EN-CLK-PS[3-0]
2355 .MACRO PS+D-[I] ::= UCON-PROC, SET-UCON-CONTROL, UCON-OPERATION,
2356 EN-CLK-PS[15-12], EN-CLK-PS[7-4], EN-CLK-PS[3-0]
2357 .MACRO BUSDIN+CUA-[I] ::= UCON-PROC, SET-UCON-CONTROL, BUSDIN+CUA[14-03]
2358 .MACRO BUSDIN+FLAGS#FPS-[I] ::= UCON-PROC, SET-UCON-CONTROL, BUSDIN+FLAG[8-0]#FPS[7-0]
2359 .MACRO BUSDIN+PS-[I] ::= UCON-PROC, SET-UCON-CONTROL, BUSDIN+PS[15-00]
2360 .MACRO BUSDIN+EMIT-[I] ::= UCON-PROC, SET-UCON-CONTROL, BUSDIN+EMIT[15-00]
2361
2362
2363
2364 .TOC *      CACHE/KT UCON FUNCTIONS
2365
2366 !SETUP, EXECUTE IN 1 MICROWORD
2367 .MACRO KT-NO-RELOCATE-[I] ::= UCON-CACHE-KT, SET-UCON-CONTROL, EN-KT-NO-RELOCATE
2368 .MACRO BUSDIN+BUS-INTERNAL-ADDR-[I] ::= UCON-CACHE-KT, SET-UCON-CONTROL, BUSDIN+BUS-INTERNAL-ADDR[15-00]
2369 .MACRO BUSDIN+CPU-INTERNAL-ADDR-[I] ::= UCON-CACHE-KT, SET-UCON-CONTROL, BUSDIN+CPU-INTERNAL-ADDR[15-00]
2370 .MACRO BUSDIN+MMR2-[I] ::= UCON-CACHE-KT, SET-UCON-CONTROL, BUSDIN+MMR2[15-00]
2371 .MACRO BUSDIN+CACHE-STATUS-[I] ::= UCON-CACHE-KT, SET-UCON-CONTROL, BUSDIN+CACHE-STATUS[15-00]
2372 .MACRO BUSDIN+SLR#CCR-[I] ::= UCON-CACHE-KT, SET-UCON-CONTROL, BUSDIN+KT-SEL, KT-SEL-SLR#CCR
2373 .MACRO BUSDIN+MMR0-[I] ::= UCON-CACHE-KT, SET-UCON-CONTROL, BUSDIN+KT-SEL, KT-SEL-MMR0
2374 .MACRO BUSDIN+PDR-[I] ::= UCON-CACHE-KT, SET-UCON-CONTROL, BUSDIN+KT-SEL, KT-SEL-PDR
2375 .MACRO BUSDIN+PAR-[I] ::= UCON-CACHE-KT, SET-UCON-CONTROL, BUSDIN+KT-SEL, KT-SEL-PAR
2376 .MACRO SLR[15-08]+D[15-08]-[I] ::= UCON-CACHE-KT, SET-UCON-CONTROL, UCON-OPERATION, KT-SEL-SLR#CCR, KT-WRITE-HIGH

```

```

2377 .MACRO CCR(07-02)+D(07-02)-[I] ::= UCON-CACHE-KT,SET-UCON-CONTROL,UCON-OPERATION,KT-SEL-SLR#CCR,KT-WRITE-LOW
2378 .MACRO NPRO+D-[I] ::= UCON-CACHE-KT,SET-UCON-CONTROL,UCON-OPERATION,KT-SEL-NPRO,KT-WRITE
2379 .MACRO NPRO(00)+D(00)-[I] ::= UCON-CACHE-KT,SET-UCON-CONTROL,UCON-OPERATION,KT-SEL-NPRO,KT-WRITE-LOW
2380 .MACRO NPRO(15-01)+D(15-01)-[I] ::= UCON-CACHE-KT,SET-UCON-CONTROL,UCON-OPERATION,KT-SEL-NPRO,KT-WRITE-HIGH
2381 .MACRO POR+D-[I] ::= UCON-CACHE-KT,SET-UCON-CONTROL,UCON-OPERATION,KT-SEL-POR,KT-WRITE
2382 .MACRO POR(03-01)+D(03-01)-[I] ::= UCON-CACHE-KT,SET-UCON-CONTROL,UCON-OPERATION,KT-SEL-POR,KT-WRITE-LOW
2383 .MACRO POR(14-08)+D(14-08)-[I] ::= UCON-CACHE-KT,SET-UCON-CONTROL,UCON-OPERATION,KT-SEL-POR,KT-WRITE-HIGH
2384 .MACRO PAR+D-[I] ::= UCON-CACHE-KT,SET-UCON-CONTROL,UCON-OPERATION,KT-SEL-PAR,KT-WRITE
2385 .MACRO PAR(07-00)+D(07-00)-[I] ::= UCON-CACHE-KT,SET-UCON-CONTROL,UCON-OPERATION,KT-SEL-PAR,KT-WRITE-LOW
2386 .MACRO PAR(11-08)+D(11-08)-[I] ::= UCON-CACHE-KT,SET-UCON-CONTROL,UCON-OPERATION,KT-SEL-PAR,KT-WRITE-HIGH

```

2390 .TOC \* I-O UCON FUNCTIONS

2391 !N.B.: SETUP IN 1 MICROWORD

```

2392 .MACRO BUSDIN+JAM-[I] ::= UCON-I-0,EN-STATUS-MUX,SET-UCON-CONTROL,BUSDIN+JAM(15-00)
2393 .MACRO BUSDIN+SERVICE-[I] ::= UCON-I-0,EN-STATUS-MUX,SET-UCON-CONTROL,BUSDIN+SERVICE(15-00)
2394 .MACRO BUSDIN+PBA-[I] ::= UCON-I-0,EN-STATUS-MUX,SET-UCON-CONTROL,BUSDIN+PBA(15-00)
2395 .MACRO BC-FCN-0-[I] ::= UCON-I-0,SET-UCON-CONTROL,UCON-OPERATION,EN-BC-FCN-0
2396 .MACRO START-DELAY-[I] ::= UCON-I-0,SET-UCON-CONTROL,UCON-OPERATION,EN-START-DELAY
2397 .MACRO CLR-JAM-ERRORS-[I] ::= UCON-I-0,SET-UCON-CONTROL,UCON-OPERATION,EN-CLR-JAM-ERRORS
2398 .MACRO CLR-NPR-TIMEOUT-[I] ::= UCON-I-0,SET-UCON-CONTROL,UCON-OPERATION,EN-CLR-NPR-TIMEOUT
2399 .MACRO CLR-PWR-FAIL-[I] ::= UCON-I-0,SET-UCON-CONTROL,UCON-OPERATION,EN-CLR-PWR-FAIL
2400 .MACRO CLR-YELLOW-ZONE-[I] ::= UCON-I-0,SET-UCON-CONTROL,UCON-OPERATION,EN-CLR-YELLOW-ZONE
2401 .MACRO ALLOW-BG(1)H-[I] ::= UCON-I-0,SET-UCON-CONTROL,UCON-OPERATION,EN-ALLOW-BG(1)H
2402 .MACRO BUS-INIT-UCON-[I] ::= UCON-I-0,SET-UCON-CONTROL,UCON-OPERATION,EN-BUS-INIT-UCON

```

2406 .TOC \* DCS UCON FUNCTIONS

2407 !SETUP IN 1 MICROWORD:

```

2408 .MACRO BUSDIN+TMUA-[I] ::= UCON-DCS,SET-UCON-CONTROL,BUSDIN+TMUA(11-00)
2409 .MACRO BUSDIN+ERROR-CODE-[I] ::= UCON-DCS,SET-UCON-CONTROL,BUSDIN+ERR#EOP#ERRCOD(11-00)

```

2413 .TOC \* CONSOLE UCON FUNCTIONS

2414 !SETS UP AND PERFORMS INDICATED OPERATION IN 1 MICROWORD

```

2415 .MACRO CONSOLE-NOP ::= UCON-I-0,EN-CONSOLE-COMMAND,SET-UCON-CONTROL,UCON-OPERATION,EN-CNSL-NOP
2416 .MACRO CLEAR-CONSOLE-COUNTER ::= UCON-I-0,EN-CC SOLE-COMMAND,SET-UCON-CONTROL,UCON-OPERATION,EN-CLR-COUNTR
2417 .MACRO INCREMENT-CONSOLE-COUNTER ::= UCON-I-0,EN-CONSOLE-COMMAND,SET-UCON-CONTROL,UCON-OPERATION,EN-INCR-COUNTR
2418 .MACRO CLEAR-CONSOLE-SERVICE ::= UCON-I-0,EN-CONSOLE-COMMAND,SET-UCON-CONTROL,UCON-OPERATION,EN-CLR-CNSL-SRVC
2419 .MACRO STROBE-CONSOLE-DISPLAY ::= UCON-I-0,EN-CONSOLE-COMMAND,SET-UCON-CONTROL,UCON-OPERATION,EN-STRB-DISP
2420 .MACRO CLEAR-CONSOLE-LED ::= UCON-I-0,EN-CONSOLE-COMMAND,SET-UCON-CONTROL,UCON-OPERATION,EN-CLR-CNSL
2421 .MACRO SET-CONSOLE-LED ::= UCON-I-0,EN-CONSOLE-COMMAND,SET-UCON-CONTROL,UCON-OPERATION,EN-SET-CNSL
2422 .MACRO SET-CONSOLE-OP-LEDS ::= UCON-I-0,EN-CONSOLE-COMMAND,SET-UCON-CONTROL,UCON-OPERATION,EN-SET-OP
2423 .MACRO BUSDIN+CONSOLE-[I] ::= UCON-I-0,EN-STATUS-MUX,SET-UCON-CONTROL,BUSDIN+CONSOLE(06-00)

```

2428 .TOC \* DBUF UCON FUNCTIONS

2429 !PREVIOUSLY SETUP UCON-I-0, EN LOAD DBUF

2430

2431  
2432  
2433  
2434  
2435  
2436  
2437  
2438  
2439  
2440  
2441  
2442  
2443  
2444  
2445  
2446  
2447  
2448  
2449  
2450  
2451  
2452  
2453  
2454  
2455  
2456  
2457  
2458  
2459  
2460  
2461  
2462  
2463  
2464  
2465  
2466  
2467  
2468  
2469  
2470  
2471  
2472  
2473  
2474  
2475  
2476  
2477  
2478  
2479  
2480  
2481  
2482  
2483  
2484

```

.MACRO DBUF+D                ::= UCON-OPERATION
!SETUP AND EXECUTE IN 1 MICROWORD:
.MACRO DBUF+D-[I]           ::= UCON-I-0,SET-UCON-CONTROL,UCON-OPERATION,EN-LOAD-DBUF[15-00]

.TOC *      MULTIPLE UCON FUNCTIONS
!THESE ARE FUNCTIONS OF MORE THAN 1 UCON ENABLED SIMULTANEOUSLY
!PREVIOUSLY SETUP:
.MACRO IR+DBUF              ::= UCON-OPERATION
!SETUP AND EXECUTE IN 1 MICROWORD:
.MACRO IR+DBUF-[I]         ::= UCON-PROC,UCON-I-0,SET-UCON-CONTROL,
                           UCON-OPERATION,EN-CLK-IR[15-00],BUSDIN+DBUF[15-00]

```

!.PAGE=====

```

.TOC *      SPECIFIC MACROS FOR PREFETCH/OVERLAP/SP-INHIBIT TESTS

.MACRO CSPD[17]+020010      ::= EMIT/020010,CSPD[17]+EMIT
.MACRO A+ASPLO[OVERLAP]+D   ::= ASPLO(R10),WR(A,B,L,A)
.MACRO A+ASPHI[PATTERN]+D  ::= ASPHI(R17),WR(A,B,H,A)
.MACRO A+ASPHI[PREFETCH]+D ::= ASPHI(R10),WR(A,B,H,A)
.MACRO ASPLO[OVERLAP]+D    ::= ASPLO(R10),WR(A,L,A)
.MACRO BSPLO[OVERLAP]+D    ::= BSPLO(R10),WR(B,L,B)
.MACRO ASPHI[PREFETCH]+D   ::= ASPHI(R10),WR(A,H,A)

.MACRO D+ASPLO[OVERLAP]-PLUS-1 ::= A-PLUS-B,ASPLO(R10),C000001-B,CLK-D
.MACRO D+BSPLO[OVERLAP]-PLUS-1 ::= A-PLUS-B,BSPLO(R10),C000001-A,CLK-D
.MACRO D+ASPHI[PREFETCH]-PLUS-1 ::= A-PLUS-B,ASPHI(R10),C000001-B,CLK-D
.MACRO D+ASPHI[PATTERN]-PLUS-020010-PLUS-1 ::= A-PLUS-B-PLUS-1,ASPHI(R17),CSPB(B17),CLK-D
.MACRO D+ASPHI[PATTERN]-AND-NOT-020010 ::= A-AND-NOT-B,ASPHI(R17),CSPB(B17),CLK-D

.MACRO D+BSPHI[PREFETCH]    ::= B,BSPHI(R10),CLK-D
.MACRO D+ASPLO[OVERLAP]-MINUS-CSPB[EXPEC] ::= A-MINUS-B,ASPLO(R10),CSPB(B17),CLK-D
.MACRO D+ASPHI[PREFETCH]-MINUS-CSPB[EXPEC] ::= A-MINUS-B,ASPHI(R10),CSPB(B17),CLK-D
.MACRO D+ASPLO[OVERLAP]-MINUS-BSPLO[OVERLAP] ::= A-MINUS-B,ASPLO(R10),BSPLO(R10),CLK-D

.MACRO CSPD[EXPEC]+EMIT    ::= CSPD[17]+EMIT

```

!.PAGE=====

```

.TOC *      SPECIFIC MACROS FOR BYTE/BYTE CONSTANT/D=ZERO TESTS

.MACRO D+ASPLO[DNONZERO]-PLUS-1 ::= A-PLUS-B,ASPLO(R11),C000001-B,CLK-D
.MACRO D+BSPLO[DZERO]-PLUS-1   ::= A-PLUS-B,C000001-A,BSPLO(R11),CLK-D
.MACRO D+BSPLO[IR-DATA]-PLUS-2  ::= A-PLUS-B-PLUS-1,C000001-A,BSPLO(R10),CLK-D
.MACRO D+BSPHI[WORD]-PLUS-1    ::= A-PLUS-B,C000001-A,BSPHI(R10),CLK-D

```

```

2485 .MACRO D+BSPL0(IR-DATA) ::= B,BSPL0(R10),CLK-D
2486 .MACRO D+BSPL0(DZERO) ::= B,BSPL0(R11),CLK-D
2487 .MACRO D+SR-MINUS-BSPHI(WORD) ::= A-MINUS-B,SR,BSPHI(R10),CLK-D,P3-T
2488 .MACRO D+ASPL0(DNONZERO)-MINUS-CSPD(17) ::= A-MINUS-B,ASPL0(R11),CSPD(D17),CLK-D,P3-T
2489 .MACRO D+ASPHI(BYTE-FIRST)-MINUS-CSPD(17) ::= A-MINUS-B,ASPHI(R10),CSPD(D17),CLK-D,P3-T
2490 .MACRO D+ASPL0(BYTE-SECOND)-MINUS-CSPD(17) ::= A-MINUS-B,ASPL0(R10),CSPD(D17),CLK-D,P3-T
2491 .MACRO D+ASPHI(BYTE-FIRST)-PLUS-CSP(1-0) ::= A-PLUS-B,ASPHI(R10),CSPD(D13),CLK-D
2492 .MACRO D+ASPL0(BYTE-SECOND)-PLUS-CSP(1-0) ::= A-PLUS-B,ASPL0(R10),CSPD(D13),CLK-D

```

!.PAGE=====

.TOC \* SUBROUTINE CALL MACROS

```

2499
2500 .MACRO CALL(DISPLAY) ::= GOTO-PAGE(7),J/DISPLAY
2501
2502 .MACRO CALL(DINTOIR) ::= GOTO-PAGE(7),J/DINTOIR
2503 .MACRO CALL(DINTOIR-5) ::= GOTO-PAGE(7),J/DINTOIRS
2504 .MACRO CALL(SRINTOIR) ::= GOTO-PAGE(7),J/SRINTOIR
2505 .MACRO CALL(SRINTOIR-5) ::= GOTO-PAGE(7),J/SRINTOIRS
2506
2507 .MACRO CALL(FLAGFPST00) ::= GOTO-PAGE(7),J/FLAGFPST00
2508 .MACRO CALL(PST00) ::= GOTO-PAGE(7),J/PST00
2509 .MACRO CALL(CUAT00) ::= GOTO-PAGE(7),J/CUAT00
2510 .MACRO CALL(CLRJAMT00) ::= GOTO-PAGE(7),J/CLRJAMT00
2511 .MACRO CALL(ODDJAMT00) ::= GOTO-PAGE(7),J/ODDJAMT00
2512 .MACRO CALL(JAMT00) ::= GOTO-PAGE(7),J/JAMT00
2513 .MACRO CALL(CLRSERVICET00) ::= GOTO-PAGE(7),J/CLRSERVICET00
2514 .MACRO CALL(DATISERVICET00) ::= GOTO-PAGE(7),J/DATISERVICET00
2515 .MACRO CALL(DATOSERVICET00) ::= GOTO-PAGE(7),J/DATOSERVICET00
2516 .MACRO CALL(CJESERVICET00) ::= GOTO-PAGE(7),J/CJESERVICET00
2517 .MACRO CALL(SERVICET00) ::= GOTO-PAGE(7),J/SERVICET00
2518 .MACRO CALL(PBAT00) ::= GOTO-PAGE(7),J/PBAT00
2519 .MACRO CALL(PSSSEQLO0) ::= GOTO-PAGE(7),J/PSSSEQLO0
2520 .MACRO CALL(FLAGFPSSEQLO0) ::= GOTO-PAGE(7),J/FLAGFPSSEQLO0
2521 .MACRO CALL(FLAGLO0) ::= GOTO-PAGE(7),J/FLAGFPS03
2522 .MACRO CALL(GETPROCDAT) ::= GOTO-PAGE(7),J/GETPROCDAT
2523
2524 .MACRO CALL(CLEAR-I-0-A) ::= GOTO-PAGE(7),J/CLEAR-I-0-A
2525 .MACRO CALL(CLEAR-I-0-B) ::= GOTO-PAGE(7),J/CLEAR-I-0-B
2526
2527 .MACRO CALL(BUSDINXOR125252) ::= GOTO-PAGE(7),J/BDX12
2528 .MACRO CALL(BUSDINXOR052525) ::= GOTO-PAGE(7),J/BDX05
2529 .MACRO CALL(CSP17XOR125252) ::= GOTO-PAGE(7),J/C17X12
2530 .MACRO CALL(CSP17XOR052525) ::= GOTO-PAGE(7),J/C17X05
2531
2532 .MACRO CALL(D15-12) ::= GOTO-PAGE(7),J/D[15-12]
2533 .MACRO CALL(D11-06) ::= GOTO-PAGE(7),J/D[11-06]
2534 .MACRO CALL(D05-00) ::= GOTO-PAGE(7),J/D[05-00]
2535 .MACRO CALL(DZERO) ::= GOTO-PAGE(7),J/DZERO
2536
2537 .MACRO CALL(ZEROSF04DF02) ::= GOTO-PAGE(7),J/ZEROSF04DF02
2538 .MACRO CALL(ZEROSF02DF04) ::= GOTO-PAGE(7),J/ZEROSF02DF04

```



```

2539 .MACRO CALL[ZEROSFDF] ::= GOTO-PAGE(7),J/ZEROSFDF
2540 .MACRO CALL[ZEROOF] ::= GOTO-PAGE(7),J/ZEROOF
2541
2542 .MACRO CALL[SFDFTOSR] ::= GOTO-PAGE(5),J/SFDFTOSR
2543
2544 .MACRO CALL[ALUCARRY1] ::= GOTO-PAGE(7),J/ALUCARRY1
2545 .MACRO CALL[ALUCARRY2] ::= GOTO-PAGE(7),J/ALUCARRY2
2546
2547 .MACRO CALL[LOADFPSCC] ::= GOTO-PAGE(7),J/LOADFPSCC
2548 .MACRO XFR-TO-BM[LOADNZW4] ::= GOTO-PAGE(4),J/LOADNZW4
2549
2550 .MACRO CALL[SETUPPSCC#DC] ::= GOTO-PAGE(7),J/SETUPPSCC#DC
2551 .MACRO CALL[PSCCTOSR3-0] ::= GOTO-PAGE(7),J/PSCCTOSR3-0
2552
2553 .MACRO CALL[CSP16XORSRTOIR-5] ::= GOTO-PAGE(7),J/CSP16XORSRTOIRS
2554 .MACRO CALL[CSP16XORFLTTOIR-5] ::= GOTO-PAGE(7),J/CSP16XORFLTTOIRS
2555
2556 .MACRO CALL[MFSS-TEST] ::= GOTO-PAGE(6),J/MFSS01
2557
2558 .MACRO CALL[KTSRCDST] ::= GOTO-PAGE(7),J/KTSRCDST01
2559 .MACRO CALL[KTDSTBSP] ::= GOTO-PAGE(7),J/KTSRCDST08
2560 .MACRO CALL[KTSRCBSP] ::= GOTO-PAGE(7),J/KTSRCDST07
2561
2562 .MACRO CALL[COUNT-TEST] ::= GOTO-PAGE(4),J/COUNTER01
2563

```

!.PAGE=====

```

2567 .TOC * JAM UPP LOG MACROS
2568
2569 !MACROS CONCERNED WITH CSP LOG AFTER UNEXPECTED JAMUPP
2570 !MACROS REQUIRE APPROPRIATE REGISTER ENABLED ON BUSDIN
2571
2572
2573 .MACRO CSPD[00]+LOG-CUA ::= CSPD(000),WR-CSP
2574 .MACRO CSPD[01]+LOG-SERVICE ::= CSPD(001),WR-CSP
2575 .MACRO CSPD[02]+LOG-JAM ::= CSPD(002),WR-CSP
2576

```

!\*\*\*\*\* END OF MACRO DEFINITIONS \*\*\*\*\*

!.PAGE=====

```

2583 .TOC * - - - MICRODIAGNOSTIC CODE - - - - -
2584
2585 .CODE
2586
2587
2588
2589
2590
2591
2592

```

2593  
2594  
2595  
2596  
2597  
2598  
2599  
2600  
2601  
2602  
2603  
2604  
2605  
2606  
2607  
2608  
2609  
2610  
2611  
2612  
2613  
2614  
2615  
2616  
2617  
2618  
2619  
2620  
2621  
2622  
2623  
2624  
2625  
2626  
2627  
2628  
2629  
2630  
2631  
2632  
2633  
2634  
2635  
2636  
2637  
2638  
2639  
2640  
2641  
2642  
2643

!.PAGE=====

!\*\*\* VERSION /V101A0/ \*\*\*

!\*\*\*\*\* MICRODIAGNOSTIC INITIAL STARTUP LOCATION \*\*\*\*\*

.TOC \* TEST001-007: NUA SEQUENCING

!\*\*\*\*\*

! \* TESTS: 001 - 007 UWORDS: 010 + 000

! \* FUNCTIONS: TESTS 001 - 007 TEST THE NUA SEQUENCING LOGIC.  
! \* PATTERNS ARE RUN THRU THE NUA LOGIC ESTABLISHING  
! \* THAT ALL BITS CAN BE SET AND CLEARED, AND THAT THE  
! \* STATE (SET, CLEARED) OF NO ONE BIT AFFECTS THE  
! \* ABILITY TO SET/CLEAR ANY OTHER BIT. THE PAGE  
! \* CHANGING FUNCTIONS OF BUTA(SUBR-A) AND BUTA(SUBR-B)  
! \* ARE ALSO TESTED FOR PAGES 4, 5, 6, & 7. NOTE  
! \* THAT THE RETURN ADDRESS SIMULTANEOUSLY LOADED  
! \* IS NOT CHECKED FOR VALIDITY AT THIS POINT.

! \* NOTES: TEST(N) DOES THE SETUP FOR TEST(N+1). THE ACTUAL TEST CONSISTS  
! \* OF BEING IN THE RIGHT PLACE (MICROWORD) AT THE RIGHT TIME.

!\*\*\*\*\*

!-----

!\*\*\* TEST 001 \*\*\*

!TEST NUA LOGIC WITH PATTERN "100 000 000 000"

4000:

TEST001:

PO, LOAD-ENUA(6252),  
LOAD-ERROR(TEST001),  
DCS-CTR(C1.)  
NEXT, PAGE(2) BUTA(SUBR-B),  
J/TEST002

!LOAD ENUA WITH ADDR(NEXT UWORD)  
!ERROR DIRECTORY KEY  
!COMPARE AT NEXT UWORD  
!CHANGING TO PAGE 2 (ACTUALLY 6), VIA SUBR-B  
!NOTE OVERLAP: NEXTPAGE(2:0)=ENUA(2:0)  
!NOTE ALSO DCS MODULE FORCES NUA(11)=1, SO  
!PAGES 2/6 ARE EQUIVALENT REFERENCES.

(4000) DCS[1.00.1.0.0.0] BM[1110..00.11..00.10..101..010...0.0.0..0..0...0.0000..0..0000.0...11.100...010.101.010]

2644  
2645  
2646  
2647  
2648  
2649  
2650  
2651  
2652  
2653  
2654  
2655  
2656  
2657  
2658  
2659  
2660  
2661  
2662  
2663  
2664  
2665  
2666  
2667  
2668  
2669  
2670  
2671  
2672  
2673  
2674  
2675  
2676  
2677  
2678  
2679  
2680  
2681  
2682  
2683  
2684  
2685  
2686  
2687  
2688  
2689  
2690  
2691  
2692  
2693  
2694

!-----

!\*\*\* TEST 002 \*\*\*  
!TEST NUA LOGIC WITH PATTERN "010 010 101 010"

6252:

TEST002:

PO, LOAD-ENUA(6631)  
LOAD-ERROR(TEST002),  
DCS-CTR(C1.),  
BUMP-VERIFY,  
NEXT J/TEST003

!LOAD ENUA WITH ADDR(NEXT UWORD)  
!ERROR DIRECTORY KEY  
!COMPARE AT NEXT UWORD  
!COUNT

(6252) DCS[1.00.1.0.0.1] BM[1110..00.11..01.10..011..001...0.0.0..0..0...0.0000...0..0000.0...11.00C...110.011.001]

!-----

!\*\*\* TEST 003 \*\*\*  
!TEST NUA LOGIC WITH PATTERN "010 110 011 001"

6631:

TEST003:

PO, LOAD-ENUA(5525)  
LOAD-ERROR(TEST003),  
DCS-CTR(C1.),  
NEXT, PAGE(5), BUTA(SUBR-A),  
J/TEST004

!LOAD ENUA WITH ADDR(NEXT UWORD)  
!ERROR DIRECTORY KEY  
!COMPARE AT NEXT UWORD  
!CHANGING TO PAGE 5, VIA SUBR-A  
!NOTE OVERLAP: NEXTPAGE<2:0>=ENUA<2:0>

(6631) DCS[1.00.1.0.0.0] BM[1110..00.10..11.01..010..101...0.0.0..0..0...0.0000...0..0000.0...11.101...101.010.101]

!-----

!\*\*\* TEST 004 \*\*\*  
!TEST NUA LOGIC WITH PATTERN "101 101 010 101"

5525:

TEST004:

PO, LOAD-ENUA(5146),  
LOAD-ERROR(TEST004),

!LOAD ENUA WITH ADDR(NEXT UWORD)  
!ERROR DIRECTORY KEY

2695 DCS-CTR(C1.), !COMPARE AT NEXT UWORD  
 2696 BUMP-VERIFY, !COUNT  
 2697 NEXT, J/TEST005  
 (5525) DCS[1.00.1.0.0.1] BM[1110..00.10..10.01..100..110...0.0.0..0..0...0.0000...0..0000.0...11.000...001.100.110]

2698  
 2699  
 2700  
 2701  
 2702  
 2703  
 2704  
 2705

! -----

2706  
 2707  
 2708 !\*\*\* TEST 005 \*\*\*  
 2709 !TEST NUA LOGIC WITH PATTERN "101 001 100 110"  
 2710 5146:  
 2711 TEST005:  
 2712 PO, LOAD-ENUA(4474) !LOAD ENUA WITH ADDR(NEXT UWORD)  
 2713 LOAD-ERROR(TEST005), !ERROR DIRECTORY KEY  
 2714 DCS-CTR(C1.), !COMPARE AT NEXT UWORD  
 2715 NEXT, PAGE(4), BUTA(SUBR-B), !CHANGING TO PAGE 4, VIA SUBR-B  
 2716 J/TEST006 !NOTE OVERLAP: NEXT PAGE<2:0>=ENUA<2:0>  
 (5146) DCS[1.00.1.0.0.0] BM[1110..00.10..01.00..111..100...0.0.0..0..0...0.0000...0..0000.0...11.100...100.111.100]

2717  
 2718  
 2719  
 2720  
 2721  
 2722  
 2723  
 2724  
 2725

! -----

2726  
 2727 !\*\*\* TEST 006 \*\*\*  
 2728 !TEST NUA LOGIC WITH PATTERN "100 100 111 100"  
 2729 4474:  
 2730 TEST006:  
 2731 PO, LOAD-ENUA(4377) !LOAD ENUA WITH ADDR(NEXT UWORD)  
 2732 LOAD-ERROR(TEST006), !ERROR DIRECTORY KEY  
 2733 DCS-CTR(C1.), !COMPARE AT NEXT UWORD  
 2734 BUMP-VERIFY, !COUNT  
 2735 NEXT, J/TEST007  
 (4474) DCS[1.00.1.0.0.1] BM[1110..00.10..00.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.000...011.111.111]

2736  
 2737  
 2738  
 2739  
 2740  
 2741  
 2742  
 2743  
 2744

! -----

2745  
2746  
2747  
2748  
2749  
2750  
2751  
2752  
2753  
2754  
  
2755  
2756  
2757  
2758  
2759  
2760  
2761  
2762  
2763  
2764  
2765  
2766  
2767  
2768  
2769  
  
2770  
2771  
2772  
2773  
2774  
2775  
2776  
2777  
2778  
2779  
2780  
2781  
2782  
2783  
2784  
2785  
2786  
2787  
2788  
2789  
2790  
2791  
2792  
2793  
2794

!\*\*\* TEST 007 \*\*\*  
!TEST NUA LOGIC WITH PATTERN "100 011 111 111"

4377:

TEST007:

PO, LOAD-ENUA(7303)  
LOAD-ERROR(TEST007),  
DCS-CTR(C1.)  
NEXT, PAGE(3), BUTA(SUBR-A),  
J/NEXT007

!LOAD ENUA WITH ADDR(NEXT UWORD)  
!ERROR DIRECTORY KEY  
!COMPARE AT NEXT UWORD  
!CHANGING TO PAGE 3 (ACTUALLY 7), VIA SUBR-A  
!NOTE OVERLAP: NEXTPAGE<2:0>=ENUA<2:0>

(4377) DCS(1.00.1.0.0.0) BM(1110..00.11..10.11..000..011...0.0.0..0..0...0.0000...0..0000.0...11.101...011.000.011)  
!NOTE ALSO DCS MODULE FORCES NUA<11>=1, SO  
!PAGES 3/7 ARE EQUIVALENT REFERENCES.

!-----

!\*\*\* TEST 007-1/2 \*\*\*  
!TEST NUA LOGIC WITH PATTERN "011 011 000 011"

7303:

NEXT007:

NEXT, PAGE(4), BUTA(SUBR-B),  
J/TEST010

!CHANGING TO PAGE 4, VIA SUBR B  
!OFF TO NEXT TEST

(7303) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..100...0.0.0..0..0...0.0000...0..0000.0...11.100...111.111.1011)

!.PAGE=====

.TOC \* TEST010-011: MICROSUBROUTINE OPERATION

!\*\*\*\*\*  
!\* TESTS: 010 - 011 UWORDS: 007 + 002 \*  
!\* FUNCTIONS: THESE TWO TESTS DETERMINE THAT THE RETURN REGISTER AND \*  
ASSOCIATED DECODE, MUX, AND ENABLING LOGIC IS ABLE TO \*  
LOAD THE 12-BIT NUA RETURN REGISTER FROM THE EMIT \*  
FIELD, AND THEN ENTER THE REGISTER CONTENTS ONTO THE \*  
UWORD ADDRESS BUS (WHEN ENABLED BY A BUTA[RETURN]) IN \*  
TIME TO FETCH THE NEXT MICROWORD. TWO ALTERNATING \*  
BIT TESTS ARE USED TO CHECK THAT EACH BIT CAN BE SET \*  
AND CLEARED, INDEPENDENT OF ADJACENT BITS. \*  
!\*\*\*\*\*

```

2795
2796
2797 ! -----
2798
2799 !*** TEST 010 ***
2800 !TEST RETURN LOGIC WITH PATTERN "1010 1010 0101" (5245)
2801 4775:
2802 TEST010:
2803     PD,      LOAD-ENUA(NEXT010),      !LOAD ENUA WITH EXPECTED RETURN ADDRESS
2804     NEXT,    LOAD-ERROR(TEST010),    !ERROR DIRECTORY KEY
2805     NEXT,    DCS-CTR(C4.),           !COMPARE ENUA:TNUA IN 4. MICROWORDS
2806     NEXT,    J/LOAD010              !GO LOAD
(4775) DCS(1.00.1.0.0.0) BM(1011..00.10..10.10..100..101...0.0.0..0..0...0.0000...0..0000.0...11.000...001.000.000)

2807
2808 4100: !(FREE)
2809 LOAD010:
2810     SETUP,  RETURN/NEXT010,         !SET RETURN ADDRESS FROM EMIT
2811     NEXT,   PAGE(7),               !"SUBR" IS ON PAGE 7
2812     NEXT,   BUTA(SUBR-B),          !WITH B VERSION
2813     NEXT,   J/SUBR010             !"SUBR" DISP. ON PAGE
(4100) DCS(0.00.0.0.C.0) BM(0101..00.01..01.00..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.000.000)

2814
2815 7000: !(FREE)
2816 SUBR010:
2817     SETUP,  RETURN/ERROR010,       !NOISE BITS IN EMIT-RETURN FIELD
2818     PD,     BUTA(CLR-FLAG-RES-UCON), !NOISE IN UBF FIELD
2819     NEXT,   J/ZTARGET777           !BEFORE DO BUTA[RETURN]
(7000) DCS(0.00.0.0.0.0) BM(0110..00.10..11.00..011..000...0.0.0..0..0...0.0000...0..0000.0...11.010...111.111.111)

2820
2821 ! REMAINDER OF SUBROUTINE IS AT ZTARGET777 (LOCATION 7777(8)), WHICH IMMEDIATELY DOES A BUTA[RETURN]
2822 ! TO THE ADDRESS IN THE RETURN REGISTER, WHICH SHOULD BE THE VALUE LOADED IN WORD
2823 ! LOAD010. THE RETURN REGISTER WAS SET TO POINT TO THE START OF THE NEXT TEST
2824 ! [NEXT010], AT WHICH POINT THE ENUA:TNUA COMPARE IS SET TO TAKE PLACE.
2825
2826 6543:
2827 ERROR010:
2828     PD,     BUMP-VERIFY,           !COUNT
2829     NEXT,   PAGE(4),              !IF WE END UP HERE, THE NOISE BITS
2830     NEXT,   J/TEST010            !WERE LOADED INTO THE RETURN REGISTER INSTEAD
(6543) DCS(0.00.0.0.0.1) BM(0000..00.00..00.00..000..100...0.0.0..0..0...0.0000...0..0000.0...11.000...111.111.101)

2831
2832 5245:
2833 NEXT010:
2834     PD,     BUMP-VERIFY,           !COUNT
2835     NEXT,   J/TEST011            !RETURNED OK, COMPARE ENUA:TNUA DONE HERE
(5245) DCS(0.00.0.0.0.1) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...111.111.111)

2836
2837
2838
2839
2840
2841

```

```

2842 ! -----
2843
2844 !*** TEST 011 ***
2845 !TEST RETURN LOGIC WITH PATTERN "1101 0101 1010" (6532)
2846 5777:
2847 TEST011:
2848     PO,      LOAD-ENUA(INITIALIZED1),      !LOAD ENUA WITH EXPECTED RETURN ADDRESS
2849     NEXT,    LOAD-ERROR(TEST011),          !ERROR DIRECTORY KEY
2850     DCS-CTR(C3.),      !COMPARE ENUA:TNUA IN 3. MICROWORDS
2851     J/LOAD011      !GO LOAD
(5777) DCS(1.00.1.0.0.0) BM(1100..00.11..01.01..011..010..0.0.0..0..0...0.0000...0..0000.0...11.000...000.000.001)

2852 5001: !(FREE)
2853 LOAD011:
2854     SETUP,   RETURN/INITIALIZED1,          !SET RETURN ADDRESS IN EMIT
2855     NEXT,    PAGE(7),                      !"SUBR" IS ON PAGE 7
2856     BUTA(SUBR-B),      !VIA B VERSION
2857     J/RESETUOMP      !"SUBR" DISP. ON PAGE
(5001) DCS(0.00.0.0.0.0) BM(0110..00.10..10.11..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.001)

2859 ! "ONE-WORD SUBROUTINE" IS AT LABEL "RESETUOMP", WHICH IMMEDIATELY DOES A BUTA(RETURN)
2860 ! TO THE ADDRESS IN THE RETURN REGISTER, WHICH SHOULD BE THE VALUE LOADED AT SYMBOLIC LABEL
2861 ! "LOAD011". THE RETURN REGISTER WAS SET TO POINT TO THE INITIALIZATION CODE ON THE NEXT PAGE:
2862 ! "INITIALIZED1" AT WHICH POINT THE ENUA:TNUA COMPARE IS SET TO TAKE PLACE.
2863 ! THIS SUBROUTINE ALSO 'ATTEMPTS' TO SET THE UCON CONTROL TO: 'PROC', 'BUSDIN+EMIT', AND
2864 ! 'EN-CLK-IR', FOR USE IN THE FOLLOWING INITIALIZATION ROUTINE.
2865 !
2866
2867
2868
2869 ! * * * * *
2870
2871 ! *** BUTA(RETURN) ERROR LOOP WORDS:
2872
2873 ! IF AT ANY TIME, A 'BUTA(RETURN)' FAILS, ONE OF THE FOLLOWING MICROWORDS WILL BE
2874 ! CONTINUOUSLY LOOPEd UPON. THIS IS DUE TO THE FACT THAT FOR EACH MICROWORD THAT
2875 ! CONTAINS THIS 'BUTA(RETURN)', THE UPF FIELD (WHICH WOULD NORMALLY NOT BE USED IN THIS
2876 ! MICROWORD EXECUTION), CONTAINS THE VALUE "J/BUTERROR#", WHERE NUMBER (#) IS THE
2877 ! CURRENT MICROWORD'S PAGE NUMBER (4-7).
2878
2879 4376:
2880 BUTERROR4:
2881     PO,      DCS-CTR(CO.),      !FORCE AN ERROR, ERROR-CODE=LAST LOADED
2882     NEXT,    J/BUTERROR4      !HANG UP
(4376) DCS(0.00.1.0.0.0) BM(1111..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...011.111.110)

2883
2884 5376:
2885 BUTERRORS:
2886     PO,      DCS-CTR(CO.),      !FORCE AN ERROR, ERROR-CODE=LAST LOADED
2887     NEXT,    J/BUTERRORS      !HANG UP
(5376) DCS(0.00.1.0.0.0) BM(1111..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...011.111.110)

2889
2890

```

```

2891 6376:
2892 BUTERROR6:
2893     PO,      DCS-CTR(CO.),      !FORCE AN ERROR, ERROR-CODE=LAST LOADED
2894     NEXT,    J/BUTERROR6        !HANG UP
(6376) DCS(0.00.1.0.0.0) BM(1111..00.00..00.00..000..000...0.0.0..0..0...0.0000...0 .0000.0...11.000...011.111.110)

```

```

2895
2896
2897 7376:
2898 BUTERROR7:
2899     PO,      !DCS-CTR(CO.),      !FORCE AN ERROR, ERROR-CODE=LAST LOADED
2900     NEXT,    J/BUTERROR7        !HANG UP
(7376) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...011.111.110)

```

```

2901
2902
2903

```

!.PAGE=====

.TOC \* INIT REGISTERS, CONSOLE DEFAULT ERROR DISPLAY, REVISION CODE

```

2904
2905
2906
2907
2908
2909 !*****
2910 !*
2911 !* TESTS:  INITIALIZATION          UWORDS:  011 + 011
2912 !*
2913 !* FUNCTIONS:
2914 !*
2915 !* TRY TO PUT AN OCTAL (000000) IN THE CONSOLE DISPLAY AS AN ERROR INDICATOR
2916 !*
2917 !* PUT REVISION CODE, BIT15 CLEAR, IN GPR R5
2918 !*
2919 !* SET FLAGS, FPS, PS, UBREAK REGISTERS TO ALL ZEROS TO DISABLE AS MUCH
2920 !* AS POSSIBLE ANY SPURIOUS HOT FLOATING POINT STARTUPS, UBREAKS, ETC.
2921 !*
2922 !* TURN OFF CACHE (SET BOTH FOFCE MISS BITS), AND TURN OFF KT (MEMORY
2923 !* MANAGEMENT (BY CLEARING ENABLE BIT).
2924 !*
2925 !*****

```

```

2926
2927
2928     !RETURNED OK, COMPARE OF ENUA:TNUA
2929     !FOR PREV TEST DONE HERE

```

```

2930 6532:
2931 INITIALIZE01:
2932     SETUP,   RETURN/INITIALIZED3,      !GO TO SUBR THAT PUTS REVISION NUMBER,
2933     NEXT,    GOTO-PAGE(7),             ! WITH B<15>=(0), INTO B.M. GPR "R5"
2934     J/INSERTREVN0
(6532) DCS(0.00.0.0.0.0) BM(0111..00.00..00.00..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.001.110)

```

```

2935
2936 7001:  !(FREE)
2937 INITIALIZE03:
2938     P3,      CSPD(17)+EMIT, EMIT/000014,      !EMITCON TO DISABLE CACHE, KT
2939     NEXT,    J/INITIALIZE04
(7001) DCS(0.00.0.0.0.0) BM(0000..10.00..00.00..001..100...0.0.0..0..0...0.0000...1..0000.0...11.000...000.000.011)

```



```

2940
2941 7003: !(FREE)
2942 INITIALIZE04:
2943     P2-T, D+CSPD(D17), D(C)+0, !GET ABOVE CONSTANT INTO D
2944     P2, RES+CSPD(D17), !BITS<14:11>=0/00/0 FOR SK LOAD, GUARD-DIS
2945     NEXT, J/INITIALIZE05
(7003) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...0.1.0..0..0...0.0000...0..1000.1...11.000...000.000.100)

2946
2947 7004: !(FREE)
2948 INITIALIZE05:
2949     P2-T, CCR(07-02)+D(07-02)-(1), !LOAD CCR, DISABLING CACHE
2950     NEXT, J/INITIALIZE06 ! BIT<3:2> SET FORCE MISS
(7004) DCS(0.00.0.0.0.0) BM(0001..00.01..00.00..000..000...0.0.0..0..0...1.1011...0..0000.0...11.000...000.000.101)

2951
2952 7005: !(FREE)
2953 INITIALIZE06:
2954     P2-T, MMRO+D-(1), !LOAD MMRO, DISABLING KT11
2955     NEXT, J/INITIALIZE07 ! BIT<00> CLEAR DISABLES KT11
(7005) DCS(0.00.0.0.0.0) BM(0001..00.11..01.00..000..000...0.0.0..0..0...1.1011...0..0000.0...11.000...000.000.110)

2956
2957 7006: !(FREE)
2958 INITIALIZE07:
2959     P2-T, D+ZERO, !DEFAULT TO ALL ZEROS FOR ERROR
2960     SR+ZERO, !ZERO SR FOR JAMUPP IS ERROR
2961     NEXT, GOTO-PAGE(4), !XFER
2962     J/INITIALIZE10
(7006) DCS(0.00.0.0.0.0) BM(0011..00.00..00.00..000..100...0.1.1..0..0...0.0000...0..0000.0...11.100...001.000.001)

2963
2964 4101: !(FREE)
2965 INITIALIZE10:
2966     SELECT, UCON-PROC, !ENABLE CLOCKING THE FOLLOWING
2967     ENABLE, EN-CLK-PS(15-12), EN-CLK-PS(7-4), !PROCESSOR REGISTERS:
2968     EN-CLK-PS(3-0), EN-CLK-FLAG(8-0),
2969     EN-CLK-FPS(7-4), EN-CLK-UBREAK(11-00),
2970     BUSDIN+EMIT(15-00), !FOR UBREAK CONSTANT
2971     P0, SET-UCON-CONTROL, !WRITE CONTROLS
2972     BUMP-VERIFY, !COUNT
2973     P3, BUTA(CUA-TRACK), !RESET CUA TRACKING
2974     NEXT, J/INITIALIZE11
(4101) DCS(0.00.0.0.0.1) BM(1000..01.00..00.01..110..011...0.0.0..0..0...1.1001...0..0000.0...11.001...001.000.010)

2975
2976 4102: !(FREE)
2977 INITIALIZE11:
2978     P0, DCS-CTR(C15.), !DISABLE DCS-CTR FOR NOW
2979     EMITC, EMIT/000000, !FOR UBREAK REGISTER LOAD
2980     P2, PS(3-0)+D(3-0),
2981     UBREAK+BUSDIN(11-00),
2982     P3, PS(15-12)+D(15-13), PS(7-4)+D(7-4),
2983     FPS(7-4)+D(7-4), FLAG(8-0)+D(15-8),
2984     NEXT, BUTD(SCOPE), !NO ERROR: "INITIALIZE12" (+1. WORDS)
2985     J/INITIALIZE12 ! ERROR: "TEST001" (BACK AT START)
(4102) DCS(0.00.1.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...1.1010...0..0000.0...11.000...000.000.001)

2986

```

```

2987 4001:
2988 INITIALIZE12:
2989     SETUP, RETURN/TEST012A, !RETURN TO NEXT TEST START
2990     NEXT,  CALL[DISPLAY] !GO DISPLAY CONTENTS OF D-REGISTER IN LIGHTS
(4001) DCS[0.00.0.0.0.0] BM[0101..00.11..11.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.010.001]

```

```

2991
2992
2993
2994 !.PAGE=====
2995

```

```

2996 .TOC * TEST012-050: IR DECODE (INSTR1, INSTR5, FLTPT, RELATED "BUTS")
2997

```

```

2998 !*****
2999 !*
3000 !* TESTS: 012 - 050 UWORDS: 274 + 206
3001 !*
3002 !* FUNCTIONS:
3003 !*
3004 !* THE FOLLOWING TESTS EXERCISE THE IR-DECODE RELATED LOGIC:
3005 !*
3006 !* INSTR-1, INSTR-5, FLOATING-POINT DECODE 'BUTS'
3007 !* IR<15:12>, IR<11> (TWO), IR<9:6>, IR<5:3> BIT 'BUTS'
3008 !* MOV/DR7, BYTE/DMD/SMD, DR6-7 DECODE-RELATED 'BUTS'
3009 !*
3010 !* NOTE ALSO THAT THE FIRST TIME THE PROCESSOR 'UCON':
3011 !* BUSDIN+EMIT, AND EN-CLK-IR, IS EMPLOYED IS IN TEST-012-A.
3012 !*
3013 !*****

```

```

3014
3015
3016
3017
3018
3019
3020
3021 ! - - - - -
3022

```

```

3023 !*** TEST 012 ***
3024 !TEST-012 USES A DATA PATTERN OF: "1 111 111 111 111 111" (177777)
3025

```

```

3026 ! - - - - -
3027

```

```

3028 !* PART A *
3029 !TEST-012-A CHECKS THAT BUT[IR<15:12>] READS THE "1111" IN IR<15:12>H CORRECTLY
3030

```

```

3031 5775:
3032 TEST012A:
3033     PO, LOAD-ENUA(ZTARGET417), !LOAD EXPECTED ADDRESS AFTER "BUT"
3034     LOAD-ERROR(TEST012A), !ERROR DIRECTORY KEY
3035     DCS-CTR(CS.), !COMPARE ENUA:TNUA IN 5. MICROWORDS
3036     NEXT, J/SETIRA !SETUP FOR IR TESTS
(5775) DCS[1.00.1.0.0.0] BM[1010..00.11..11.00..001..111...0.0.0..0..0...0.0000...0..0000.0...11.000...111.110.000]

```

```

3037 !*** SETUP PROCESSOR UCON FOR BUSDIN <- EMIT, CLOCKING INSTRUCTION REGISTER ***
3038 5760:
3039 SETIRA:
3040     SELECT, UCON-PROC           !SELECT PROCESSOR UCON CONTROL:
3041     ENABLE, EN-CLK-IR(15:00)   !  ENABLE CLOCK IR OPERATION
3042     BUSDIN+EMIT(15:00),       !  PUT EMIT(15:00) ONTO BUSDIN
3043     PO, SET-UCON-CONTROL,     !LOAD UCON REGISTER AT PO
3044     BUMP-VERIFY,             !COUNT
3045     NEXT, J/LOAD012A          !GO TO FIRST TEST, PART A
(5760) DCS(0.00.0.0.0.0.1) BM(0000..00.00..00.01..000..100...0.0.0..0..0...1.1001...0..0000.0...11.100...000.000.010)

```

```

3046
3047
3048 5002: !(FREE)
3049 LOAD012A:
3050     P2-U, IR+EMIT,           !LOAD IR WITH TEST PATTERN
3051     EMIT/177777,           ! (177777)
3052     NEXT, J/GOBUTO12A      !GO SETUP FOR "BUT"
(5002) DCS(0.00.0.0.0.0.0) BM(1111..00.11..11.11..111..111...0.0.0..0..0...1.1010...0..0000.0...11.000...000.000.011)

```

```

3053
3054
3055 5003: !(FREE)
3056 GOBUTO12A:
3057     SETUP, RETURN/TEST012B, !RETURN TO START OF NEXT SUBTEST
3058     NEXT, GOTO-PAGE(7),     !BUT TABLE IS ON PAGE 7
3059     J/BUTIR15-12,          !GO DO "BUT" ON IR<15:12>H
(5003) DCS(0.00.0.0.0.0.0) BM(0101..00.11..11.11..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.000)

```

```

3060
3061 ! - - - - -
3062
3063
3064 !* PART B *
3065 !TEST-012-B CHECKS THAT BUT(IR<11>#FLTPT<3:0>) READS THE "1" IN IR<11>H CORRECTLY,
3066 !AND THE FLTPT DECODE ROM GETS ADDRESS (776), WHICH IS A CMPF/D INSTR;
3067 !DATA OUTPUT SHOULD BE (17)
3068 5773:
3069 TEST012B:
3070     PO, LOAD-ENUA(ZTARGET437), !LOAD EXPECTED ADDRESS AFTER "BUT"
3071     LOAD-ERROR(TEST012B),     !ERROR DIRECTORY KEY
3072     DCS-CTR(C3.),            !COMPARE ENUA:TNUA IN 3. MICROWORDS
3073     NEXT, J/GOBUTO12B       !GO SETUP FOR "BUT"
(5773) DCS(1.00.1.0.0.0.0) BM(1100..00.11..11.00..011..111...0.0.0..0..0...0.0000...0..0000.0...11.000...000.000.100)

```

```

3074
3075
3076 5004: !(FREE)
3077 GOBUTO12B:
3078     SETUP, RETURN/TEST012C, !RETURN TO START OF NEXT SUBTEST
3079     NEXT, GOTO-PAGE(7),     !BUT TABLE IS ON PAGE 7
3080     J/BUTIR11FLTPT3-0,     !GO DO "BUT" ON IR<11>H#FLTPT<3:0>H
(5004) DCS(0.00.0.0.0.0.0) BM(0101..00.11..11.11..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.010)

```

```

3081 ! - - - - -
3082
3083

```

```

3084
3085 !* PART C *
3086 !TEST-012-C CHECKS THAT BUTR(IR<11>B) READS THE "1" IN IR<11>H CORRECTLY
3087 5771:
3088 TEST012C:
3089     PO,          LOAD-ENUA(ZTARGET403),      !LOAD EXPECTED ADDRESS AFTER "BUT"
3090                LOAD-ERROR(TEST012C),        !ERROR DIRECTORY KEY
3091                DCS-CTR(C3.),                !COMPARE ENUA:TNUA IN 3. MICROWORDS
3092                BUMP-VERIFY,                  !COUNT
3093     NEXT,        J/GOBUTO12C                 !GO SETUP FOR "BUT"
(5771) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...000.000.101)

```

```

3094
3095 5005: !(FREE)
3096 GOBUTO12C:
3097     SETUP,      RETURN/TEST012D,             !RETURN TO START OF NEXT SUBTEST
3098     NEXT,       GOTO-PAGE(7),                !BUT TABLE IS ON PAGE 7
3099                J/BUTIR11B                   !GO DO "BUT" ON IR<11>H
3100 (5005) DCS(0.00.0.0.0.0) BM(0101..00.11..11.10..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.010)

```

```

3101
3102 ! - - - - -
3103
3104

```

```

3105 !* PART D *
3106 !TEST-012-D CHECKS THAT BUT(IR<9:6>) READS THE "1111" IN IR<9:6>H CORRECTLY
3107 5767:
3108 TEST012D:
3109     PO,          LOAD-ENUA(ZTARGET417),      !LOAD EXPECTED ADDRESS AFTER "BUT"
3110                LOAD-ERROR(TEST012D),        !ERROR DIRECTORY KEY
3111                DCS-CTR(C3.),                !COMPARE ENUA:TNUA IN 3. MICROWORDS
3112     NEXT,        J/GOBUTO12D                 !GO SETUP FOR "BUT"
(5767) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..001..111...0.0.0..0..0...0.0000...0..0000.0...11.000...000.000.110)

```

```

3113
3114 5006: !(FREE)
3115 GOBUTO12D:
3116     SETUP,      RETURN/TEST012E,             !RETURN TO START OF NEXT SUBTEST
3117     NEXT,       GOTO-PAGE(7),                !BUT TABLE IS ON PAGE 7
3118                J/BUTIR9-6                   !GO DO "BUT" ON IR<9:6>H
3119 (5006) DCS(0.00.0.0.0.0) BM(0101..00.11..11.10..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.100)

```

```

3120
3121 ! - - - - -
3122
3123

```

```

3124 !* PART E *
3125 !TEST-012-E CHECKS THAT BUT(MOV-DR7#IR<5:3>) READS THE (-FLTPT*MOV+FLTPT*DR7) AND "111" IN IR<5:3>H CORRECTLY
3126 5765:
3127 TEST012E:
3128     PO,          LOAD-ENUA(ZTARGET417),      !LOAD EXPECTED ADDRESS AFTER "BUT"
3129                LOAD-ERROR(TEST012E),        !ERROR D.IRECTORY KEY
3130                DCS-CTR(C3.),                !COMPARE ENUA:TNUA IN 3. MICROWORDS
3131     NEXT,        J/GOBUTO12E                 !GO SETUP FOR "BUT"

```

```

3132 (5765) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..001..111...0.0.0..0..0...0.0000...0..0000.0...11.000...000.000.111)
3133
3134 5007: !(FREE)
3135 GOBUTO12E:
3136     SETUP, RETURN/TEST012F, !RETURN TO START OF NEXT SUBTEST
3137     NEXT,   GOTO-PAGE(7)      !BUT TABLE IS ON PAGE 7
3138     J/BUTMOVDR7IR5-3         !GO DO "BUT" ON (-FLTPT*MOV+FLTPT*DR7) # IR<5:3>H
(5007) DCS(0.00.0.0.0.0) BM(0101..00.11..11.10..100..111...0.0.0..0..0..0.0000...0..0000.0...11.100...011.000.101)

3139
3140 ! - - - - -
3141
3142 !* PART F *
3143 !TEST-012-F CHECKS THAT BUT(DR6-7 L) READS THE "11" IN IR<2:1> H CORRECTLY
3144 !AND DOES ASSERT THE SIGNAL
3145 5764:
3146 TEST012F:
3147     PO,      LOAD-ENUA(ZTARGET4G2), !LOAD EXPECTED ADDRESS AFTER "BUT"
3148             LOAD-ERROR( TEST012F), !ERROR DIRECTORY KEY
3149             DCS-CTR(C3.),          !COMPARE ENUA:TNUA IN 3. MICROWORDS
3150     NEXT,    J/GOBUTO12F           !GO SETUP FOR "BUT"
3151 (5764) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...000.001.000)

3152
3153 5010: !(FREE)
3154 GOBUTO12F:
3155     SETUP, RETURN/TEST012G, !RETURN TO START OF NEXT SUBTEST
3156     PO,    BUMP-VERIFY       !COUNT
3157     NEXT,  GOTO-PAGE(7),     !BUT TABLE IS ON PAGE 7
3158             J/BUTDR6-7L      !GO DO "BUT" ON DR 6/7 L
3159 (5010) DCS(0.00.0.0.0.1) BM(0101..00.11..11.10..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.111)

3160
3161 ! - - - - -
3162
3163 !* PART G *
3164 !TEST-012G CHECKS THAT BUT(INSTR1) READS THE ONES IN IR<15:00>H CORRECTLY
3165 !AS CLASS-A-FLIPT, AND CORRECTLY TARGETS TO INSTR1 FLIPT (474)
3166 ! BIT<1:0> = FLAG<4:5>H = "00" FROM INITIALIZATION CODE
3167 5763:
3168 TEST012G:
3169     PO,      LOAD-ENUA(ZTARGET474), !LOAD EXPECTED ADDRESS AFTER "BUT"
3170             LOAD-ERROR( TEST012G), !ERROR DIRECTORY KEY
3171             DCS-CTR(C3.),          !COMPARE ENUA:TNUA IN 3. MICROWORDS
3172     NEXT,    J/GOBUTO12G           !GO SETUP FOR "BUT"
3173 (5763) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..111..100...0.0.0..0..0...0.0000...0..0000.0...11.000...000.001.001)

3174
3175 5011: !(FREE)
3176 GOBUTO12G:
3177     SETUP, RETURN/SCOPE012, !RETURN TO SCOPE LOOP TEST WORD
3178

```

3179 NEX, GOTO-PAGE(7), !BUT TABLE IS ON PAGE 7  
 3180 J/BUTINSTR1 !GO DO INSTR1 "BUT"  
 (5011) DCS(0.00.0.0.0.0) BM(0101..00.00..00.01..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100. .011.000.110)

3181  
 3182  
 3183  
 3184 5012: !(FREE)  
 3185 SCOPE012:  
 3186 PO, BUMP-VERIFY, !COUNT  
 3187 NEXT, BUTD(SCOPE), !NO ERROR: "TEST013A" [+]. WORDS]  
 3188 J/TEST013A ! ERROR: "SETIRA" [-16. WORDS]  
 (5012) DCS(0.00.0.1.0.1) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...111.110.001)

3189  
 3190  
 3191  
 3192  
 3193  
 3194  
 3195 ! - - - - -  
 3196  
 3197 !\*\*\* TEST 013 \*\*\*  
 3198 !TEST-013 USES AN IR DATA PATTERN OF: "0 000 000 000 000 000" (000000)  
 3199  
 3200 ! - - - - -

3201  
 3202 !\* PART A \*  
 3203 !TEST-013-A CHECKS THAT BUT [IR<15:12>] READS THE "0000" IN IR<15:12>H CORRECTLY  
 3204 5761:  
 3205 TEST013A:  
 3206 PO, LOAD-ENUA(ZTARGET400), !LOAD EXPECTED ADDRESS AFTER "BUT"  
 3207 LOAD-ERROR(TEST013A), !ERROR DIRECTORY KEY  
 3208 DCS-CTR(C4.), !COMPARE ENUA:TNUA IN 4. MICROWORDS  
 3209 NEXT, J/LOAD013A !GO LOAD PATTERN  
 (5761) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...111.100.100)

3210  
 3211 5744:  
 3212 LOAD013A:  
 3213 P2-U, IR+EMIT, !LOAD IR WITH TEST PATTERN  
 3214 EMIT/000000, ! (000000)  
 3215 NEXT, J/GOBUTO13A !GO SETUP FOR "BUT"  
 3216 (5744) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...1.1010...0..0000.0...11.000...000.001.011)

3217  
 3218 5013: !(FREE)  
 3219 GOBUTO13A:  
 3220 SETUP, RETURN/TEST013B, !RETURN TO START OF NEXT SUBTEST  
 3221 NEXT, GOTO-PAGE(7), !BUT TABLE IS ON PAGE 7  
 3222 J/BUTIR15-12 !GO DO "BUT" ON IR<15:12>H  
 3223 (5013) DCS(0.00.0.0.0.0) BM(0101..00.11..11.01..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.000)  
 3224  
 3225

```

3226 ! - - - - -
3227
3228 !* PART B *
3229 !TEST-013-B CHECKS THAT BUT(IR<11>#FLTPT<3:0>) READS THE "0" IN IR<11>H CORRECTLY.
3230 !AND THE FLTPT DECODE ROM GETS ADDRESS (000), WHICH IS A SETF/SETI/CFCC INSTR;
3231 !DATA OUTPUT SHOULD BE (00)
3232 5757:
3233 TEST0138:
3234     PO,          LOAD-ENUA(ZTARGET400),          !LOAD EXPECTED ADDRESS AFTER "BUT"
3235                LOAD-ERROR(TEST0138),          !ERROR DIRECTORY KEY
3236                DCS-CTR(C3.),                  !COMPARE ENUA-TNUA IN 3. MICROWORDS
3237                BUMP-VERIFY,                   !COUNT
3238     NEXT,        J/GOBUTO138                   !GO SETUP FOR "BUT"
(5757) DCS(1.00.1.0.0.1) BM(1100..00.11..11.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...000.001.100)

3239
3240 5014: !(FREE)
3241 GOBUTO138:
3242     SETUP,      RETURN/TEST013C,              !RETURN TO START OF NEXT SUBTEST
3243     NEXT,       GOTO-PAGE(7),                 !BUT TABLE IS ON PAGE 7
3244                J/BUTIR11FLTPT3-0           !GO DO "BUT" ON IR<11>H#FLTPT<3:0>H
3245 (5014) DCS(0.00.0.0.0.0) BM(0101..00.11..11.01..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.010)

3246 ! - - - - -
3247
3248 !* PART C *
3249 !TEST-013-C CHECKS THAT BUTR(IR<11>B) READS THE "0" IN IR<11>H CORRECTLY
3250 5755:
3251 TEST013C:
3252     PO,          LOAD-ENUA(ZTARGET401),          !LOAD EXPECTED ADDRESS AFTER "BUT"
3253                LOAD-ERROR(TEST013C),          !ERROR DIRECTORY KEY
3254                DCS-CTR(C3.),                  !COMPARE ENUA:TNUA IN 3. MICROWORDS
3255     NEXT,        J/GOBUTO13C                   !GO SETUP FOR "BUT"
(5755) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...000.001.101)

3256
3257 5015: !(FREE)
3258 GOBUTO13C:
3259     SETUP,      RETURN/TEST013C,              !RETURN TO START OF NEXT SUBTEST
3260     NEXT,       GOTO-PAGE(7),                 !BUT TABLE IS ON PAGE 7
3261                J/BUTIR11B                    !GO DO BUT ON IR<11>H
3262 (5015) DCS(0.00.0.0.0.0) BM(0101..00.11..11.01..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.010)

3263 ! - - - - -
3264
3265 !* PART D *
3266 !TEST-013-D CHECKS THAT BUT(IR<9:6>) READS THE "0000" IN IR<9.6>H CORRECTLY
3267 5753:
3268 TEST013D:
3269     PO,          LOAD-ENUA(ZTARGET400),          !LOAD EXPECTED ADDRESS AFTER "BUT"
3270
3271
3272
3273

```

```

3274          LOAD-ERROR(TEST013D),          !ERROR DIRECTORY KEY
3275          DCS-CTR(C3.),                    !COMPARE ENUA:TNUA IN 3. MICROWORDS
3276          NEXT, J/GOBUTO13D                !GO SETUP FOR "BUT"
(5753) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...000.001.1101)

```

```

3277
3278
3279          5016: !(FREE)
3280          GOBUTO13D:
3281          SETUP, RETURN/TEST013E,          !RETURN TO START OF NEXT SUBTEST
3282          NEXT, GOTO-PAGE(7),              !BUT TABLE IS ON PAGE 7
3283          J/BUTIR9-6                        !GO DO "BUT" ON IR<9:6>H
(5016) DCS(0.00.0.0.0.0) BM(0101..00.11..11.01..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.1001)

```

```

3284
3285          ! - - - - -
3286
3287          !* PART E *
3288          !TEST-013-E CHECKS THAT BUT(MOV-DR7#IR<5:3>) READS THE -(-FLTPT*MOV+FLTPT*DR7) AND "000" IN IR<5:3>H CORRECTLY
3289          5751:
3290          TEST013E:

```

```

3291          PO, LOAD-ENUA(ZTARGET400),        !LOAD EXPECTED ADDRESS AFTER "BUT"
3292          LOAD-ERROR(TEST013E),           !ERROR DIRECTORY KEY
3293          DCS-CTR(C3.),                    !COMPARE ENUA:TNUA IN 3. MICROWORDS
3294          NEXT, J/GOBUTO13E                !GO SETUP FOR "BUT"
3295          (5751) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...000.001.1111)

```

```

3296
3297          5017: !(FREE)
3298          GOBUTO13F:
3299          SETUP, RETURN/TEST013F,          !RETURN TO START OF NEXT SUBTEST
3300          PO, B#MP-VERIFY,                 !COUNT
3301          NEXT, GOTO-PAGE(7),              !BUT TABLE IS ON PAGE 7
3302          J/BUTMOVDR7IR5-3                 !GO DO "BUT" ON (-FLTPT*MOV+FLTPT*DR7) # IR<5:3>H
3303          (5017) DCS(0.00.0.0.0.1) BM(0101..00.11..11.01..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.1011)

```

```

3304
3305          ! - - - - -
3306
3307          !* PART F *
3308          !TEST-013-F CHECKS THAT BUT(INSTR5) READS THE ZEROS IN IR<15:00>H CORRECTLY
3309          !AS (00000)=HALT, AND CORRECTLY TARGETS TO (434)
3310          5750:
3311          TEST013F:

```

```

3312          PO, LOAD-ENUA(ZTARGET434),        !LOAD EXPECTED ADDRESS AFTER "BUT"
3313          LOAD-ERROR(TEST013F),           !ERROR DIRECTORY KEY
3314          DCS-CTR(C3.),                    !COMPARE ENUA:TNUA IN 3. MICROWORDS
3315          NEXT, J/GOBUTO13F                !GO SETUP FOR "BUT"
3316          (5750) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...000.010.0001)

```

```

3317
3318          5020: !(FREE)
3319          GOBUTO13F:
3320

```



```

3321          SETUP, RETURN/TEST013G,          !RETURN TO START OF NEXT SUBTEST
3322          NEXT,  GOTO-PAGE(7),              !BUT TABLE IS ON PAGE 7
3323          J/BUTINSTRS                        !GO DO INSTRS "BUT"
(5020) DCS(0.00.0.0.0.0) BM(0101..00.11..11.00..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.001)
3324          ! - - - - -
3325          !* PART G *
3326          !TEST-013-G CHECKS THAT BUT(INSTR1) READS THE ZEROS IN IR<15:00>H CORRECTLY
3327          !AS NOT(CLASS-A THRU G), AND CORRECTLY TARGETS TO (417) (OTHER)
3328          5747:
3329          TEST013G:
3330          PO,          LOAD-ENUA(ZTARGET417),          !LOAD EXPECTED ADDRESS AFTER "BUT"
3331          LOAD-ERROR(TEST013G),          !ERROR DIRECTORY KEY
3332          DCS-CTR(C3.),          !COMPARE ENUA:TNUA IN 3. MICROWORDS
3333          NEXT, J/GOBUTO13G          !GO SETUP FOR "BUT"
(5747) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..001..111...0.0.0..0..0...0.0000...0..0000.0...11.000...000.010.001)
3336          5021: !(FREE)
3337          GOBUTO13G:
3338          SETUP, RETURN/SCOPE013,          !RETURN TO SCOPE LOOP TEST WORD
3339          NEXT,  GOTO-PAGE(7),              !BUT TABLE IS ON PAGE 7
3340          J/BUTINSTR1                    !GO DO INSTR1 "BUT"
(5021) DCS(0.00.0.0.0.0) BM(0101..00.00..00.10..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110)
3343          5022: !(FREE)
3344          SCOPE013:
3345          NEXT,  BUTD(SCOPE1),          !NO ERROR: "TEST014A" [+1. WORD]
3346          J/TEST014A                    ! ERROR: "LOAD013A" [-14. WORDS]
(5022) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...111.100.101)
3350          ! - - - - -
3351          !*** TEST 014 ***
3352          !TEST-014 USES AN IR DATA PATTERN OF: "0 000 000 001 010 101" (000125)
3353          ! - - - - -
3354          !* PART A *
3355          !TEST-014-A CHECKS THAT BUT(INSTR5) READS THE IR CORRECTLY
3356          !AS ROM ADDRESS=(125) ON THE INSTR5 E8B ROM, AND RECEIVES THE DIAGNOSTIC VALUE
3357          !OF (12), TARGETING TO (432) AFTER THE DECODE
3358          5745:
3359          TEST014A:

```

3369 PO, LOAD-ENUA(ZTARGET432), !LOAD EXPECTED ADDRESS AFTER "BUT"  
 3370 LOAD-ERROR(TEST014A), !ERROR DIRECTORY KEY  
 3371 DCS-CTR(C4.), !COMPARE ENUA:TNUA IN 4. MICROWORDS  
 3372 NEXT, J/LOAD014A !GO LOAD PATTERN  
 (5745) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..011..010...0.0.0..0..0...0.0000...0..0000.0...11.000...111.100.010)

3373  
 3374  
 3375 5742:  
 3376 LOAD014A:  
 3377 P2-U, IR+EMIT, !LOAD IR WITH TEST PATTERN  
 3378 EMIT/000125, ! (000125)  
 3379 NEXT, J/GOBUTO14A !GO SETUP FOR "BUT"  
 (5742) DCS(0.00.0.0.0.0) BM(0000..00.00..00.01..010..101...0.0.0..0..0...1.1010...0..0000.0...11.000...000.010.011)

3380  
 3381 5023: !(FREE)  
 3382 GOBUTO14A:  
 3383 SETUP, RETURN/TEST014B, !RETURN TO START OF NEXT SUBTEST  
 3384 NEXT, GOTO-PAGE(7), !BUT TABLE IS ON PAGE 7  
 3385 J/BUTINSTRS !GO DO INSTRS "BUT"  
 (5023) DCS(0.00.0.0.0.0) BM(0101..00.11..11.01..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.001)

3386  
 3387  
 3388 ! - - - - -  
 3389

3390 !\* PART B \*  
 3391 !TEST-014-B CHECKS THAT BUT(DR6-7 L) READS THE "10" IN IR<2:1> H CORRECTLY  
 3392 !AND DOES NOT ASSERT THE SIGNAL  
 3393 5752:  
 3394 TEST014B:

3395 PO, LOAD-ENUA(ZTARGET403), !LOAD EXPECTED ADDRESS AFTER "BUT"  
 3396 LOAD-ERROR(TEST014B), !ERROR DIRECTORY KEY  
 3397 DCS-CTR(C3.), !COMPARE ENUA:TNUA IN 3. MICROWORDS  
 3398 BUMP-VERIFY, !COUNT  
 3399 NEXT, J/GOBUTO14B !GO SETUP FOR "BUT"  
 (5752) DCS(1.00.1.0.0.1) BM(1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...000.010.100)

3400  
 3401  
 3402 5024: !(FREE)  
 3403 GOBUTO14B:  
 3404 SETUP, RETURN/TEST014C, !RETURN TO START OF NEXT SUBTEST  
 3405 NEXT, GOTO-PAGE(7), !BUT TABLE IS ON PAGE 7  
 3406 J/BUTDR6-7L !GO DO DR 6-7 L "BUT"  
 (5024) DCS(0.00.0.0.0.0) BM(0101..00.11..11.10..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.111)

3407  
 3408 ! - - - - -  
 3409

3410 !\* PART C \*  
 3411 !TEST-014-C CHECKS THAT BUT(INSTR1) READS THE IR CORRECTLY  
 3412 !AS CLASS-E=JMP, IR<5:3>H="010", AND TARGETS TO (652)  
 3413 5762:  
 3414 TEST014C:  
 3415 PO, LOAD-ENUA(ZTARGET652), !LOAD EXPECTED ADDRESS AFTER "BUT"

```

3416          LOAD-ERROR(TEST014C),          !ERROR DIRECTORY KEY
3417          DCS-CTR(C3.),                    !COMPARE ENUA:TNUA IN 3. MICROWORDS
3418          NEXT, J/GOBUTO14C                !GO SETUP FOR "BUT"
(5762) DCS(1.00.1.0.0.0) BM(1100..00.11..11.10..101..010...0.0.0..0..0...0.0000...0..0000.0...11.000...000.010.101)

3419
3420
3421          5025: !(FREE)
3422          GOBUTO14C:
3423          SETUP, RETURN/TEST014D,          !RETURN TO START OF NEXT SUBTEST
3424          NEXT,  GOTO-PAGE(7),            !BUT TABLE IS ON PAGE 7
3425          J/BUTINSTR1                     !GO DO INSTR1 "BUT"
(5025) DCS(0.00.0.0.0.0) BM(0101..00.11..10.01..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110)

3426
3427
3428
3429          ! - - - - -
3430
3431          !* PART D *
3432          !TEST-014-D CHECKS THAT BUT(IR<11>#FLTPT<3:0>) READS THE "0" IN IR<11>H CORRECTLY,
3433          !AND THE FLTPT DECODE ROM GETS ADDRESS (020), WHICH IS A STST INSTR;
3434          !DATA OUTPUT SHOULD BE (01)
3435          5710:
3436          TEST014D:
3437          PO,          LOAD-ENUA(ZTARGET401),          !LOAD EXPECTED ADDRESS AFTER "BUT"
3438          LOAD-ERROR(TEST014D),                    !ERROR DIRECTORY KEY
3439          DCS-CTR(C3.),                    !COMPARE ENUA:TNUA IN 3. MICROWORDS
3440          NEXT, J/GOBUTO14C                !GO SETUP FOR "BUT"
(5710) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...000.010.110)

3441
3442
3443          5026: !(FREE)
3444          GOBUTO14D:
3445          SETUP, RETURN/SCOPE014,          !RETURN TO SCOPE LOOP TEST WORD
3446          NEXT,  GOTO-PAGE(7),            !BUT TABLE IS ON PAGE 7
3447          J/BUTIR11FLTPT3-0                !GO DO "BUT" ON IR<11>H#FLTPT<3:0>H
(5026) DCS(0.00.0.0.0.0) BM(0101..00.00..00.10..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.010)

3448
3449
3450
3451          5027: !(FREE)
3452          SCOPE014:
3453          NEXT,  BUTD(SCOPE),              !NO ERROR: "TEST015A" [+1. WORD]
3454          J/TEST015A                          ! ERROR: "LOAD014A" [-8. WORDS]
(5027) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...111.100.011)

3455
3456
3457
3458
3459
3460
3461          ! - - - - -
3462

```

```

3463 **** TEST 015 ***
3464 !TEST-015 USES AN IR DATA PATTERN OF: "0 000 000 001 101 010" (000152)
3465
3466 ! - - - - -
3467
3468 !* PART A *
3469 !TEST-015-A CHECKS THAT BUT(INSTR5) READS THE IR CORRECTLY
3470 !AS ROM ADDRESS=(152) ON THE INSTR5 E88 ROM, AND RECEIVES THE DIAGNOSTIC VALUE
3471 !OF (05), TARGETING TO (425) AFTER THE DECODE
3472 5743:
3473 TEST015A:
3474     PO,          LOAD-ENUA(ZTARGET425),          !LOAD EXPECTED ADDRESS AFTER "BUT"
3475                LOAD-ERROR(TEST015A),          !ERROR DIRECTORY KEY
3476                DCS-CTR(C4.),                  !COMPARE ENUA:TNUA IN 4. MICROWORDS
3477     NEXT,        J/LOAD015A                    !GO LOAD PATTERN
(5743) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..010..101...0.0.0..0..0...0.0000...0..0000.0...11.000...111.100.000)

3478
3479
3480 5740:
3481 LOAD015A:
3482     PO,          BUMP-VERIFY,                  !COUNT
3483     P2-U,        IR+EMIT,                      !LOAD IR WITH TEST PATTERN
3484     NEXT,        EMIT/000152,                  ! (000152)
3485     J/GOBUTO15A, J/GOBUTO15A                  !GO SETUP FOR "BUT"
(5740) DCS(0.00.0.0.0.1) BM(0000..00.00..00.01..101..010...0.0.0..0..0...1.1010...0..0000.0...11.000...000.011.000)

3486
3487 5030: !(FREE)
3488 GOBUTO15A:
3489     SETUP,       RETURN/TEST015B,             !RETURN TO START OF NEXT SUBTEST
3490     NEXT,        GOTO-PAGE(?),               !BUT TABLE IS ON PAGE ?
3491     J/BUTINSTR5, J/BUTINSTR5                 !GO DO INSTR5 "BUT"
(5030) DCS(0.00.0.0.0.0) BM(0101..00.11..11.11..010..111...0.0.0..0..0...0.000J...0..0000.0...11.100...011.000.001)

3492
3493 ! - - - - -
3494
3495
3496 !* PART B *
3497 !TEST-015-B CHECKS THAT BUT(DR6 7 L) READS THE "01" IN IR<2:1> H CORRECTLY
3498 !AND DOES NOT ASSERT THE SIGNAL
3499 5772:
3500 TEST015B:
3501     PO,          LOAD-ENUA(ZTARGET403),          !LOAD EXPECTED ADDRESS AFTER "BUT"
3502                LOAD-ERROR(TEST015B),          !ERROR DIRECTORY KEY
3503                DCS-CTR(C3.),                  !COMPARE ENUA:TNUA IN 3. MICROWORDS
3504     NEXT,        J/GOBUTO15B                    !GO SETUP FOR "BUT"
(5772) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0.. 11.000...000.011.001)

3505
3506
3507 5031: !(FREE)
3508 GOBUTO15B:
3509     SETUP,       RETURN/TEST015C,             !RETURN TO START OF NEXT SUBTEST
3510     NEXT,        GOTO-PAGE(?),               !BUT TABLE IS ON PAGE ?
3511     J/BUTDR6-7L, J/BUTDR6-7L                 !GO DO DR 6-7 L "BJT"

```

(5031) DCS(0.00.0.0.0.0) BM(0101..00.11..10.10..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.111)

3512  
3513  
3514  
3515  
3516  
3517  
3518  
3519  
3520  
3521  
3522  
3523

!\* PART C \*

!TEST-015-C CHECKS THAT BUT(INSTR1) READS THE IR CORRECTLY  
!AS CLASS-E=JMP, IR<5:3>H="101", AND TARGETS TO (655)

5722:

TEST015C:

PO, LOAD-ENUA(ZTARGET655),  
LOAD-ERROR(TEST015C),  
DCS-CTR(C3.),  
NEXT J/GOBUTO15C

!LOAD EXPECTED ADDRESS AFTER "BUT"  
!ERROR DIRECTORY KEY  
!COMPARE ENUA:TNUA IN 3. MICROWORDS  
!GO SETUP FOR "BUT"

(5722) DCS(1.00.1.0.0.0) BM(1100..00.11..11.10..101..101...0.0.0..0..0...0.0000...0..0000.0...11.000...000.011.010)

3524  
3525  
3526  
3527  
3528  
3529  
3530

5032: !(FREE)

GOBUTO15C:

SETUP, RETURN/SCOPE015,  
NEXT, GOTO-PAGE(7),  
J/BUTINSTR1

!RETURN TO SCOPE LOOP TEST WORD  
!BUT TABLE IS ON PAGE 7  
!GO DO INSTR1 "BUT"

(5032) DCS(0.00.0.0.0.0) BM(0101..00.00..00.11..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110)

3531  
3532  
3533  
3534  
3535  
3536

5033: !(FREE)

SCOPE015:

PO, BUMP-VERIFY,  
NEXT, BUT(SCOPE),  
J/TEST016A

!COUNT  
!NO ERROR: "TEST016A" [+1. WORD]  
!ERROR: "LOAD015A" [-6. WORDS]

(5033) DCS(0.00.0.1.0.1) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...111.100.001)

3537  
3538  
3539  
3540  
3541  
3542  
3543  
3544  
3545  
3546  
3547  
3548  
3549

!\*\*\* TEST 016 \*\*\*

!TEST-016 USES AN IR DATA PATTERN OF: "1 000 000 000 000 000" (100000)

!\* PART A \*

!TEST-016-A CHECKS THAT BUT(INSTR1) READS THE IR CORRECTLY  
!AS CLASS-F=BRANCH, IR<14:11>H="0000", IR<15#10:08>H#"0000",  
!AND TARGETS TO (757)

5741:

TEST016A:

PO, LOAD-ENUA(ZTARGET757),  
LOAD-ERROR(TEST016A),  
DCS-CTR(C4.),  
NEXT, J/LOAD016A

!LOAD EXPECTED ADDRESS AFTER "BUT"  
!ERROR DIRECTORY KEY  
!COMPARE ENUA:TNUA IN 4. MICROWORDS  
!GO LOAD PATTERN

```

3560 (5741) DCS(1.00.1.0.0.0) BM(1011..00.11..11.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.000...111.011.1101
3561
3562 5736:
3563 LOAD016A:
3564 P2-U, IR+EMIT, !LOAD IR WITH TEST PATTERN
3565 EMIT/100000, !:(100000)
3566 NEXT, J/GOBUTO16A !GO SETUP FOR "BUT"
(5736) DCS(0.00.0.0.0.0) BM(1000..00.00..00.00..000..000...0.0.0..0..0...1.1010...0..0000.0...11.000...000.011.1001
3567
3568 5034: !(FREE)
3569 GOBUTO16A:
3571 SETUP, RETURN/SCOPE016, !RETURN TO SCOPE LOOP TEST WORD
3572 PO, BUMP-VERIFY, !COUNT
3573 NEXT, GOTO-PAGE(7), !BUT TABLE IS ON PAGE 7
3574 J/BUTINSTR1 !GO DO INSTR1 "BUT"
(5034) DCS(0.00.0.0.0.1) BM(0101..00.00..00.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.1101
3575
3576 5035: !(FREE)
3577 SCOPE016:
3578 NEXT, BUTD(SCOPE), !NO ERROR: "TEST017A" [+1. WORD]
3579 J/TEST017A ! ERROR: "LOAD016A" [-2. WORDS]
3580 (5035) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...111.011.1111
3581
3582
3583
3584
3585
3586
3587 ! - - - - -
3588
3589 !*** TEST 017 ***
3590 !TEST-017 USES A DATA PATTERN OF: "0 000 000 101 010 010" (000522)
3591
3592 ! - - - - -
3593
3594 !* PART A *
3595 !TEST-017-A CHECKS THAT BUT(IR<9:6>) READS THE ALTERNATING PATTERN "0101"
3596 !IN IR<9:6>H CORRECTLY
3597 5737:
3598 TEST017A:
3599 PO, LOAD-ENUA(ZTARGET405), !LOAD EXPECTED ADDRESS AFTER "BUT"
3600 LOAD-ERROR(TEST017A), !ERROR DIRECTORY KEY
3601 DCS-CTR(C4.), !COMPARE ENUA:TNUA IN 4. MICROWORDS
3602 NEXT, J/LOAD017A !GO LOAD PATTERN
(5737) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...111.011.0101
3603
3604
3605 5732:
3606 LOAD017A:
3607 P2-U, IR+EMIT, !LOAD IR WITH TEST PATTERN

```

```

3608          EMIT/000522,          !(000522)
3609      NEXT,  J/GOBUTO17A        !GO SETUP FOR "BUT"
(5732) DCS(0.00.0.0.0.0) BM(0000..00.00..01.01..010..010...0.0.0..0..0...1.1010...0..0000.0...11.000...000.011.110)
3610
3611      5036: !(FREE)
3612      GOBUTO17A:
3613          SETUP,  RETURN/TEST017B,  !RETURN TO START OF NEXT SUBTEST
3614          NEXT,  GOTO-PAGE(7),      !BUT TABLE IS ON PAGE 7
3615          J/BUTIR3-6                !GO DO "BUT" ON IR<9:6>H
3616      (5036) DCS(0.00.0.0.0.0) BM(0101..00.11..10.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.100)
3617
3618      ! - - - - -
3619
3620      !* PART B *
3621      !TEST-017-B CHECKS THAT BUT(INSTR1) READS THE IR CORRECTLY
3622      !AS CLASS-F=BRANCH, IR<14:11>H="0000", IR<15:10:08>H="0000",
3623      !AND TARGETS TO (757)
3624      5735:
3625      TEST017B:
3626          PO,          LOAD-ENUA(ZTARGET757),  !LOAD EXPECTED ADDRESS AFTER "BUT"
3627          LOAD-ERROR(TEST017B),  !ERROR DIRECTORY KEY
3628          DCS-TR(C3.),  !COMPARE ENUA:TNUA IN 3. MICROWORDS
3629          BUMP-VERIFY,  !COUNT
3630          NEXT,  J/GOBUTO17B        !GO SETUP FOR "BUT"
3631      (5735) DCS(1.00.1.0.0.1) BM(1100..00.11..11.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.000...000.011.111)
3632
3633      5037: !(FREE)
3634      GOBUTO17B:
3635          SETUP,  RETURN/SCOPE017,  !RETURN TO SCOPE LOOP TEST WORD
3636          NEXT,  GOTO-PAGE(7),      !BUT TABLE IS ON PAGE 7
3637          J/BUTINSTR1              !GO DO INSTR1 "BUT"
3638      (5037) DCS(0.00.0.0.0.0) BM(0101..00.00..01.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110)
3639
3640      5040: !(FREE)
3641      SCOPE017:
3642          NEXT,  BUTD(SCOPE),  !NO ERROR: "TEST020A" [+1. WORD]
3643          J/TEST020A          !ERROR: "LOAD017A" [-4. WORDS]
3644      (5040) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...111.011.011)
3645
3646
3647
3648
3649
3650
3651      ! - - - - -
3652
3653      !*** TEST 020 ***
3654      !TEST-020 USES A DATA PATTERN OF: "0 000 001 010 101 111" (001257)

```

```

3655
3656 ! - - - - -
3657
3658 !* PART A *
3659 !TEST-020-A CHECKS THAT BUT[IR<9:6>] READS THE ALTERNATING PATTERN "1010"
3660 !IN IR<9:6>H CORRECTLY
3661 5733:
3662 TEST020A:
3663     PO,          LOAD-ENUA(ZTARGET412),          !LOAD EXPECTED ADDRESS AFTER "BUT"
3664                LOAD-ERROR(TEST020A),          !ERROR DIRECTORY KEY
3665                DCS-CTR(C4.),                  !COMPARE ENUA:TNUA IN 4. MICROWORDS
3666     NEXT,        J/LOAD020A                    !GO LOAD PATTERN
(5733) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...111.010.100)

3667
3668
3669 5724:
3670 LOAD020A:
3671     P2-U,        IR+EMIT,                      !LOAD IR WITH TEST PATTERN
3672                EMIT/001257,                  ! (001257)
3673     NEXT,        J/GOBUTO20A                    !GO SETUP FOR "BUT"
(5724) DCS(0.00.0.0.0.0) BM(0000..00.00..10.10..101..111...0.0.0..0..0...1.1010...0..0000.0...11.000...000.100.001)

3674
3675
3676 5041: !(FREE)
3677 GOBUTO20A:
3678     SETUP,      RETURN/TEST020B,              !RETURN TO START OF NEXT SUBTEST
3679     NEXT,        GOTO-PAGE(7),                !BUT TABLE IS ON PAGE 7
3680     J/BUTIR9-6, !GO DO "BUT" ON IR<9:6>H
(5041) DCS(0.00.0.0.0.0) BM(0101..00..1..10.11..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.100)

3681
3682 ! - - - - -
3683
3684 !* PART B *
3685 !TEST-020-B CHECKS THAT BUT[MOV-DR7#IR<5:3>] READS THE --(FLTPT*MOV+FLTPT*DR7) AND
3686 !ALTERNATING PATTERN "101" IN IR<5:3>H CORRECT.Y
3687
3688 5731:
3689 TEST020B:
3690     PO,          LOAD-ENUA(ZTARGET405),          !LOAD EXPECTED ADDRESS AFTER "P"
3691                LOAD-ERROR(TEST020B),          !ERROR DIRECTORY KEY
3692                DCS-CTR(C3.),                  !COMPARE ENUA:TNUA IN 3. MICROWORDS
3693     NEXT,        J/GOBUTO20B                    !GO SETUP FOR "BUT"
(5731) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...000.100.010)

3694
3695
3696 5042: !(FREE)
3697 GOBUTO20B:
3698     SETUP,      RETURN/TEST020C,              !RETURN TO START OF NEXT SUBTEST
3699     PO,          BUMP-VERIFY,                  !COUNT
3700     NEXT,        GOTO-PAGE(7),                !BUT TABLE IS ON PAGE 7
3701     J/BUTMOVDR7IR5-3, !GO DO "BUT"
(5042) DCS(0.00.0.0.0.1) BM(0101..00.11..10.10..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.101)

```



3702  
3703  
3704  
3705  
3706  
3707  
3708  
3709  
3710  
3711  
3712  
3713  
3714  
3715  
  
3716  
3717  
3718  
3719  
3720  
3721  
3722  
  
3723  
3724  
3725  
3726  
3727  
3728  
  
3729  
3730  
3731  
3732  
3733  
3734  
3735  
3736  
3737  
3738  
3739  
3740  
3741  
3742  
3743  
3744  
3745  
3746  
3747  
3748  
3749  
3750

! - - - - -

!\* PART C \*  
!TEST-020-C CHECKS THAT BUT(INSTR1) READS THE IP CORRECTLY  
!AS CLASS-F=BRANCH, IR<14:11>H="0000", IR<15:10:08>H="0000",  
!AND TARGETS TO (757)

5727:  
TEST020C:  
    PO,          LOAD-ENUA(ZTARGET757),          !LOAD EXPECTED ADDRESS AFTER "BUT"  
                  LOAD-ERROR(TEST020C),          !ERROR DIRECTORY KEY  
                  DCS-CTR(C3.),                  !COMPARE ENUA:TNUA IN 3. MICROWORDS  
    NEXT,        J/GOBUTO20C                  !GO SETUP FOR "BUT"  
(5727) DCS[1.00.1.0.0.0] BM[1100..00.11..11.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.000...000.100.011]

5043: !(FREE)  
GOBUTO20C:  
    SETUP,      RETURN/SCOPE020,          !RETURN TO SCOPE LOOP TEST WORD  
    NEXT,      GOTO-PAGE(7),              !BL\* TABLE IS ON PAGE 7  
              J/BUTINSTR1                  !GO DO INSTR1 "BUT"  
(5043) DCS[0.00.0.0.0.0] BM[0101..00.00..01.00..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110]

5044: !(FREE)  
SCOPE020:  
    NEXT,      BUTD(SCOPE),              !NO ERROR: "TEST021A" [+1. WRD]  
              J/TEST021A                  !  ERROR: "LOAD020A" [-6. WORDS]  
(5044) DCS[0.00.0.1.0.0] BM[0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...111.010.101]

! - - - - -

!\*\*\* TEST 021 \*\*\*  
!TEST-021 USES A DATA PATTERN OF: "0 000 010 011 001 101" (002315)

! - - - - -

!\* PART A \*  
!TEST-021-A CHECKS THAT BUT(DMO#SMO#BYTE)  
!DM=IR<5:3>H="001", DMOH=0; SM=IR<11:9>H="010", SMOH=0; BYTE H=0

5725:  
TEST021A:  
    PO,          LOAD-ENUA(ZTARGET400),          !LOAD EXPECTED ADDRESS AFTER "BUT"  
                  LOAD-ERROR(TEST021A),          !ERROR DIRECTORY KEY  
                  DCS-CTR(C4.),                  !COMPARE ENUA:TNUA IN 4. MICROWORDS  
    NEXT,        J/LOAD021A                  !GO LOAD PATTERN  
(5725) DCS[1.00.1.0.0.0] BM[1011..00.11..11.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...111.010.000]

```

3751
3752
3753 5720:
3754 LOAD021A,
3755 PO, BUMP-VERIFY, !COUNT
3756 P2-U, IR+EMIT, !LOAD IR WITH TEST PATTERN
3757 EMIT/002315, !(002315)
3758 NEXT, J/GOBUT021A !GO SETUP FOR "BUT"
(5720) DCS(0.00.0.0.0.1) BM(0000..00.01..00.11 .001..101...0.0.0..0..0...1.1010...0..0000.0...11.000...000.100.101)

```

```

3759
3760
3761 5045: !(FREE)
3762 GOBUT021A:
3763 SETUP, RETURN/TEST021B, !RETURN TO START OF NEXT SUBTEST
3764 NEXT, GOTO-PAGE(7), !BUT TABLE IS ON PAGE 7
3765 J/BUTDMSBYTE !GO DO "BUT" ON DMO#SMD#BYTE
(5045) DCS(0.00.0.0.0.0) BM(0101..00.11..10.10..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.011)

```

! - - - - -

```

3766
3767
3768 !* PART B *
3769 !TEST-021-B CHECKS THAT BUT(INSTR1) READS THE IR CORRECTLY.
3770 !AS CLASS-F=BRANCH, IR<14:11>H="0000", IR<15:10:08>H="0000",
3771 !AND TARGETS TO (757)
3772 5723:
3773 TEST021B:
3774 PO, LOAD-ENUA(ZTARGET757), !LOAD EXPECTED ADDRESS AFTER "BUT"
3775 LOAD-ERROR(TEST021B), !ERROR DIRECTORY KEY
3776 DCS-CTR(C3.), !COMPARE ENUA:TNUA IN 3. MICROWORDS
3777 NEXT, J/GOBUT021B !GO SETUP FOR "BUT"
(5723) DCS(1.00.1.0.0.0) BM(1100..00.11..11.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.000...000.100.110)

```

```

3780
3781
3782 5046: !(FREE)
3783 GOBUT021B:
3784 SETUP, RETURN/TEST021C, !RETURN TO START OF NEXT SUBTEST
3785 NEXT, GOTO-PAGE(7), !BUT TABLE IS ON PAGE 7
3786 J/BUTINSTR1 !GO DO INSTR1 "BUT"
(5046) DCS(0.00.0.0.0.0) BM(0101..00.11..11.11..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110)

```

! - - - - -

```

3787
3788
3789 !* PART C *
3790 !TEST-021-C CHECKS THAT BUT(IR<11>#FLTPT<3:0>) READS THE "0" IN IR<11>H CORRECTLY,
3791 !AND THE FLTPT DECODE ROM GETS ADDRESS (462), WHICH IS A ADDF/D MODEL-7 INSTR;
3792 !DATA OUTPUT SHOULD BE (11)
3793 5774:
3794 TEST021C:
3795 PO, LOAD-ENUA(ZTARGET411), !LOAD EXPECTED ADDRESS AFTER "BUT"
3796

```

```

3799          LOAD-ERROR(TEST021C),          !ERROR DIRECTORY KEY
3800          DCS-CTR(C3.),                    !COMPARE ENUA:TNUA IN 3. MICROWORDS
3801          NEXT, J/GOBUTO21C                !GO SETUP FOR "BUT"
(5774) DCS[1.00.1.0.0.0] BM[1100..00.11..11.00..001..001...0.0.0..0..0...0.0000...0..0000.0...11.000...000.100.111]

3802
3803
3804          5047: !(FREE)
3805          GOBUTO21C:
3806          SETUP, RETURN/SCOPE021,          !RETURN TO SCOPE LOOP TEST WORD
3807          NEXT, GOTO-PAGE(7),              !BUT TABLE IS ON PAGE 7
3808          J/BUTIR11FLTP3-0                 !GO DO "BUT" ON IR<11>H#FLTP3<3:0>H
(5047) DCS[0.00.0.0.0.0] BM[0101..00.00..01.01..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.010]

3809
3810
3811          5050: !(FREE)
3812          SCOPE021:
3813          NEXT, BUTD[SCOPE1],              !NO ERROR: "TEST022A" [+1. WORDS]
3814          J/TEST022A                        ! ERROR: "LOAD021A" [-6. WORDS]
(5050) DCS[0.00.0.1.0.0] BM[0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...111.010.001]

3815
3816
3817
3818
3819
3820
3821          ! -----
3822
3823          !*** TEST 022 ***
3824          !TEST-022 USES A DATA PATTERN OF: "0 000 011 000 110 011" (003063)
3825
3826          ! -----
3827
3828          !* PART A *
3829          !TEST-022-A CHECKS THAT BUT[INSTR1] READS THE IR CORRECTLY
3830          !AS CLASS-F=BRANCH, IR<14:11>H="0000", IR<15#10:08>H#="0000",
3831          !AND TARGETS TO (757)
3832          5721:
3833          TEST022A:
3834          PO, LOAD-ENUA(ZTARGET757),          !LOAD EXPECTED ADDRESS AFTER "BUT"
3835          LOAD-ERROR(TEST022A),              !ERROR DIRECTORY KEY
3836          DCS-CTR(C4.),                       !COMPARE ENUA:TNUA IN 4. MICROWORDS
3837          NEXT, J/LOAD022A                    !GO LOAD PATTERN
(5721) DCS[1.00.1.0.0.0] BM[1011..00.11..11.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.000...111.001.110]

3838
3839
3840          5716:
3841          LOAD022A:
3842          P2-U, IR+EMIT,                       !LOAD IR WITH TEST PATTERN
3843          EMIT/003063,                         ! (003063)
3844          NEXT, J/GOBUTO22A                    !GO SETUP FOR "BUT"
(5716) DCS[0.00.0.0.0.0] BM[0000..00.01..10.00..110..011...0.0.0..0..0...1.1010...0..0000.0...11.000...000.101.001]

3845
3846

```

```

3847 5051: !(FREE)
3848 GOBUTO22A:
3849     SETUP, RETURN/SCOPE022, !RETURN TO SCOPE LOOP TEST WORD
3850     NEXT, GOTO-PAGE(7), !BUT TABLE IS ON PAGE 7
3851     J/BUTINSTR1 !GO DO INSTR1 "BUT"
(5051) DCS(0.00.0.0.0.0) E (0101..00.00..01.01..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110)

```

```

3852
3853
3854
3855 5052: !(FREE)
3856 SCOPE022:
3857     NEXT, BUTD(SCOPE), !NO ERROR: "TEST023A" [+1. WORD]
3858     J/TEST023A ! ERROR: "LOAD022A" [-2. WORDS]
(5052) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...C.0.0..0..0...0.0000...0..0000.0...11.000...111.001.11:)

```

```

3859
3860
3861
3862
3863
3864
3865 ! - - - - -
3866
3867 !*** TEST 023 ***
3868 !TEST-023 USES A DATA PATTERN OF: "1 000 110 000 000 100" (106004)
3869 ! - - - - -
3870
3871

```

```

3872 !* PART A *
3873 !TEST-023-A CHECKS THAT BUT(INSTR1) READS THE IR CORRECTLY
3874 !AS CLASS-D=SOP*DMO; IR<14:06>H="000 110 000"; IR<15>H="1";
3875 !DM=IR<5:3>H="000", DMOH=1; AND TARGETS TO (560)
3876 5717:
3877 TEST023A:
3878     PO, LOAD-ENUA(ZTARGET560), !LOAD EXPECTED ADDRESS AFTER "BUT"
3879     LOAD-ERROR(TEST023A), !ERROR DIRECTORY KEY
3880     DCS-CTR(C4.), !COMPARE ENUA:TNUA IN 4. MICROWORDS
3881     NEXT, J/LOAD023A !GO LOAD PATTERN
(5717) DCS(1.00.1.0.0.0) BM(1011..00.11..11.01..110..000...0.0.0..0..0...0.0000...0..0000.0...11.000...111.001.100)

```

```

3882
3883
3884 5714:
3885 LOAD023A:
3886     P2-U, IR+EMIT, !LOAD IR WITH TEST PATTERN
3887     EMIT/106004, !(106004)
3888     NEXT, J/GOBUTO23A !GO SETUP FOR "BUT"
(5714) DCS(0.00.0.0.0.0) BM(1000..00.11..00.00..000..100...0.0.0..0..0...1.1010...0..0000.0...11.000...000.101.011)

```

```

3889
3890
3891 5053: :(FREE)
3892 GOBUTO23A:
3893     SETUP, RETURN/SCOPE023, !RETURN TO SCOPE LOOP TEST WORD
3894     NEXT, GOTO-PAGE(7), !BUT TABLE IS ON PAGE 7
3895     J/BUTINSTR1 !GO DO INSTR1 "BUT"

```

(5053) DCS(0.00.0.0.0.0) BM(0101..00.00..01.01..100..111...0.0.0..0.0...0.0000...0..0000.0...11.100...011.000.1101

3896  
3897  
3898  
3899  
3900  
3901  
3902

5054: !(FREE)

SCOPE023:

PO, BUMP-VERIFY,  
NEXT, BUTD(SCOPE),  
J/TEST024A

!COUNT

!MC ERROR: "TEST024A" [+1. WORD]

!ERROR: "LOAD023A" [-2. WORDS]

(5054) DCS(0.00.0.1.0.1) BM(0000..00.00..00.00..000..000...0.0.0..0.0...0.0000...0..0000.0...11.000...111.001.1011

3903  
3904  
3905  
3906  
3907  
3908

! - - - - -

!\*\*\* TEST 024 \*\*\*

!TEST-024 USES A DATA PATTERN OF: "0 000 101 111 000 110" (005706)

3911  
3912  
3913  
3914  
3915

! - - - - -

!\* PART A \*

!TEST-024-A CHECKS THAT BUT(DR6-7 L) READS THE "11" IN IR<2:1> H CORRECTLY

!AND DOES ASSERT THE SIGNAL

3916  
3917  
3918  
3919  
3920  
3921  
3922  
3923  
3924

5715:

TEST024A:

PO, LOAD-ENUA(ZTARGET402),  
LOAD-ERROR(TEST024A),  
DCS-CTR(C4.),  
NEXT, J/LOAD024A

!LOAD EXPECTED ADDRESS AFTER "BUT"

!ERROR DIRECTORY KEY

!COMPARE ENUA:TNUA IN 4. MICROWORDS

(5715) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..000..010...0.0.0..0.0...0.0000...0..0000.0...11.000...111.000.1101

3925  
3926  
3927  
3928  
3929  
3930  
3931  
3932

5706:

LOAD024A:

PO, BUMP-VERIFY,  
P2-U, IR+EMIT,  
EMIT/005706,  
NEXT, J/GOBUTO24A

!COUNT

!LOAD IR WITH TEST PATTERN

!(005706)

!GO SETUP FOR "BUT"

(5706) DCS(0.00.0.0.0.1) BM(0000..00.10..11.11..000..110...0.0.0..0.0...1.1010...0..0000.0...11.000...000.101.1011

3933  
3934  
3935  
3936  
3937  
3938  
3939

5055: !(FREE)

GOBUTO24A:

SETUP, RETURN/TEST024B,  
NEXT, GOTO-PAGE(7),  
J/BUTDR6-7L

!RETURN TO START OF NEXT SUBTEST

!BUT TABLE IS ON PAGE 7

!GO DO "BUT" ON DR 6/7 L

(5055) DCS(0.00.0.0.0.0) BM(0101..00.11..10.01..001..111...0.0.0..0.0...0.0000...0..0000.0...11.100...011.100.1111

3940  
3941  
3942

! - - - - -

```

3943
3944 : * PART B *
3945 !TEST-024-B CHECKS THAT BUT(INSTR1) READS THE IR CORRECTLY
3946 !AS CLASS-D=SOP*DMD; I<14:06>H="000 101 111"; IR<15>H="0";
3947 !DM=IR<5:3>H="000", DMDH=1; AND TARGETS TO (557)
3948 5711:
3949 TEST024B:
3950     PO,          LOAD-ENUA(ZTARGET557),          !LOAD EXPECTED ADDRESS AFTER "BUT"
3951                LOAD-ERROR(TEST024B),          !ERROR DIRECTORY KEY
3952                DCS-CTR(C3.),                  !COMPARE ENUA:TNUA IN 3. MICROWORDS
3953     NEXT,        J/GOBUT024B                   !GO SETUP FOR "BUT"
(5711) DCS(1.00.1.0.0.0) BM(1100..00.11..11.01..101..111...0.0.0..0..0...0.0000...0..0000.0...11.000...000.101.110)

3954
3955
3956 5056: !(FREE)
3957 GOBU*024B:
3958     SETUP,      RETURN/SCOPE024,              !RETURN TO SCOPE LOOP TEST WORD
3959     PO,         BUMP-VERIFY,                  !COUNT
3960     NEXT,       GOTO-PAGE(7),                 !BUT TABLE IS ON PAGE 7
3961     J/BUTINSTR1, !GO DO INSTR1 "BUT"
(5056) DCS(0.00.0.0.0.1) BM(0101..00.00..01.01..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110)

3962
3963
3964 5057: !(FREE)
3965 SCOPE024:
3966     NEXT,       BUTD(SCOPE1),                 !NO ERROR: "TEST025A" [+1. WORD]
3967     J/TEST025A, !ERROR: "LOAD024A" [-4. WORDS]
(5057) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...111.000.111)

3968
3969
3970
3971
3972
3973
3974 ! - - - - -
3975
3976 !*** TEST 025 ***
3977 !TEST-025 USES A DATA PATTERN OF: "0 000 110 011 000 011" (006303)
3978 ! - - - - -
3979
3980
3981 : * PART A *
3982 !TEST-025-A CHECKS THAT BUT(I 5(R1)) READS THE IR CORRECTLY
3983 !AS CLASS-D=SOP*DMD; IR<14:0>H="000 110 011"; IR<15>H="0";
3984 !DM=IR<5:3>H="000", DMDH=1 AND TARGETS TO (543)
3985 5707:
3986 TEST025A:
3987     PO,          LOAD-ENUA(ZTARGET543),          !LOAD EXPECTED ADDRESS AFTER "BUT"
3988                LOAD-ERROR(TEST025A),          !ERROR DIRECTORY KEY
3989                DCS-CTR(C4.),                  !COMPARE ENUA:TNUA IN 4. MICROWORDS
3990     NEXT,        J/LOAD025A                   !GO LOAD PATTERN
(5707) DCS(1.00.1.0.0.0) BM(1011..00.11..11.01..100..011...0.0.0..0..0...0.0000...0..0000.0...11.000...111.000.100)
3991

```

```

3992
3993 5704:
3994 LOAD025A:
3995     P2-U,      IR+EMIT,          !LOAD IR WITH TEST PATTERN
3996     EMIT/006303,      ! (006303)
3997     NEXT,      J/GOBUT025A,      !GO SETUP FOR "BUT"
(5704) DCS(0.00.0.0.0.0) BM(0000..00.11..00.11..000..011...0.0.0..0..0...1.1010...0..0000.0...11.000...000.110.000)

3998
3999
4000 5060: !(FREE)
4001 GOBUT025A:
4002     SETUP,      RETURN/SCOPE025,  !RETURN TO SCOPE LOOP TEST WORD
4003     NEXT,      GOTO-PAGE(7),      !BUT TABLE IS ON PAGE 7
4004     J/BUTINSTR1, !GO DO INSTR1 "BUT"
(5060) DCS(0.00.0.0.0.0) BM(10101..00.00..01.10..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110)

4005
4006
4007 5061: !(FREE)
4008 SCOPE025:
4009     NEXT,      BUTD(SCOPE),      !NO ERROR: "TEST026A" [+1: WORD]
4010     J/TEST026A,      !ERROR: "LOAD025A" [-2: WORDS]
(5061) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...111.000.101)

4011
4012
4013
4014
4015
4016
4017 ! -----
4018
4019 !*** TEST 026 ***
4020 !TEST-026 USES A DATA PATTERN OF: "1 000 101 100 000 011" (105403)
4021 ! -----
4022
4023
4024 !* PART A *
4025 !TEST-026-A CHECKS THAT BUT(INSTR1) READS THE IR CORRECTLY
4026 !AS CLASS-D=SOP*DMO; IR<14:06>H="000 101 100"; IR<15>H="1";
4027 !DM=IR<5:3>H="000", DMOH=1; AND TARGETS TO (574)
4028 5705:
4029 TEST026A:
4030     PO,      LOAD-ENUA(ZTARGETS574), !LOAD EXPECTED ADDRESS AFTER "BUT"
4031     LOAD-ERROR(TEST026A), !ERROR DIRECTORY KEY
4032     DCS-CTR(C4.), !COMPARE ENUA:TNUA IN 4. MICROWORDS
4033     BUMP-VERIFY, !COUNT
4034     NEXT,      J/LOAD026A, !GO LOAD PATTERN
(5705) DCS(1.00.1.0.0.1) BM(1011..00.11..11.01..111..100...0.0.0..0..0...0.0000...0..0000.0...11.000...111.000.010)

4035
4036
4037 5702:
4038 LOAD026A:
4039     P2-U,      IR+EMIT,          !LOAD IR WITH TEST PATTERN
4040     EMIT/105403,      ! (105403)

```

4041 (5702) DCS(0.00.0.0.0.0) BM(1000.00.10.11.00.000.011...0.0.0.0.0.0...1.1010...0.0000.0...11.000...000.110.010) NEXT, J/GOBUTO26A !GO SETUP FOR "BUT"

4042  
4043  
4044 5062: !(FREE)  
4045 GOBUTO26A:  
4046 SETUP, RETURN/SCOPE026, !RETURN TO SCOPE LOOP TEST WORD  
4047 PO, BUMP-VERIFY, !COUNT  
4048 NEXT, GOTO-PAGE(7), !BUT TABLE IS ON PAGE 7  
4049 J/BUTINSTR1 !GO DO INSTR1 "BUT"  
(5062) DCS(0.00.0.0.0.1) BM(10101.00.00.01.10.011.111...0.0.0.0.0.0...0.0000...0.0000.0...11.100...011.000.110)

4050  
4051 5063: !(FREE)  
4052 SCOPE026:  
4053  
4054 NEXT, BUTD(SCOPE), !NO ERROR: "TEST027A" [+1. WORD]  
4055 J/TEST027A !ERROR: "LOAD026A" [-2. WORDS]  
(5063) DCS(0.00.0.1.0.0) BM(0000.00.00.00.00.000.000...0.0.0.0.0.0...0.0000...0.0000.0...11.000...111.000.011)

4056  
4057  
4058  
4059  
4060  
4061  
4062 ! - - - - -  
4063  
4064 !\*\*\* TEST 027 \*\*\*  
4065 !TEST-027 USES A DATA PATTERN OF: "1 000 110 001 000 010" (106102)  
4066  
4067 ! - - - - -

4068  
4069 !\* PART A \*  
4070 !TEST-027-A CHECKS THAT BUT(INSTR1) READS THE IR CORRECTLY  
4071 !AS CLASS-D=SOP\*DMO; IR<14:06>H="000 110 001"; IR<15>H="1";  
4072 !DM=IR<5:3>H="000", DMOH=1; AND TARGETS TO (561)  
4073 5703:  
4074 TEST027A:  
4075 PO, LOAD-ENUA(ZTARGET561), !LOAD EXPECTED ADDRESS AFTER "BUT"  
4076 LOAD-ERROR(TEST027A), !ERROR DIRECTORY KEY  
4077 DCS-CTR(C4.), !COMPARE ENUA:TNUA IN 4. MICROWORDS  
4078 NEXT, J/LOAD027A !GO LOAD PATTERN  
(5703) DCS(1.00.1.0.0.0) BM(1011.00.11.11.01.110.001...0.0.0.0.0.0...0.0000...0.0000.0...11.000...111.000.000)

4079  
4080 5700:  
4081 LOAD027A:  
4082 P2-U, IR+EMIT, !LOAD IR WITH TEST PATTERN  
4083 EMIT/106102, !(106102)  
4084 NEXT, J/GOBUTO27A !GO SETUP FOR "BUT"  
4085 (5700) DCS(0.00.0.0.0.0) BM(1000.00.11.00.01.000.010...0.0.0.0.0.0...1.1010...0.0000.0...11.000...000.110.100)

4086  
4087 5064: !(FREE)  
4088



```

4089 GOBUTO27A:
4090     SETUP, RETURN/SCOPE027,           !RETURN TO SCOPE LOOP TEST WORD
4091     NEXT,  GOTO-PAGE(7),             !BUT TABLE IS ON PAGE 7
4092     J/BU*INSTR1                       !GO DO INSTR1 "BUT"
(5064) DCS(0.00.0.0.0.0) BM(0101..00.00..01.10..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110)

4093
4094     5065: !(FREE)
4095     SCOPE027:
4096     PO,      BUMP-VERIFY,             !COUNT
4097     NEXT,    BUTD(SCOPE),             !NO ERROR: "TEST030A" [+1. WORD]
4098     J/TEST030A                       ! ERROR: "LOAD027A" [-2. WORDS]
(5065) DCS(0.00.0.1.0.1) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...111.000.001)

4099
4100
4101
4102
4103
4104
4105 ! -----
4106
4107 !*** TEST 030 ***
4108 !TEST-030 USES A DATA PATTERN OF: "0 000 101 010 000 100" (005204)
4109 ! -----
4110
4111 !* PART A *
4112 !TEST-030-A CHECKS THAT BUT(INSTR1) READS THE IR CORRECTLY
4113 !AS CLASS-D=SOP*DMO; IR<14:06>H="000 101 010"; IR<15>H="0";
4114 !DM=IR<5:3>H="000", DMOH=1; AND TARGETS TO (552)
4115 5701:
4116 TEST030A:
4117     PO,      LOAD-ENUA(ZTARGET552),     !LOAD EXPECTED ADDRESS AFTER "BUT"
4118     LOAD-ERROR(TEST030A),             !ERROR DIRECTORY KEY
4119     DCS-CTR(C4.),                     !COMPARE ENUA:TNUA IN 4. MICROWORDS
4120     NEXT,    J/LOAD030A                !GO LOAD PATTERN
4121 (5701) DCS(1.00.1.0.0.0) BM(1011..00.11..11.01..101..010...0.0.0..0..0...0.0000...0..0000.0...11.000...110.111.110)

4122
4123
4124     5676:
4125     LOAD030A:
4126     P2-U,   IR+EMIT,                   !LOAD IR WITH TEST PATTERN
4127     EMIT/005204,                       ! (005204)
4128     NEXT,   J/GOBUTO30A                !GO SETUP FOR "BUT"
(5676) DCS(0.00.0.0.0.0) BM(0000..00.10..10.10..000..100...0.0.0..0..0...1.1010...0..0000.0...11.000...000.110.110)

4129
4130
4131     5066: !(FREE)
4132     GOBUTO30A:
4133     SETUP,  RETURN/TEST030B,           !RETURN TO START OF NEXT SUBTEST
4134     NEXT,   GOTO-PAGE(7),             !BUT TABLE IS ON PAGE 7
4135     J/BUTINSTR1                       !GO DO INSTR1 "BUT"
(5066) DCS(0.00.0.0.0.0) BM(0101..00.11..10.01..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110)

```

```

4136
4137
4138
4139
4140 ! - - - - -
4141
4142 !* PART B *
4143 !TEST-030-B CHECKS THAT BUT(IR<11>#FLTPT<3:0>) READS THE "1" IN IR<11>H CORRECTLY,
4144 !AND THE FLTPT DECODE ROM GETS ADDRESS (240), WHICH IS A MULF/MODE-0 INSTR,
4145 !DATA SHOULD BE (04)
4146 5712:
4147 TEST030B:
4148     PO,          LOAD-ENUA(ZTARGET424),          !LOAD EXPECTED ADDRESS AFTER "BUT"
4149                 LOAD-ERROR(TEST030B),          !ERROR DIRECTORY KEY
4150                 DCS-CTR(C3.),                  !COMPARE ENUA:TNUA IN 3. MICROWORDS
4151     NEXT,        J/GOBUTO30B
(5712) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..010..100...0.0.0..0..0...0.0000...0..0000.0...11.00C...000.110.111)

4152
4153 5067: !(FREE)
4154 GOBUTO30B:
4155     SETUP,      RETURN/TEST030C,          !RETURN TO START OF NEXT SUBTEST
4156     NEXT,       GOTO-PAGE(7),            !BUT TABLE IS ON PAGE 7
4157                 J/BUTIR11FLTPT3-0       !GO DO "BUT" ON IR<11>H#FLTPT<3:0>H
4158 (5067) DCS(0.00.0.0.0.0) BM(0101..00.11..10.0!..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.010)

4159
4160 ! - - - - -
4161
4162 !* PART C *
4163 !TEST-030-C CHECKS THAT BUTR(IR<11>B) READS THE "1" IN IR<11>H CORRECTLY
4164 5713:
4165 TEST030C:
4166     PO,          LOAD-ENUA(ZTARGET403),          !LOAD EXPECTED ADDRESS AFTER "BUT"
4167                 LOAD-ERROR(TEST030C),          !ERROR DIRECTORY KEY
4168                 DCS-CTR(C3.),                  !COMPARE ENUA:TNUA IN 3. MICROWORDS
4169     NEXT,        J/GOBUTO30C              !GO SETUP FOR "BUT"
(5713) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...000.111.000)

4171
4172 5070: !(FREE)
4173 GOBUTO30C:
4174     SETUP,      RETURN/TEST030C,          !RETURN TO START OF NEXT SUBTEST
4175     NEXT,       GOTO-PAGE(7),            !BUT TABLE IS ON PAGE 7
4176                 J/BUTIR11B              !GO DO "BUT" ON IR<11>H
4177 (5070) DCS(0.00.0.0.0.0) BM(0101..00.11..10.11..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.010)

4178
4179
4180
4181 ! - - - - -
4182
4183

```

```

4184 !* PART D *
4185 !TEST-030-D CHECKS THAT BUT(INSTRS) READS THE IR CORRECTLY
4186 !AS ROM ADDRESS=(452) ON THE INSTRS E78 ROM, AND RECEIVES THE DIAGNOSTIC VALUE
4187 !OF (05), TARGETING TO (405) AFTER THE DECODE
4188 5734:
4189 TEST0300:
4190     PO,          LOAD-ENUA(ZTARGET405),      !LOAD EXPECTED ADDRESS AFTER "BUT"
4191                LOAD-ERROR(TEST0300),      !ERROR DIRECTORY KEY
4192                DCS-CTR(C3.),              !COMPARE ENUA:TNUA IN 3. MICROWORDS
4193     NEXT,        J/GOBUTO300                !GO SETUP FOR "BUT"
(5734) DCS[1.00.1.0.0.0] BM[1100..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...000.111.001]

4194
4195
4196 5071: !(FREE)
4197 GOBUTO300:
4198     SETUP,      RETURN/SCOPE030,           !RETURN TO SCOPE LOOP TEST WORD
4199     NEXT,       GOTO-PAGE(7),              !BUT TABLE IS ON PAGE 7
4200     J/BUTINSTRS !GO DO "BUT" ON INSTRS<4:0>H
(5071) DCS[0.0.0.0.0.0] BM[0101..00.00..01.11..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.001]

4201
4202
4203 5072: !(FREE)
4204 SCOPE030:
4205     NEXT,       BUTD(SCOPE),               !NO ERROR: "TEST031A" [+1. WORD]
4206     J/TEST031A !ERROR: "LOAD030A" [-8. WORDS]
(5072) DCS[0.00.0.1.0.0] BM[0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.111.111]

4207
4208
4209
4210
4211
4212
4213 !-----
4214
4215 !*** TEST 031 ***
4216 !TEST-031 USES A DATA PATTERN OF: "0 001 000 010 100 010" (040212)
4217 !-----
4218
4219
4220 !* PART A *
4221 !TEST-031-A CHECKS THAT BUT(DMO#SMO#BYTE)
4222 !DM=IR<5:3>H="100", DMOH=0; SM=IR<11:9>H="000", SMOH=1; BYTE H=0
4223 5677:
4224 TEST031A:
4225     PO,          LOAD-ENUA(ZTARGET402),      !LOAD EXPECTED ADDRESS AFTER "BUT"
4226                LOAD-ERROR(TEST031A),      !ERROR DIRECTORY KEY
4227                DCS-CTR(C4.),              !COMPARE ENUA:TNUA IN 4. MICROWORDS
4228     NEXT,        J/LOAD031A                !GO LOAD PATTERN
(5677) DCS[1.00.1.0.0.0] BM[1011..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...110.111.010]

4229
4230
4231 5672:
4232 LOAD031A:

```

```

4233          FN,      BUMP-VERIFY,          !COUNT
4234          P2-U,    IR+EMIT              !LOAD IR WITH TEST PATTERN
4235          EMT/010242,          ! (010242)
4236          NEXT,    J/GOBUTO31A          !GO SETUP FOR "BUT"
(5672) DCS(0.00.0.0.0.0.1) BM(0001..00.00..00.10..100..010...0.0.0..0..0...1.1010...0..0000.0...11.000...000.111.011)

```

```

4237
4238
4239          5073:  !(FREE)
4240          GOBUTO31A:
4241          SETUP,   RETURN/TEST031B,      !RETURN TO START OF NEXT SUBTEST
4242          NEXT,    GOTO-PAGE(7),          !BUT TABLE IS ON PAGE 7
4243          J/BUTDMSM8YTE                    !GO DO "BUT"
(5073) DCS(0.00.0.0.0.0) BM(0101..00.11..01.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.011)

```

```

4244
4245
4246          ! - - - - -
4247

```

```

4248          !* PART 9 *
4249          !TEST-031-B CHECKS THAT BUT(INSTR1) READS THE IR CORRECTLY
4250          !AS CLASS-C=MOV*SMD*-DMO: IR<14:9>H="001 000"; IR<15>H="0";
4251          !DM=IR<5:3>H="100"; AND TARGETS TO (604)
4252          5675:

```

```

4253          TEST031B:
4254          PO,      LOAD-ENUA(ZTARGET604),  !LOAD EXPECTED ADDRESS AFTER "BUT"
4255          LOAD-ERROR(TEST031B),          !ERROR DIRECTORY KEY
4256          DCS-CTR(C3.),                  !COMPARE ENUA:TNUA IN 3. MICROWORDS
4257          NEXT,    J/GOBUTO31B          !GO SETUP FOR "BUT"
(5675) DCS(1.00.1.0.0.0) BM(1100..00.11..11.10..000..100...0.0.0..0..0...0.0000...0..0000.0...11.000...000.111.100)

```

```

4258
4259          5074:  !(FREE)
4260          GOBUTO31B:
4261          SETUP,   RETURN/SCOPE031,      !RETURN TO SCOPE LOOP TEST WORD
4262          NEXT,    GOTO-PAGE(7),          !BUT TABLE IS ON PAGE 7
4263          J/BUTINSTR1                    !GO DO INSTR1 "BUT"
(5074) DCS(0.00.0.0.0.0) BM(0101..00.00..01.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110)

```

```

4265
4266          5075:  !(FREE)
4267          SCOPE031:
4268          NEXT,    BUTD(SCOPE),          !NO ERROR: "TEST032A" [+1. WORDS]
4269          J/TEST032A                    ! ERROR: "LOAD031A" [-4. WORDS]
(5075) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.111.011)

```

```

4270
4271
4272
4273
4274
4275
4276
4277          ! - - - - -
4278
4279          !*** TEST 032 ***

```

```

4280 !TEST-032 USES A DATA PATTERN OF: "1 001 000 001 010 101" (110125)
4281 !
4282 ! - - - - -
4283 !
4284 !* PART A *
4285 !TEST-032-A CHECKS THAT BUT(MOV-DR7#IR<5:3>] READS THE (-FLTPT*MOV+FLTPT*DR7) AND
4286 !ALTERNATING PATTERN "010" IN IR<5:3>H CORRECTLY
4287 5673:
4288 TEST032A:
4289     PO,          LOAD-ENUA(ZTARGET412),          !LOAD EXPECTED ADDRESS AFTER "BUT"
4290                LOAD-ERROR(TEST032A),          !ERROR DIRECTORY KEY
4291                DCS-CTR(C4.),                  !COMPARE ENUA:TNUA IN 4. MICROWORDS
4292                BUMP-VERIFY,                   !COUNT
4293     NEXT         J/LOAD032A                    !GO LOAD PATTERN
(5673) DCS[1.00.1.0.0.0] BM[1011..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...110.110.110]

4294
4295 5666:
4296 LOAD032A:
4297     P2-U,       IR+EMIT,                       !LOAD IR WITH TEST PATTERN
4298                EMIT/110125,                   ! (110125)
4299     NEXT         J/GOBUT032A                    !GO SETUP FOR "BUT"
(5666) DCS[0.00.0.0.0.0] BM[1001..00.00..00.01..010..101...0.0.0..0..0...1.1010...0..0000.0...11.000...000.111.110]

4301
4302 5076: !(FREE)
4303 GOBUT032A:
4304     SETUP,     RETURN/TEST032B,                !RETURN TO START OF NEXT SUBTEST
4305     NEXT,      GOTO-PAGE(7),                  !BUT TABLE IS ON PAGE 7
4306                J/BUTMOVDR7IRS-3             !GO DO "BUT"
(5076) DCS[0.00.0.0.0.0] BM[0101..00.11..01.11..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.101]

4308
4309 ! - - - - -
4310 !
4311 !* PART B *
4312 !TEST-032-B CHECKS THAT BUT(INSTR1] READS THE IR CORRECTLY
4313 !AS CLASS-C=MOV#SMD#-DMD: IR<14:9>H="001 000"; IR<15>H="1";
4314 !DM=IR<5:3>H="010"; AND TARGETS TO (612)
4315 5671:
4316 TEST032B:
4317     PO,          LOAD-ENUA(ZTARGET612),          !LOAD EXPECTED ADDRESS AFTER "BUT"
4318                LOAD-ERROR(TEST032B),          !ERROR DIRECTORY KEY
4319                DCS-CTR(C3.),                  !COMPARE ENUA:TNUA IN 3. MICROWORDS
4320     NEXT         J/GOBUT032B                    !GO SETUP FOR "BUT"
(5671) DCS[1.00.1.0.0.0] BM[1100..00.11..11.10..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...000.111.111]

4322
4323 5077: !(FREE)
4324 GOBUT032B:
4325     SETUP,     RETURN/SCOPE032,                !RETURN TO SCOPE LOOP TEST WORD
4326     PO,        BUMP-VERIFY,                   !COUNT
4327     NEXT,      GOTO-PAGE(7),                  !BUT TABLE IS ON PAGE 7
4328

```

```

4329          T/BUTINSTR1          !GO DO INSTR1 "BUT"
(5077) DCS(0.00.0.0.0.1) BM(C:01..00.00..10.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110)
4330
4331
4332          5100: !(FREE)
4333          SCOPE032:
4334          NEXT, BUTD(SCOPE),          !NO ERROR: "TEST033A" [+1. WORDS]
4335          J/TEST033A          !ERROR: "LOAD032A" [-4. WORDS]
(5100) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.110.111)
4336
4337
4338
4339
4340
4341
4342          ! -----
4343
4344          !*** TEST 033 ***
4345          !TEST-033 USES A DATA PATTERN OF: "1 000 110 010 111 100" (106274)
4346
4347          ! -----
4348
4349          !* PART A *
4350          !TEST-033-A CHECKS THAT BUT(INSTR1) READS THE IR CORRECTLY
4351          !AS CLASS-B=SOP*-DMO; IR<14:08>H="000 110 0":
4352          !DM=IR<5:3>H="111", DMOH=0; AND TARGETS TO (517)
4353          5667:
4354          TEST033A:
4355          PO,          LOAD-ENUA(ZTARGETS17),          !LOAD EXPECTED ADDRESS AFTER "BUT"
4356          LOAD-ERROR(TEST033A),          !ERROR DIRECTORY KEY
4357          DCS-CTR(C4.),          !COMPARE ENUA:TNUA IN 4. MICROWORDS
4358          BUMP-VERIFY,          !COUNT
4359          NEXT, J/LOAD033A          !GO LOAD PATTERN
(5667) DCS(1.00.1.0.0.1) BM(1011..00.11.. 01..001..111...0.0.0..0..0...0.0000...0..0000.0...11.000...110.110.100)
4360
4361
4362          5664:
4363          LOAD033A:
4364          P2-U,          IR+EMIT,          !LOAD IR WITH TEST PATTERN
4365          EMIT/106274,          !(106274)
4366          NEXT, J/GOBUT033A          !GO SETUP FOR "BUT"
(5664) DCS(0.00.0.0.0.0) BM(1000..00.11..00.10..111..100...0.0.0..0..0...1.1010...0..0000.0...11.000...001.000.001)
4367
4368
4369          5101: !(FREE)
4370          GOBUT033A:
4371          SETUP,          RETURN/SCOPE033,          !RETURN TO SCOPE LOOP TEST WORD
4372          PO,          BUMP-VERIFY,          !COUNT
4373          NEXT,          GOTO-PAGE(7),          !BUT TABLE IS ON PAGE 7
4374          J/BUTINSTR1          !GO DO INSTR1 "BUT"
(5101) DCS(0.00.0.0.0.1) BM(0101..00.00..10.00..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110)
4375

```

```

4376
4377 5102: !(FREE)
4378 SCOPE033:
4379     NEXT, BUTD(SCOPE), !NO ERROR: "TEST034" [+1. WORD]
4380     J/TEST034A ! ERROR: "LOAD033A" [-2. WORDS]
(5102) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.110.101)

```

```

4381
4382
4383
4384
4385
4386
4387

```

```

4388 ! - - - - -
4389 !*** TEST 034 ***
4390 !TEST-034 USES A DATA PATTERN OF: "0 000 101 001 001 010" (005112)
4391 ! - - - - -

```

```

4392
4393 !* PART A *
4394 !TEST-034-A CHECKS THAT BUT(INSTR1) READS THE IR CORRECTLY
4395 !AS CLASS-B=SOP*-DMO; IR(14:09)H="000 101";
4396 !DM=IR(5:3)H="001", DMOH=0; AND TARGETS TO (511)
4397 5665:
4398 TEST034A:

```

```

4399     PO, LOAD-ENUA(ZTARGET511), !LOAD EXPECTED ADDRESS AFTER "BUT"
4400     LOAD-ERROR(TEST034A), !ERROR DIRECTORY KEY
4401     DCS-CTR(C4.), !COMPARE ENUA:TNUA IN 4. MICROWORDS
4402     NEXT, J/LOAD034A !GO LOAD PATTERN
(5665) DCS(1.00.1.0.0.0) BM(1011..00.11..11.01..001..001...0.0.0..0..0...0.0000...0..0000.0...11.000...110.110.010)

```

```

4403
4404 5662:
4405 LOAD034A:
4406     PO, BUMP-VERIFY, !COUNT
4407     P2-U, IR+EMIT, !LOAD IR WITH TEST PATTERN
4408     EMIT/005112, !(005112)
4409     NEXT, J/GOBUT034A !GO SETUP FOR "BUT"
(5662) DCS(0.00.0.0.0.1) BM(0000..00.10..10.01..001..010...0.0.0..0..0...1.1010...0..0000.0...11.000...001.000.011)

```

```

4410
4411 5103: !(FREE)
4412 GOBUT034A:
4413     SETUP, RETURN/SCOPE034, !RETURN TO SCOPE LOOP TEST WORD
4414     NEXT, GOTO-PAGE(7), !BUT TABLE IS ON PAGE 7
4415     J/BUTINSTR1 !GO DO INSTR1 "BUT"
(5103) DCS(0.00.0.0.0.0) BM(0101..00.00..10.00..100...11...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110)

```

```

4416
4417
4418 5104: !(FREE)
4419 SCOPE034:
4420     NEXT, BUTD(SCOPE), !NO ERROR: "TEST035A" [+1. WORD]
4421     J/TEST035A ! ERROR: "LOAD034A" [-2. WORDS]
(5104) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.110.011)
4422
4423

```

4424  
4425  
4426  
4427  
4428  
4429  
4430  
4431  
4432  
4433  
4434  
4435  
4436  
4437  
4438  
4439  
4440  
4441  
4442  
4443  
4444  
4445  
4446  
4447  
4448  
4449  
4450  
4451  
4452  
4453  
4454  
4455  
4456  
4457  
4458  
4459  
4460  
4461  
4462  
4463  
4464  
4465  
4466  
4467  
4468  
4469  
4470  
4471  
4472

!-----

!\*\*\* TEST 035 \*\*\*  
!TEST-035 USES A DATA PATTERN OF: "0 011 000 111 000 001" (030701)

!-----

!\* PART A \*  
!TEST-035-A CHECKS THAT BUT(INSTR1) READS THE IR CORRECTLY  
!AS CLASS-A=DOP\*SMD\*DMO; IR<15:12>H="0011";  
!SM=IR<11:9>H="000", SMOH=1; DM=IR<5:3>H="000", DMOH=1; AND TARGETS TO (443)

5663:

TEST035A:

PO, LOAD-ENUA(ZTARGET443), !LOAD EXPECTED ADDRESS AFTER "BUT"  
LOAD-ERROR(TEST035A), !ERROR DIRECTORY KEY  
DCS-CTR(C4.), !COMPARE ENUA:TNUA IN 4. MICROWORDS  
NEXT, J/LOAD035A !GO LOAD PATTERN

(5663) DCS(1.00.1.0.0.0) BM(1011.00.11.11.00.100.011...0.0.0.0.0.0...0.0000...G.0000.0...11.000...110.110.000)

5660:

LOAD035A:

P2-U, IR\*EMIT, !LOAD IR WITH TEST PATTERN  
EMIT/030701, ! (030701)  
NEXT, J/GOBUT035A !GC SETUP FOR "BUT"

(5660) DCS(0.00.0.0.0.0) BM(0011.00.00.01.11.000.001...0.0.0.0.0.0...1.1010...0.0000.0...11.000...001.000.101)

5105: !(FREE)

GOBUT035A:

SETUP, RETURN/SCOPE035, !RETURN TO SCOPE LOOP TEST WORD  
PO, BUMP-VERIFY, !COUNT  
NEXT, GOTO-PAGE(7), !BUT TABLE IS ON PAGE 7  
J/BUTINSTR1 !GO DO INSTR1 "BUT"

(5105) DCS(0.00.0.0.0.1) BM(0101.00.00.10.00.110.111...0.0.0.0.0.0...0.0000...0.0000.0...11.100...011.000.110)

5106: !(FREE)

SCOPE035:

NEXT, BUTD(SCOPE), !NO ERROR: "TEST036A" [+1. WORD]  
J/TEST036A ! ERROR: "LOAD035A" 1-2. WORDS

(5106) DCS(0.00.0.1.0.0) BM(0000.00.00.00.00.000.000...0.0.0.0.0.0...0.0000...0.0000.0...11.000...110.110.001)



```

4473
4474 :-----
4475
4476 !*** TEST 036 ***
4477 !TEST-036 USES A DATA PATTERN OF: "1 101 000 101 000 110" (150506)
4478
4479 :-----
4480
4481 !* PART A *
4482 !TEST-036-A CHECKS THAT BUT(INSTR1) READS THE IR CORRECTLY
4483 !AS CLASS-A=DOP*SMD*DMO: IR<15:12>H="1101":
4484 !SM=IR<11:9>H="000", SMDH=1; DM=IR<5:3>H="000", DM:DH=1; AND TARGETS TO (455)
4485 5661:
4486 TEST036A:
4487     PO,      LOAD-ENUA(ZTARGET455),      !LOAD EXPECTED ADDRESS AFTER "BUT"
4488             LOAD-ERROR(TEST036A),      !ERROR DIRECTORY KEY
4489             DCS-CTR(C4.),              !COMPARE ENUA:TNUA IN 4. MICROWORDS
4490     NEXT,    J/LOAD036A                 !GO LOAD PATTERN
(5661) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..101..101...0.0.0..0..0...0.0000...0..0000.0...11.000...110.101.110)
4491
4492
4493 5656:
4494 LOAD036A:
4495     PO,      BUMP-VERIFY,              !COUNT
4496     P2-U,    IR+EMIT,                  !LOAD IR WITH TEST PATTERN
4497             EMIT/150506,              ! (150506)
4498     NEXT,    J/GOBUT036A              !GO SETUP FOR "BUT"
(5656) DCS(0.00.0.0.0.1) BM(1101..00.00..01.01..000..110...0.0.0..0..0...1.1010...0..0000.0...11.000...001.000.111)
4499
4500
4501 5107: !(FREE)
4502 GOBUT036A:
4503     SETUP,   RETURN/SCOPE036,         !RETURN TO SCOPE LOOP TEST WORD
4504     NEXT,    GOTO-PAGE(7),           !BUT TABLE IS ON PAGE 7
4505             J/BUTINSTR1              !GO DO INSTR1 "BUT"
(5107) DCS(0.00.0.0.0.0) BM(0101..00.00..10.01..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110)
4506
4507
4508
4509 5110: !(FREE)
4510 SCOPE036:
4511     NEXT,    BUTD(SCOPE),             !NO ERROR: "TEST037A" [+1. WORD]
4512             J/TEST037A               ! ERROR: "LOAD036A" [-2. WORDS]
(5110) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.101.111)
4513
4514
4515
4516
4517
4518
4519 :-----
4520
4521 !*** TEST 037 ***

```

```

4522 !TEST-037 USES A DATA PATTERN OF: "0 110 000 010 000 101" (060205)
4523
4524 ! - - - - -
4525
4526 !* PART A *
4527 !TEST-037-A CHECKS THAT BUT(INSTR1) READS THE IR CORRECTLY
4528 !AS CLASS-A=DOP*SMO*DMO; IR<15:12>H="0110";
4529 !SM=IR<11:9>H="000", SMOH=1; DM=IR<5:3>H="000", DM=1; AND TARGETS TO (446)
4530 S657:
4531 TEST037A:
4532     PO,          LOAD-ENUA(ZTARGET446),          !LOAD EXPECTED ADDRESS AFTER "BUT"
4533     LOAD-ERROR(TEST037A),          !ERROR DIRECTORY KEY
4534     DCS-CTR(C4.),          !COMPARE ENUA:1129 IN 4. MICROWORDS
4535     NEXT,        J/LOAD037A          !GO LOAD PATTERN
(5657) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..100..110...0.0.0..0..0...0.0000...0..0000.0...11.000...110.101.100)

4536
4537
4538 S654:
4539 LOAD037A:
4540     P2-U,        IR+EMIT,          !LOAD IR WITH TEST PATTERN
4541     EMIT/060205,          ! (060205)
4542     NEXT,        J/GOBUT037A       !GO SETUP FOR "BUT"
(5654) DCS(0.00.0.0.0.0) BM(0110..00.00..00.10..000..101...0.0.0..0..0...1.1010...0..0000.0...11.000...001.001.001)

4543
4544
4545 S111: !(FREE)
4546 GOBUT037A:
4547     SETUP,      RETURN/SCOPE037,    !RETURN TO SCOPE LOOP TEST WORD
4548     PC,         BUMP-VERIFY,        !COUNT
4549     NEXT,       GOTO-PAGE(7),       !BUT TABLE IS ON PAGE 7
4550     J/BUTINSTR1 !GO DO INSTR1 "BUT"
(5111) DCS(0.00.0.0.0.1) BM(0101..00.00..10.01..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110)

4551
4552
4553 S112: !(FREE)
4554 SCOPE037:
4555     NEXT,       BUTD(SCOPE),        !NO ERROR: "TEST040A" [+1. WORD]
4556     J/TEST040A !ERROR: "LOAD037A" [-2. WORDS]
(5112) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.101.101)

4557
4558
4559
4560
4561
4562
4563 ! - - - - -
4564
4565 !*** TEST 040 ***
4566 !TEST-040 USES A DATA PATTERN OF: "0 101 000 111 111 111 " (050777)
4567 ! - - - - -
4568
4569
4570 !* PART A *

```

```

4571 !TEST-040-A CHECKS THAT BUT [IR<15:12>] READS THE
4572 !ALTERNATING PATTERN "0101" IN IR<15:12>H CORRECTLY
4573 5655:
4574 TEST040A:
4575     PO,          LOAD-ENUA(ZTARGET405),      !LOAD EXPECTED ADDRESS AFTER "BUT"
4576                LOAD-ERROR(TEST040A),      !ERROR DIRECTORY KEY
4577                DCS-CTR(C4.),              !COMPARE ENUA:TNUA IN 4. MICROWORDS
4578     NEXT,        J/LOAD040A                !GO LOAD PATTERN
(5655) DCS[1.00.1.0.0.0] BM[1011..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...110.101.000]

```

```

4579
4580 5650:
4581 LOAD040A:
4582     P2-U,        IR+EMIT,                  !LOAD IR WITH TEST PATTERN
4583                EMIT/OS0777,              ! (OS0777)
4584     NEXT,        J/GOBUT040A              !GO SETUP FOR "BUT"
(5650) DCS[0.00.0.0.0.0] BM[0101..00.00..01.11..111..111...0.0.0..0..0...1.1010...0..0000.0...11.000...001.001.011]

```

```

4586
4587 5113: !(FREE)
4588 GOBUT040A:
4589     SETUP,       RETURN/TEST040B,         !RETURN TO START OF NEXT SUBTEST
4590     NEXT,        GOTO-PAGE(7),            !BUT TABLE IS ON PAGE 7
4591                J/BUTIR15-12              !GO DO "BUT"
(5113) DCS[0.00.0.0.0.0] BM[0101..00.11..01.01..011..111...0.0.0..0..0...0.0000 ..0..0000.0...11.100...011.000.000]

```

! - - - - -

```

4593
4594 !* PART B *
4595 !TEST-040-B CHECKS THAT BUT[INSTR1] READS THE IR CORRECTLY
4596 !AS CLASS-B=DOP*-MOV*SMD*-DMO; DM=IR<5:3>H="111";
4597 !DOP=IR<14:12>H="101"; TARGETS TO (517)
4598 5653:
4599 TEST040B:
4600     PO,          LOAD-ENUA(ZTARGETS17),    !LOAD EXPECTED ADDRESS AFTER "BUT"
4601                LOAD-ERROR(TEST040B),    !ERROR DIRECTORY KEY
4602                DCS-CTR(C3.),            !COMPARE ENUA:TNUA IN 3. MICROWORDS
4603                BUMP-VERIFY,             !COUNT
4604     NEXT,        J/GOBUT040B              !GO SETUP FOR "BUT"
(5653) DCS[1.00.1.0.0.1] BM[1100..00.11..11.01..001..111...0.0.0..0..0...0.0000...0..0000.0...11.000...001.001.100]

```

```

4608
4609 5114: !(FREE)
4610 GOBUT040B:
4611     SETUP,       RETURN/SCOPE040,         !RETURN TO SCOPE LOOP TEST WORD
4612     NEXT,        GOTO-PAGE(7),            !BUT TABLE IS ON PAGE 7
4613                J/BUTINSTR1              !GO DO INSTR1 "BUT"
(5114) DCS[0.00.0.0.0.0] BM[0101..00.00..10.01..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110]

```

```

4615 5115: !(FREE)
4616
4617

```

4618 SCOPE040:  
 4619 NEXT, BUTD(SCOPE), !NO ERROR: "TEST041A" [+1. WORDS]  
 4620 J/TEST041A ! ERROR: "LOAD040A" [-4. WORDS]  
 (5115) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..0000..0000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.101.001)

4621  
 4622  
 4623  
 4624  
 4625  
 4626

4627 ! - - - - -  
 4628  
 4629 !\*\*\* TEST 041 \*\*\*  
 4630 !TEST-041 USES A DATA PATTERN OF: "1 010 000 111 111 111 " (120777)  
 4631 ! - - - - -

4632  
 4633  
 4634 !\* PART A \*  
 4635 !TEST-041-A CHECKS THAT BUT(IR<15:12>) READS THE  
 4636 !ALTERNATING PATTERN "1010" IN IR<15:12>H CORRECTLY

4637 5651:  
 4638 TEST041A:  
 4639 PO, LOAD-ENUA(ZTARGET412), !LOAD EXPECTED ADDRESS AFTER "BUT"  
 4640 LOAD-ERROR(TEST041A), !ERROR DIRECTORY KEY  
 4641 DCS-CTR(C4.), !COMPARE ENUA:TNUA IN 4. MICROWORDS  
 4642 NEXT, J/LOAD041A !GO LOAD PATTERN  
 (5651) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000. .110.100.100)

4643  
 4644  
 4645 5644:  
 4646 LOAD041A:  
 4647 PO, BUMP-VERIFY, !COUNT  
 4648 P2-U, IR+EMIT !LOAD IR WITH TEST PATTERN  
 4649 EMIT/120777, !(120777)  
 4650 NEXT, J/GOBUTO41A !GO SETUP FOR "BUT"

(5644) DCS(0.00.0.0.0.1) BM(1010..00.00..01.11..111..111...0.0.0..0..0...1.1010...0..0000.0...11.000...001.001.110)

4651  
 4652  
 4653 5116: !(FREE)  
 4654 GOBUTO41A:  
 4655 SETUP, RETURN/TEST041B, !RETURN TO START OF NEXT SUBTEST  
 4656 NEXT, GOTO-PAGE(7), !BUT TABLE IS ON PAGE 7  
 4657 J/BUTIR15-12 !GO DO "BUT"

(5116) DCS(0.00.0.0.0.0) BM(0101..00.11..01.00..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.000)

4658  
 4659  
 4660 ! - - - - -  
 4661  
 4662 !\* PART B \*  
 4663 !TEST041-B CHECKS THAT BUT(INSTR1) READS THE IR CORRECTLY  
 4664 !AS CLASS-B=DOP\*-MOV\*SMO\*-DMO; DM=IR<5:3>H="111";  
 4665 !DOP=IR<14:12>H="010"; TARGETS TO (517)  
 4666 5647:

```

4667 TEST041B:
4668 PO, LOAD-ENUA(ZTARGET517), !LOAD EXPECTED ADDRESS AFTER "BUT"
4669 LOAD-ERROR(TEST041B), !ERROR DIRECTORY KEY
4670 DCS-CTR(C3.), !COMPARE ENUA:TNUA IN 3. MICROWORDS
4671 BUMP-VERIFY, !COUNT
4672 NEXT, J/GOBUTO41B !GO SETUP FOR "BUT"
(5647) DCS(1.00.1.0.0.1) BM(1100..00.11.11.01..00!..111...0.0.0..0..0...0.0000...0..0000.0...11.000...001.001.111)

```

```

4673
4674
4675 5117: !(FREE)
4676 GOBUTO41B:
4677 SETUP, RETURN/SCOPE041, !RETURN TO SCOPE LOOP TEST WORD
4678 NEXT, GOTO-PAGE(7), !BUT TABLE IS ON PAGE 7
4679 J/BUTINSTR1 !GO DO INSTR1 "BUT"
(5117) DCS(0.00.0.0.0.0) BM(0101..00.00..10.10..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110)

```

```

4680
4681
4682 5120: !(FREE)
4683 SCOPE041:
4684 NEXT, BUTD(SCOPE), !NO ERROR: "TEST042A" [+1. WORDS]
4685 J/TEST042A !ERROR: "LOAD041A" [-4. WORDS]
(5120) DCS(0.00.1.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.100.101)

```

```

4686
4687
4688
4689
4690
4691
4692 ! - - - - -
4693
4694 !*** TEST 042 ***
4695 !TEST-042 USES A DATA PATTERN OF: "1 100 100 000 010 000" (144020)
4696
4697 ! - - - - -
4698

```

```

4699 !* PART A *
4700 !TEST-042-A CHECKS THAT BUT(DMO#SMO#BYTE)
4701 !DM=I<5:3>H="010", DMOH=0; SM=IR<11:9>H="100", SMOH=0; BYTE H=1
4702 5645:
4703 TEST042A:
4704 PO, LOAD-ENUA(ZTARGET401), !LOAD EXPECTED ADDRESS AFTER "BUT"
4705 LOAD-ERROR(TEST042A), !ERROR DIRECTORY KEY
4706 DCS-CTR(C4.), !COMPARE ENUA:TNUA IN 4. MICROWORDS
4707 NEXT, J/LOAD042A !GO LOAD PATTERN
(5645) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...110.100.000)

```

```

4708
4709
4710 5640:
4711 LOAD042A:
4712 P2-U, IR+EMIT, !LOAD IR WITH TEST PATTERN
4713 EMIT/144020, !(144020)
4714 NEXT, J/GOBUTO42A !GO SETUP FOR "BUT"
(5640) DCS(0.00.0.0.0.0) BM(1100..00.10..00.00..010..000...0.0.0..0..0...1.1010...0..0000.0...11.000...001.010.001)

```

```

4715
4716
4717 5121: !(FREE)
4718 GOBUTO42A:
4719     SETUP, RETURN/TEST042B, !RETURN TO START OF NEXT SUBTEST
4720     NEXT,   GOTO-PAGE(7),    !BUT TABLE IS ON PAGE 7
4721     J/BUTDMSMBYTE          !GO DO "BUT"
(5121) DCS(0.00.0.0.0.0) BM(0101..00.11..01.00..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.011)
4722
4723
4724 ! - - - - -
4725
4726 !* PART B *
4727 !TEST-042-B CHECKS THAT BUT(INSTR1) READS THE IR CORRECTLY
4728 !AS CLASS-G=DOP*-SMO; DOP=IR<14:12>H="100";
4729 !SM=IR<11:9>H="100"; TARGETS TO (714)
4730 5643:
4731 TEST042B:
4732     PO,     LOAD-ENUA(ZTARGET714), !LOAD EXPECTED ADDRESS AFTER "BUT"
4733     LOAD-ERROR(TEST042B),        !ERROR DIRECTORY KEY
4734     DCS-CTR(C3.),                !COMPARE ENUA:TNUA IN 3. MICROWORDS
4735     NEXT,   J/GOBUTO42B          !GO SETUP FOR "BUT"
(5643) DCS(1.00.1.0.0.0) BM(1100..00.11..11.11..001..100...0.0.0..0..0...0.0000...0..0000.0...11.000...001.010.010)
4736
4737
4738 5122: !(FREE)
4739 GOBUTO42B:
4740     SETUP, RETURN/SCOPE042,      !RETURN TO SCOPE LOOP TEST WORD
4741     PO,     BUMP-VERIFY,          !COUNT
4742     NEXT,   GOTO-PAGE(7),        !BUT TABLE IS ON PAGE 7
4743     J/BUTINSTR1                  !GO DO INSTR1 "BUT"
(5122) DCS(0.00.0.0.0.1) BM(0101..00.00..10.10..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110)
4744
4745
4746 5123: !(FREE)
4747 SCOPE042:
4748     NEXT,   BUTD(SCOPE),          !NO ERROR: "TEST043A" [+1. WORDS]
4749     J/TEST043A                    !ERROR: "LOAD042A" [-4. WORDS]
(5123) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.100.001)
4750
4751
4752
4753
4754
4755
4756 ! - - - - -
4757
4758 !*** TEST 043 ***
4759 !TEST-043 USES A DATA PATTERN OF: "1 110 001 111 000 111" (161707)
4760
4761 ! - - - - -
4762

```

```

4763  !* PART A *
4764  !TEST-043-A CHECKS THAT BUT(DMO#SMO#BYTE)
4765  !DM=IR<5:3>H="000", DMOH=1; SM=IR<11:9>H="001", SMOH=0; BYTE H=0
4766  5641:
4767  TEST043A:
4768      PO,          LOAD-ENUA(ZTARGET404),          !LOAD EXPECTED ADDRESS AFTER "BUT"
4769      LOAD-ERROR(TEST043A),          !ERROR DIRECTORY KEY
4770      DCS-CTR(C4.),          !COMPARE ENUA:TNUA IN 4. MICROWORDS
4771      NEXT,         J/LOAD043A          !GO LOAD PATTERN
(5641) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..000..100..0.0.0..0..0..0.0000...0..0000.0...11.000...110.011.100)

4772
4773
4774  5634:
4775  LOAD043A:
4776      PO,          BUMP-VERIFY,          !COUNT
4777      P2-U,        IR*EMIT          !LOAD IR WITH TEST PATTERN
4778      EMIT/161707,          ! (161707)
4779      NEXT,         J/GOBUTO43A          !GO SETUP FOR "BUT"
(5634) DCS(0.00.0.0.0.1) BM(1110..00.00..11.11..000..111...0.0.0..0..0...1.1010...0..0000.0...11.000...001.010.100)

4780
4781
4782  5124:  !(FREE)
4783  GOBUTO43A:
4784      SETUP,       RETURN/TEST0438,          !RETURN TO START OF NEXT SUBTEST
4785      NEXT,        GOTO-PAGE(7),          !BUT TABLE IS ON PAGE 7
4786      J/BUTDMSM8YTE          !GO DO "BUT"
(5124) DCS(0.00.0.0.0.0) BM(0101..00.11..00.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.011)

4787
4788
4789  ! - - - - -
4790
4791  !* PART B *
4792  !TEST-043-B CHECKS THAT BUT(INSTR1) READS THE IR CORRECTLY
4793  !AS CLASS-G=DOP*-SMD; DOP=IR<14:12>H="110";
4794  !SM=IR<11:9>H="001"; TARGETS TO (711)
4795  5637:
4796  TEST0438:
4797      PO,          LOAD-ENUA(ZTARGET711),          !LOAD EXPECTED ADDRESS AFTER "BUT"
4798      LOAD-ERROR(TEST0438),          !ERROR DIRECTORY KEY
4799      DCS-CTR(C3.),          !COMPARE ENUA:TNUA IN 3. MICROWORDS
4800      NEXT,         J/GOBUTO438          !GO SETUP FOR "BUT"
(5637) DCS(1.00.1.0.0.0) BM(1100..00.11..11.11..001..001...0.0.0..0..0...0.0000...0..0000.0...11.000...001.010.101)

4801
4802
4803  5125:  !(FREE)
4804  GOBUTO438:
4805      SETUP,       RETURN/SCOPE043,          !RETURN TO SCOPE LOOP TEST WORD
4806      NEXT,        GOTO-PAGE(7),          !BUT TABLE IS ON PAGE 7
4807      J/BUTINSTR1          !GO DO INSTR1 "BUT"
(5125) DCS(0.00.0.0.0.0) BM(0101..00.00..10.10..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110)

4808
4809

```

```

4810 5126: !(FREE)
4811 SCOPE043:
4812 PO, BUMP-VERIFY, !COUNT
4813 NEXT, BUTD(SCOPE), !NO ERROR: "TEST044A" [+1: WORDS]
4814 J/TEST044A !ERROR: "LOAD043A" [-4: WORDS]
(5126) DCS(0.00.0.1.0.1) BM(0000..00.00..00.00..000..000...0.0.0..0...0.0000...0..0000.0...11.000...110.011.101)

```

```

4815
4816
4817
4818
4819
4820
4821
4822
4823

```

```

4824 ! - - - - -
4825 !*** TEST 044 ***
4826 !TEST-044 USES A DATA PATTERN OF: "0 111 110 010 101 000" (076250)
4827 ! - - - - -

```

```

4828
4829 !* PART A *
4830 !TEST-044-A CHECKS THAT BUT(INSTR1) READS THE IR CORRECTLY
4831 !AS CLASS-A=XFC; IR<15:9>H="0 111 110"; AND TARGETS TO (447)
4832 5635:

```

```

4833 TEST044A:
4834 PO, LOAD-ENUA(ZTARGET447), !LOAD EXPECTED ADDRESS AFTER "BUT"
4835 LOAD-ERROR(TEST044A), !ERROR DIRECTORY KEY
4836 DCS-CTR(C4.), !COMPARE ENUA:TNUA IN 4. MICROWORDS
4837 NEXT, J/LOAD044A !GO LOAD PATTERN
(5635) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..100..111...0.0.0..0...0.0000...0..0000.0...11.000...110.011.010)

```

```

4838
4839
4840 5632:
4841 LOAD044A:
4842 P2-U, IR+EMIT, !LOAD IR WITH TEST PATTERN
4843 EMIT/076250, !(076250)
4844 NEXT, J/GOBUTO44A !GO SETUP FOR "BUT"
(5632) DCS(0.00.0.0.0.0) BM(0111..00.11..00.10..101..000...0.0.0..0...0...1.1010...0..0000.0...11.000...001.010.111)

```

```

4845
4846
4847 5127: !(FREE)
4848 GOBUTO44A:
4849 SETUP, RETURN/SCOPE044, !RETURN TO SCOPE LOOP TEST WORD
4850 NEXT, GOTO-PAGE(7), !BUT TABLE IS ON PAGE 7
4851 J/BUTINSTR1 !GO DO INSTR1 "BUT"
(5127) DCS(0.00.0.0.0.0) BM(0101..00.00..10.11..000..111...0.0.0..0...0...0.0000...0..0000.0...11.100...011.000.110)

```

```

4852
4853
4854 5130: !(FREE)
4855 SCOPE044:
4856 PO, BUMP-VERIFY, !COUNT
4857 NEXT, BUTD(SCOPE), !NO ERROR: "TEST045A" [+1. WORD]
4858 J/TEST045A !ERROR: "LOAD044A" [-2. WORDS]

```



(5130) DCS(0.00.0.1.0.1) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.011.011)

4859  
4860  
4861  
4862  
4863  
4864  
4865  
4866  
4867  
4868  
4869  
4870  
4871  
4872  
4873  
4874  
4875  
4876  
4877  
4878  
4879  
4880  
4881  
4882  
4883  
4884  
4885  
4886  
4887  
4888  
4889  
4890  
4891  
4892  
4893  
4894  
4895  
4896  
4897  
4898  
4899  
4900  
4901  
4902  
4903  
4904  
4905  
4906

!-----

!\*\*\* TEST 045 \*\*\*  
!TEST-045 USES A DATA PATTERN OF: "0 000 100 011 000 000" (004300)

!-----

!\* PART A \*  
!TEST-045-A CHECKS THAT BUT(DM#SM#BYTE) READS THE IR CORRECTLY AS:  
!DM H="000" DMOH=1; SM H="100" SMOH=0; BYTE H=1 (IR=SWAB, SORT OF)  
!CHECKS THAT SWAB INSTR ASSERTS BYTE H

5633:  
TEST045A:  
PO, LOAD-ENUA(ZTARGET405), !LOAD EXPECTED ADDRESS AFTER "BUT"  
LOAD-ERROR(TEST045A), !ERROR DIRECTORY KEY  
DCS-CTR(C4.), !COMPARE ENUA:TNUA IN 4. MICROWORDS  
NEXT, J/LOAD045A !GO LOAD PATTERN

(5633) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...110.011.000)

5630:  
LOAD045A:  
P2-U, IR+EMIT, !LOAD IR WITH TEST PATTERN  
EMIT/004300, ! (004300)  
NEXT, J/GOBUT045A !GO SETUP FOR "BUT"

(5630) DCS(0.00.0.0.0.0) BM(0000..00.10..00.11..000..000...0.0.0..0..0...1.1010...0..0000.0...11.000...001.011.001)

5131: !(FREE)  
GOBUT045A:  
SETUP, RETURN/SCOPE045, !RETURN TO SCOPE LOOP TEST WORD  
NEXT, GOTO-PAGE(7), !BUT TABLE IS ON PAGE 7  
J/BUTDMSM#BYTE !GO DO BUT ON DM#SM#BYTE

(5131) DCS(0.00.0.0.0.0) BM(0101..00.00..10.11..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.011)

5132: !(FREE)  
SCOPE045:  
NEXT, BUTD(SCOPE), !NO ERROR "TEST046A" [+1. WORD]  
J/TEST046A ! ERROR: "LOAD045A" [-2. WORDS]

(5132) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.011.001)

```

4907
4908 ! -----
4909
4910 !*** TEST 046 ***
4911 !TEST-046 USFS A DATA PATTERN OF: "0 000 000 010 000 000" (000200)
4912
4913 ! -----
4914
4915 !* PART A *
4916 !TEST-046-A CHECKS THAT BUT[INSTRS] READS THE IR CORRECTLY
4917 !AS ROM ADDRESS=(426) ON THE INSTRS E88 ROM, AND RECEIVES THE VALUE
4918 !OF (06), TARGETING TO (426) AFTER THE DECODE
4919 5631:
4920 TEST046A:
4921     PO,          LOAD-ENUA(ZTARGET426),      !LOAD EXPECTED ADDRESS AFTER "BUT"
4922                 LOAD-ERROR(TEST046A),      !ERROR DIRECTORY KEY
4923                 DCS-CTR(C4.),              !COMPARE ENUA:TNUA IN 4. MICROWORDS
4924     NEXT,        J/LOAD0046A                !GO LOAD PATTERN
(5631) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..010..110...0.0.0..0..0...0.0000...0..0000.0...11.000...101.110.000)
4925
4926
4927 5560:
4928 LOAD0046A:
4929     P2-U,        IR+EMIT,                    !LOAD IR WITH TEST PATTERN
4930                 EMIT/000200,                ! (000200)
4931     NEXT,        J/GOBUT046A                !GO SETUP FOR "BUT"
(5560) DCS(0.00.0.0.0.0) BM(0000..00.00..00.10..000..000...0.0.0..0..0...1.10!0...0..0000.0...11.000...001.011.011)
4932
4933
4934 5133: !(FREE)
4935 GOBUT046A:
4936     SETUP,      RETURN/TEST046B,           !RETURN TO START OF NEXT SUBTEST
4937     NEXT,        GOTO-PAGE(7),             !BUT TABLE IS ON PAGE 7
4938                 J/BUTINSTRS               !GO DO INSTRS "BUT"
(5133) DCS(0.00.0.0.0.0) BM(0101..00.11..11.11..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.001)
4939
4940
4941
4942
4943 ! -----
4944
4945 !* PART B *
4946 !TEST-046-B CHECKS THAT BUT[IR<11>#FLTPT<3:0>] READS THE "0" IN IR<11>H CORRECTLY,
4947 !AND THE FLTPT DECODE ROM GETS ADDRESS (040), WHICH IS A STST INSTR,
4948 !DATA SHOULD BE (01)
4949 5770:
4950 TEST046B:
4951     PO,          LOAD-ENUA(ZTARGET401),      !LOAD EXPECTED ADDRESS AFTER "BUT"
4952                 LOAD-ERROR(TEST046B),      !ERROR DIRECTORY KEY
4953                 DCS-CTR(C3.),              !COMPARE ENUA:TNUA IN 3. MICROWORDS
4954     NEXT,        J/GOBUT046B                !
(5770) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...001.011.100)
4955

```

```

4956
4957 5134: !(FREE)
4958 GOBUTO468:
4959     SETUP, RETURN/SCOPE046, !RETURN TO SCOPE LOOP TEST WORD
4960     NEXT,  GOTO-PAGE(7),    !BUT TABLE IS ON PAGE 7
4961     J/BUTIR11FLTPT3-0      !GO DO "BUT" ON IR<11>H#FLTPT<3:0>H
(5134) DCS[0.00.0.0.0.0] BM[0101..00.00..10.11..101..111...0.C.0..0..0...0.0000...0..0000.0...11.100...011.000.010]

```

```

4962
4963
4964
4965
4966 5135: !(FREE)
4967 SCOPE046:
4968     NEXT,  BUTD(SCOPE),      !NO ERROR: "TEST047A" [+1. WORD]
4969     J/TEST047A              ! ERROR: "LOAD046A" [-4. WORDS]
(5135) DCS[0.00.0.1.0.0] BM[0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...101.110.001]

```

```

4970
4971
4972
4973
4974
4975

```

```

4976 ! - - - - -
4977
4978 !*** TEST 047 ***
4979 !TEST-047 USES A DATA PATTERN OF: "1 010 010 101 110 110" (122566)
4980
4981 ! - - - - -

```

```

4982
4983 !* PART A *
4984 !TEST-047-A CHECKS THAT BUT[INSTRS] READS THE IR CORRECTLY
4985 !AS ROM ADDRESS=(325) ON THE INSTRS E78 ROM, AND RECEIVES THE DIAGNOSTIC VALUE
4986 !OF (12), TARGETING TO (412) AFTER THE DECODE
4987 5561:
4988 TEST047A:
4989     PO,      LOAD-ENUA(ZTARGET412), !LOAD EXPECTED ADDRESS AFTER "BUT"
4990             LOAD-ERROR(TEST047A), !ERROR DIRECTORY KEY
4991             DCS-CTR(C4.),         !COMPARE ENUA:TNUA IN 4. MICROWORDS
4992             BUMP-VERIFY,          !COUNT
4993     NEXT,    J/LOAD047A           !GO LOAD PATTERN
(5561) DCS[1.00.1.0.0.1] BM[1011..00.11..11.00..001..010...0.0 0..0..0...0.0000...0..0000.0...11.000...101.101.000]

```

```

4994
4995
4996 5550:
4997 LOAD047A:
4998     P2-U,    IR+EMIT,           !LOAD IR WITH TEST PATTERN
4999             EMIT/122566,        !(122566)
5000     NEXT,    J/GOBUTO47A       !GO SETUP FOR "BUT"
(5550) DCS[0.00.0.0.0.0] BM[1010..00.01..01.01..110..110...0.0.0..0..0...1.1010...0..0000.0...11.000...001.011.110]

```

```

5001
5002
5003 5136: !(FREE)
5004 GOBUTO47A:

```

```

5005      SETUP, RETURN/TEST047B,      !RETURN TO START OF NEXT SUBTEST
5006      NEXT,  GOTO-PAGE(7),          !BUT TABLE IS ON PAGE 7
5007      J/BUTINSTRS                    !GO DO INSTRS "BUT"

```

(5136) DCS[0.00.0.0.0.0] BM[0101..00.11..01.11..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.001]

5008  
5009  
5010  
5011

! - - - - -

```

5014      !* PART B *
5015      !TEST-047-B CHECKS THAT BUT[IR<11>#FLTPT<3:0>] READS THE "0" IN IR<11>H CORRECTLY,
5016      !AND THE FLTPT DECODE ROM GETS ADDRESS (534), WHICH IS A LOAD/MODE-6 INSTR,
5017      !DATA SHOULD BE (13)
5018      5670:
5019      TEST047B:

```

```

5020      PO,      LOAD-ENUA(ZTARGET413),      !LOAD EXPECTED ADDRESS AFTER "BUT"
5021      LOAD-ERROR(TEST047B),                !ERROR DIRECTORY KEY
5022      DCS-CTR(C3.),                          !COMPARE ENUA:TNUA IN 3. MICROWORDS
5023      NEXT,    J/GOBUTO47B

```

(5670) DCS[1.00.1.0.0.0] BM[1100..00.11..11.00..001..011...0.0.0..0..0...0.0000...0..0000.0...11.000...001.011.111]

5024  
5025

```

5026      5137:  !(FREE)
5027      GOBUTO47B:

```

```

5028      PO,      BUMP-VERIFY,                  !COUNT
5029      SETUP,   RETURN/TEST047C,            !RETURN TO START OF NEXT SUBTEST
5030      NEXT,    GOTO-PAGE(7),                !BUT TABLE IS ON PAGE 7
5031      J/BUTIR11FLTPT3-0                    !GO DO "BUT" ON IR<11>H#FLTPT<3:0>H

```

(5137) DCS[0.00.0.0.0.1] BM[0101..00.11..10.11..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.010]

5032  
5033

! - - - - -

```

5036      !* PART C *
5037      !TEST-047-C CHECKS THAT BUT[MOV-DR7#IR<5:3>] READS THE -(-FLTPT*MOV+FLTPT*DR7), IR<5:3>="110"
5038      5730:
5039      TEST047C:

```

```

5040      PO,      LOAD-ENUA(ZTARGET406),      !LOAD EXPECTED ADDRESS AFTER "BUT"
5041      LOAD-ERROR(TEST047C),                !ERROR DIRECTORY KEY
5042      DCS-CTR(C3.),                          !COMPARE ENUA:TNUA IN 3. MICROWORDS
5043      NEXT,    J/GOBUTO47C                    !GO SETUP FOR "BUT"

```

(5730) DCS[1.00.1.0.0.0] BM[1100..00.11..11.00..000..110...0.0.0..0..0...0.0000...0..0000.0...11.000...001.100.000]

5044  
5045

```

5046      5140:  !(FREE)
5047      GOBUTO47C:

```

```

5048      SETUP,   RETURN/SCOPE047,            !RETURN TO SCOPE LOOP TEST WORD
5049      NEXT,    GOTO-PAGE(7),                !BUT TABLE IS ON PAGE 7
5050      J/BUTMOVDR7IR5-3                    !GO DO "BUT" ON (MOV/DR7)#IR<5:3>H

```

(5140) DCS[0.00.0.0.0.0] BM[0101..00.00..11.00..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.101]

5051  
5052  
5053  
5054  
5055  
5056  
5057  
5058  
5059  
5060  
5061  
5062  
5063  
5064  
5065  
5066  
5067  
5068  
5069  
5070  
5071  
5072  
5073  
5074  
5075  
5076  
5077  
5078  
5079  
5080  
5081  
5082  
5083  
5084  
5085  
5086  
5087  
5088  
5089  
5090  
5091  
5092  
5093  
5094  
5095  
5096  
5097  
5098  
5099

5141: !(FREE)  
SCOPE047:

NEXT, BUTD(SCOPE),  
J/TEST050A

!NO ERROR: "TEST050A" [+1. WORD]  
!ERROR: "LOAD047A" [-6. WORDS]

(5141) DCS(0.00.0.1.0.0) BM(0000.00.00.00.00.000.000.0.0.0.0.0.0.0.0000.0.0.0000.0.0.11.000.0.101.101.001)

!-----

!\*\*\* TEST 050 \*\*\*

!TEST-050-A USES A DATA PATTERN OF: "1 111 010 101 000 000" (172500)  
!TEST-050-B USES A DATA PATTERN OF: "1 111 101 010 000 000" (175200)

!-----

!\* PART A \*

!TEST-050-A CHECKS THAT BUT(INSTRS) READS THE IR CORRECTLY  
!AS ROM ADDRESS=(725) ON THE INSTRS E78 ROM, AND RECEIVES THE DIAGNOSTIC VALUE  
!OF (05), TARGETING TO (405) AFTER THE DECODE

5551:

TEST050A:

PO, LOAD-ENUA(ZTARGET405),  
LOAD-ERROR(TEST050A),  
DCS-CTR(C4.),  
BUMP-VERIFY,  
NEXT, J/LOAD050A

!LOAD EXPECTED ADDRESS AFTER "BUT"  
!ERROR DIRECTORY KEY  
!COMPARE ENUA:TNUA IN 4. MICROWORDS  
!COUNT  
!GO LOAD PATTERN

(5551) DCS(1.00.1.0.0.1) BM(1011.00.11.11.00.000.101.0.0.0.0.0.0.0.0000.0.0.0000.0.0.11.000.0.101.101.010)

5552:

LOAD050A:

P2-U, IR+EMIT,  
EMIT/172500,  
NEXT, J/GOBUT050A

!LOAD IR WITH TEST PATTERN  
!(172500)  
!GO SETUP FOR "BUT"

(5552) DCS(0.00.0.0.0.0) BM(1111.00.01.01.01.000.000.0.0.0.0.0.0.1.1010.0.0.0000.0.0.11.000.0.001.100.010)

5142: !(FREE)  
GOBUT050A:

SETUP, RETURN/TEST050B,  
NEXT, GOTO-PAGE(7),  
J/BUTINSTRS

!RETURN TO START OF NEXT SUBTEST  
!BUT TABLE IS ON PAGE 7  
!GO DO INSTRS "BUT"

(5142) DCS(0.00.0.0.0.0) BM(0101.00.11.10.10.110.111.0.0.0.0.0.0.0.0000.0.0.0000.0.0.11.100.0.011.000.001)

```

S100
S101 ! - - - - -
S102
S103 !* PART B *
S104 !TEST-050-B CHECKS THAT BUT(INSTR-5) READS THE IR CORRECTLY
S105 !AS ROM ADDR=(752) ON THE INSTRS E78 ROM, AND RECEIVES THE DIAGNOSTIC VALUE
S106 !OF (12), TARGETING TO (412) AFTER THE DECODE
S107 5726:
S108 TEST050B:
S109     PO,      LOAD-ENUA(ZTARGET412),      !LOAD EXPECTED ADDRESS AFTER "BUT"
S110           LOAD-ERROR(TEST050B),        !ERROR DIRECTORY KEY
S111           DCS-CTR(C4.),                !COMPARE ENUA:TMUA IN 4. MICROWORDS
S112     NEXT,   J/LOAD050B
(5726) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..001..010...0.0.0..0..0...0.000...0..0000.0...11.000...001.100.011)

S113
S114
S115 5143: !(FREE)
S116 LOAD050B:
S117     P2-U,   IR+EMIT,                    !LOAD IR WITH TEST PATTERN
S118           EMIT/175200,                 ! (175200)
S119     NEXT,   J/GOBUT050B                !GO SETUP FOR "BUT"
(5143) DCS(0.00.0.0.0.0) BM(1111..00.10..10.10..000..000...0.0.0..0..0...1.1010...0..0000.0...11.000...001.100.100)

S120
S121 5144: !(FREE)
S122 GOBUT050B:
S123     SETUP, RETURN/SCOPE050,           !RETURN TO SCOPE LOOP TEST WORD
S124     NEXT,  GOTO-PAGE(7),              !BUT TABLE IS ON PAGE 7
S125           J/BUTINSTRS                 !DO INSTRS BUT
(5144) DCS(0.00.0.0.0.0) BM(0101..00.00..11.00..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.001)

S126
S127
S128 5145: !(FREE)
S129 SCOPE050:
S130     PO,    BUSDIN+EMIT-(!),           !RESET PROC UCON
S131           EN-CLK-IR(15-00),
S132     NEXT,  BUTD(SCOPE),                !NO ERROR: "TEST101A" [+1. WORD]
S133           J/TEST101A                   ! ERROR: "LOAD050A" [-5. WORDS]
(5145) DCS(0.00.0.1.0.0) BM(0000..00.00..00.01..000..100...0.0.0..0..0...1.1001...0..0000.0...11.000...101.101.011)

S134
S135
S136
S137 !.PAGE=====
S138
S139 .TOC * TEST101: D -> DBUF -> IR PATH
S140
S141
S142 !*****
S143 !*
S144 !* TESTS: 101 - 104                                UWORDS: 057 + 067
S145 !*
S146 !* FUNCTIONS:
S147 !*
S148 !* THESE TESTS VERIFY THE "EMIT -> CSP -> BBUS -> D -> DBUF -> IR" DATAPATH.

```

```

5149  !*
5150  !* TEST 101 FIRST VERIFIES THE "D -> DBUF -> IR" DATA PATH, INSURING THAT THE
5151  !* DBUF LATCH CAN BE WRITTEN WITH ZEROES, AND ENABLED ONTO BUSDIN, TO BE PUT
5152  !* INTO THE IR AND VERIFIED (VIA INSTRS DECODE, AS A HALT INSTRUCTION).
5153  !*
5154  !* TESTS 102-104 THEN GO ON TO FURTHER TEST THE FULL DATAPATH FROM EMIT TO
5155  !* IR, VIA THE EXTENDED ROUTE. THESE TESTS THEN VERIFY THE CSP WRITE, ADDRESSING AND
5156  !* DATAPATHS LOGIC.
5157  !*
5158  !*****
5159
5160  5553:
5161  TEST101A:
5162      PC,      LOAD-ENVA(ZTARGET+34),      !INSTRS E88 OUTPUT FOR IR=(000000)
5163      LOAD-ERROR(TEST101A),      !ERROR DIRECTORY KEY
5164      DCS-CTR(C7.),      !COMPARE AT TARGET
5165      NEXT,    J/LOAD0101A
5166  (5553) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...101.010.000)
5167
5168  5520:
5169  LOAD0101A:
5170      P2-T,    D+ZERO,      !PUT (000000) IN D
5171      NEXT,    J/GOTEST101A
5172  (5520) DCS(0.00.0.0.0.0) BM(0011..00.00..00.00..000..000...0.1.0..0..0...0.0000...0..0000.0...11.000...001.100.111)
5173
5174  5147:  !(FREE)
5175  GOTEST101A:
5176      SETUP,  RETURN/SCOPE101,      !GOT TO SUBR THAT:
5177      NEXT,    CALL(DINTOIR-5)      !D -> DBUF -> IR, THEN BUT(INSTR5)
5178  (5147) DCS(0.00.0.0.0.0) BM(0101..00.00..11.01..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)
5179
5180  5150:  !(FREE)
5181  SCOPE101:
5182      NEXT,    BUTD(SCOPE1),      !NO ERROR: "TEST102A" (+1. WORDS)
5183      J/TEST102A      ! ERROR: "LOADCSP101A" (-3. WORDS)
5184  (5150) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...101.010.001)
5185
5186  ! -----
5187
5188  .TOC * TEST102-104: TESTING CSP ADDRESS/READ/WRITE FUNCTIONS
5189
5190  !THE FOLLOWING SET OF FOUR TESTS VERIFIES THAT THE CSP, AND THE CSP ADDRESS FIELD "CSPADDR"
5191  !HAS NO STUCK ZERO BITS, AND THAT THE EMIT -> CSP -> D -> DBUF -> IR DATAPATH
5192  !IS FULLY FUNCTIONAL.
5193
5194  ! AFTER TEST 104B COMPLETES, THE CSP WILL LOOK AS FOLLOWS:
5195
5196  ! "BAS-CON"      "CSP-ADR"      CSP      INSTRS
5197  ! U(41:40)H      U(23:20)H      LOCT      -DATA-  TARGET

```

5198					
5199		1111	00	000000	E78/434
5200		1110	01	000152	E78/425
5201		1101	02	000125	E78/432
5202		1100	03	-----	
5203		1011	04	125200	E88/412
5204		1010	05	-----	
5205		1001	06	-----	
5206		1000	07	152500	E88/405
5207		0111	10	125200	E88/405
5208		0110	11	-----	
5209		0101	12	-----	
5210		0100	13	125200	E88/412
5211	11	0011	14	-----	
5212	10	0010	15	000125	E78/432
5213	01	0001	16	000152	E78/425
5214	00	0000	17	000000	E78/434

5221 . TEST 102A VERIFIES THAT CSPD(02) WAS WRITTEN WITH THE UNIQUE INSTRS PATTERN:  
5222 : (000125), E78 TARGET (432). LOOKING FOR CSP ADDRESS BIT<01> STUCK ONE/ZERO,  
5223 ! OR ERRORS IN DATAPATH FROM EMIT -> CSP -> ALU-B -> D -> DBUF -> IR.

```

5224 5521:
5225 TEST102A:
5226     PO,      LOAD-EMUA(ZTARGET432),      !INSTRS E78 JUTPUT
5227             LOAD-ERROR(TEST102A),      !ERROR DIRECTORY KEY
5228             DCS-CTR(C12.),              !COMPARE AT TARGET
5229     NEXT,     J/LOAD01-102A
(5521) DCS(1.00.1.0.0.0) BM(0011..00.11..11.00..011..010...0.0.0..0..0...0.0000...0..0000.0...11.000...101.011.000)

```

```

5230 5530:
5231 LOAD01-102A:
5232     P3,      CSPD(01)+EMIT, EMIT/000152,      !INSTRS DATA PATTERN:
5233     NEXT,     J/LOAD02-102A                  ! (000152)=E78(425)
(5530) DCS(0.00.0.0.0.0) BM(0000..10.00..00.01..101..010...0.0.0..0..0...0.1110...1..0000.0...11.000...001.101.001)

```

```

5235 5151: !(FREE)
5236 LOAD02-102A:
5237     P3,      CSPD(02)+EMIT, EMIT/000125,      !INSTRS DATA PATTERN:
5238     NEXT,     J/LOAD04-102A                  ! (000125)=E78(432)
(5151) DCS(0.00.0.0.0.0) BM(0000..10.00..00.01..010..101...0.0.0..0..0...0.1101...1..0000.0...11.000...001.101.010)

```

```

5240 5152: !(FREE)
5241 LOAD04-102A:
5242     P3,      CSPD(04)+EMIT, EMIT/125200,      !INSTRS DATA PATTERN:
5243     NEXT,     J/LOAD10-102A                  ! (125200)=E88(412)
(5152) DCS(0.00.0.0.0.0) BM(1010..10.10..10.10..000..000...0.0.0..0..0...0.1011...1..0000.0...11.000...001.101.011)

```

```

5245 5153: !(FREE)
5246 LOAD10-102A:
5247

```



```

5248      P3,      CSPD(10)+EMIT, EMIT/152500,      ! INSTRS DATA PATTERN:
5249      NEXT,     J/LOAD00-102A,                   ! (152500)=E88(405)
(5153) DCS(0.00.0.0.0.0) BM(1101..10.01..01.01..000..000...0.0.0..0..0...0.0111...1..0000.C...11.000...001.101.100)

5250      5154: !(FREE)
5251      LOAD00-102A:
5252      P3,      CSPD(00)+EMIT, EMIT/000000,      ! INSTRS DATA PATTERN:
5253      NEXT,     J/LOAD0102A,                     ! (000000)=E78(434)
5254      ! IF THIS DATA USED, CSP ADDRESSING ERROR
5255 (5154) DCS(0.00.0.0.0.0) BM(1000..10.00..00.00..000..000...0.0.0..0..0...0.1111...1..0000.0...11.000...001.101.101)

5256      5155: !(FREE)
5257      LOAD0102A:
5258      P2-T,     D+CSPD(D02), BSEL/B17,           ! GET CSP LOC VIA CSPADDR, BASCON FIELD "00"
5259      NEXT,     J/GOTEST102A
5260 (5155) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...0.1.0..0..0...0.1101...0..0000.0...11.000...001.101.110)

5261      5156: !(FREE)
5262      GOTEST102A:
5263      SETUP,    RETURN/TEST102B,                 ! GO TO SUBR WHICH:
5264      NEXT,     CALL(DINTOIR-5)                  ! D -> DBUF -> IR, THEN BUT(INSTR5)
5265 (5156) DCS(0.00.0.0.0.0) BM(0101..00.10..11.11..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

5266
5267
5268
5269
5270
5271      ! - - - - -
5272
5273      ! TEST 102B VERIFIES THAT CSPD(10) WAS WRITTEN WITH THE UNIQUE INSTRS PATTERN:
5274      ! (152500), E88 TARGET (405). LOOKING FOR CSP ADDRESS BIT<03> STUCK ONE/ZERO.
5275      ! OR ERRORS IN DATAPATH FROM EMIT -> CSP -> ALU-B -> D -> DBUF -> IR.
5276      5574:
5277      TEST102B:
5278      PD,       LOAD-ENVA(ZTARGET405),             ! INSTRS E88 OUTPUT
5279      LOAD-ERROR(TEST102B),                       ! ERROR DIRECTORY KEY
5280      DCS-CTR(C7.),                                ! COMPARE AT TARGET
5281      NEXT,     J/LOAD0102B
(5574) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...001.101.111)

5282      5157: !(FREE)
5283      LOAD0102B:
5284      P2-T,     D+CSPD(D10), BSEL/B17,           ! GET CSP LOC VIA CSPADDR, BASCON FIELD "00"
5285      NEXT,     J/GOTEST102B
5286 (5157) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...0.1.0..0..0...0.0111...0..0000.0...11.000...001.110.000)

5287      5160: !(FREE)
5288      GOTEST102B:
5289      SETUP,    RETURN/TEST102C,                 ! GO TO SUBR WHICH:
5290      NEXT,     CALL(DINTOIR-5)                  ! D -> DBUF -> IR, THEN BUT(INSTR5)
5291 (5160) DCS(0.00.0.0.0.0) BM(0101..00.10..11.10..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

5292

```

5293  
5294  
5295  
5296  
5297  
5298  
5299  
5300  
5301  
5302  
5303  
5304  
5305  
5306  
5307  
5308  
5309  
5310  
5311  
5312  
5313  
5314  
5315  
5316  
5317  
5318  
5319  
5320  
5321  
5322  
5323  
5324  
5325  
5326  
5327  
5328  
5329  
5330  
5331  
5332  
5333  
5334  
5335  
5336  
5337  
5338  
5339  
5340

! - - - - -

!TEST 102C VERIFIES THAT CSPD(04) WAS WRITTEN WITH THE UNIQUE INSTRS PATTERN:  
! (125200) E88 TARGET (412). LOOKING FOR CSP ADDRESS BIT<02> STUCK ONE/ZERO,  
! OR ERRORS IN DATAPATH FROM EMIT -> CSF -> ALU-B -> D -> DBUF -> IR.

5564:

TEST102C:

PO, LOAD-ENVA(ZTARGET412), !INSTRS E88 OUTPUT  
LOAD-ERROR(TEST102C), !ERROR DIRECTORY KEY  
DCS-CTR(C7.), !COMPARE AT TARGET  
BUMP-VERIFY, !COUNT  
NEXT J/LOADD102C

(5564) DCS[1.00.1.0.0.0] BM[1000..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.00C...001.110.001]

5161: !(FREE)

LOADD102C:

P2-T, D+CSPD(004), BSEL/B17, !GET CSP LOC VIA CSPADDR, BASCON FIELD "00"  
NEXT J/GOTEST102C

(5161) DCS[0.00.0.0.0.0] BM[1010..10.00..00.00..000..000...0.1.0..0..0...0.1011...0..0000.0...11.000...001.110.010]

5162: !(FREE)

GOTEST102C:

SETUP, RETURN/TEST102D, !GO TO SUBR WHICH:  
NEXT CALL[DINTOIR-5] ! D -> DBUF -> IR, THEN BUT(INSTRS)

(5162) DCS[0.00.0.0.0.0] BM[0101..00.10..11.01..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011]

! - - - - -

!TEST 102D VERIFIES THAT CSPD(01) WAS WRITTEN WITH THE UNIQUE INSTRS PATTERN:  
! (000152) E78 TARGET (425). LOOKING FOR CSP ADDRESS BIT<00> STUCK ONE/ZERO,  
! OR ERRORS IN DATAPATH FROM EMIT -> CSP -> ALU-B -> D -> DBUF -> IR.

5554:

TEST102D:

PO, LOAD-ENVA(ZTARGET425), !INSTRS E88 OUTPUT  
LOAD-ERROR(TEST102D), !ERROR DIRECTORY KEY  
DCS-CTR(C7.), !COMPARE AT TARGET  
NEXT J/LOADD102D

(5554) DCS[1.00.1.0.0.0] BM[1000..00.11..11.00..010..101...0.0.0..0..0...0.0000...0..0000.0...11.000...001.110.011]

5163: !(FREE)

LOADD102D:

P2-T, D+CSPD(001), BSEL/B17, !GET CSP LOC VIA CSPADDR, BASCON FIELD "00"  
NEXT J/GOTEST102D

(5163) DCS[0.00.0.0.0.0] BM[1010..10.00..00.00..000..000...0.1.0..0..0...0.1110...0..0000.0...11.000...001.110.100]

```

5341 5164: !(FREE)
5342 GOTEST1020:
5343     SETUP, RETURN/SCOPE102,           !GO TO SUBR WHICH:
5344     NEXT,  CALL[DINTOIR-5]           ! D -> DBUF -> IR, THEN BUT(INSTRS)
(5164) DCS[0.00.0.0.0.0] BM[0101..00.00..11.10..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011]

```

```

5345
5346
5347
5348
5349 5165: !(FREE)
5350 SCOPE102:
5351     NEXT,  BUTD[SCOPE],             !NO ERROR: "TEST103A" (+1. WORDS)
5352     J/TEST103A                     ! ERROR: "LOAD01-102A" (-16. WORDS)
(5165) DCS[0.00.0.1.0.0] BM[0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...101.011.001]

```

```

5353
5354
5355 ! -----
5356
5357
5358 !THE FOLLOWING SET OF FOUR TESTS VERIFIES THAT THE CSP, AND THE CSP ADDRESS FIELD "CSPADDR"
5359 !HAS NO STUCK ONE BITS, AND THAT THE EMIT -> CSP -> D -> DBUF -> IR DATAPATH
5360 !IS FULLY FUNCTIONAL.
5361
5362
5363 ! -----
5364
5365

```

```

5366 *TEST 103A VERIFIES THAT CSPD(13) WAS WRITTEN WITH THE UNIQUE INSTRS PATTERN:
5367 : (125200), EBB TARGET (412). LOOKING FOR CSP ADDRESS BIT<02> STUCK ONE/ZERO,
5368 : OR ERRORS IN DATAPATH FROM EMIT -> CSP -> ALU-B -> D -> DBUF -> IR.
5369 5531:
5370 TEST103A:
5371     PO,  LOAD-ENUA(ZTARGET412),      !INSTRS EBB OUTPUT
5372     P3,  LOAD-ERROR(TEST103A),      !ERROR DIRECTORY KEY
5373     NEXT, DCS-CTR(C12.),            !COMPARE AT TARGET
5374     J/LOAD16-103A
(5531) DCS[1.00.1.0.0.0] BM[0011..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...101.100.110]

```

```

5375
5376 5546:
5377 LOAD16-103A:
5378     PO,  BUMP-VERIFY,               !COUNT
5379     P3,  CSPD[16]*EMIT, EMIT/000152, !INSTRS DATA PATTERN:
5380     NEXT, J/LOAD15-103A             ! (000152)=E78(425)
(5546) DCS[0.00.0.0.0.1] BM[0000..10.00..00.01..101..010...0.0.0..0..0...0.0001...1..0000.0...11.000...001.110.110]

```

```

5381
5382 5166: !(FREE)
5383 LOAD15-103A:
5384     P3,  CSPD[15]*EMIT, EMIT/000125, !INSTRS DATA PATTERN:
5385     NEXT, J/LOAD13-103A             ! (000125)=E78(432)
(5166) DCS[0.00.0.0.0.0] BM[0000..10.00..00.01..010..101...0.0.0..0..0...0.0010...1..0000.0...11.000...001.110.111]

```

```

5386
5387 5167: !(FREE)
5388 LOAD13-103A:

```

```

5389      P3,      CSPD(13)+EMIT, EMIT/125200,      ! INSTRS DATA PATTERN:
5390      NEXT,     J/LOAD07-103A                    ! (125200)=E88(412)
(5167) DCS(0.00.0.0.0.0) BM(1010..10.10..10.10..000..000...0.0.0..0..0...0.0100...1..0000.0...11.000...001.111.000)

5391      5170:  !(FREE)
5392      LOAD07-103A:
5393      P3,      CSPD(07)+EMIT, EMIT/152500,      ! INSTRS DATA PATTERN:
5394      NEXT,     J/LOAD17-103A                    ! (152500)=E88(405)
5395      (5170) DCS(0.00.0.0.0.0) BM(1101..10.01..01.01..000..000...0.0.0..0..0...0.1000...1..0000.0...11.000...001.111.001)

5396      5171:  !(FREE)
5397      LOAD17-103A:
5398      P3,      CSPD(17)+EMIT, EMIT/000000,      ! INSTRS DATA PATTERN:
5399      NEXT,     J/LOADD103A                      ! (000000)=E78(434)
5400      (5171) DCS(0.00.0.0.0.0) BM(0000..10.00..00.00..000..000...0.0.0..0..0...0.0000...1..0000.0...11.000...001.111.010)
                    ! IF THIS DATA USED, CSP ADDRESSING ERROR

5402      5172:  !(FREE)
5403      LOADD103A:
5404      P2-T,     D+CSPD(D13), BSEL/B17,           ! GET CSP LOC VIA CSPADDR, BASCON FIELD "00"
5405      NEXT,     J/GOTEST103A
5406      (5172) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...0.1.0..0..0...0.0100...0..0000.0...11.000...001.111.011)

5407      5173:  !(FREE)
5408      GOTEST103A:
5409      SETUP,    RETURN/TEST1038,                ! GO TO SUBR WHICH:
5410      NEXT,     CALL(DINTOIR-5)                 ! D -> DBUF -> IR, THEN BUT(INSTR5)
5411      (5173) DCS(0.00.0.0.0.0) BM(0101..00.10..11.11..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

5412
5413
5414
5415
5416
5417      ! - - - - -
5418
5419      ! TEST 1038 VERIFIES THAT CSPD(15) WAS WRITTEN WITH THE UNIQUE INSTRS PATTERN:
5420      ! (000125), E78 TARGET (432).  LOOKING FOR CSP ADDRESS BIT(01) STUCK ONE/ZERO.
5421      ! OR ERRORS IN DATAPATH FROM EMIT -> CSP -> ALU-B -> D -> DBUF -> IR.
5422      5572:
5423      TEST1038:
5424      PO,      LOAD-ENUA(Z1TARGET432),           ! INSTRS E78 OUTPUT
5425      LOAD-ERROR(TEST1038),                     ! ERROR DIRECTORY KEY
5426      DCS-CTR(C7.),                             ! COMPARE AT TARGET
5427      NEXT,     J/LOADD1038
(5572) DCS(1.00.1.0.0.0) BM(1000 .00.11..11.00..011..010...0.0.0..0..0...0.0000...0..0000.0...11.000...001.111.100)

5428      5174:  !(FREE)
5429      LOADD1038:
5430      P2-T,     D+CSPD(D15), BSEL/B17,           ! GET CSP LOC VIA CSPADDR, BASCON FIELD "00"
5431      NEXT,     J/GOTEST1038
5432      (5174) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...0.1.0..0..0...0.0010...0..0000.0...11.000...001.111.101)

5433      5175:  !(FREE)
5434

```

```

5435 GOTEST1038:
5436     SETUP, RETURN/TEST103C,           !GO TO SUBR WHICH:
5437     NEXT,  CALL(DINTOIR-5)           ! D -> DBUF -> IR, THEN BUT(INSTRS)
(5175) DCS[0.00.0.0.0.0] BM[0101. 00.10..11.11..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011]

5438
5439
5440
5441
5442
5443 ! - - - - -
5444
5445 !TEST 103C VERIFIES THAT CSPD(16) WAS WRITTEN WITH THE UNIQUE INSTRS PATTERN:
5446 ! (000152), E78 TARGET (425).  LOOKING FOR CSP ADDRESS BIT<00> STUCK ONE/ZERO,
5447 ! OR ERRORS IN DATAPATH FROM EMIT -> CSP -> ALU-B -> D -> DBUF -> IR.
5448 5576:
5449 TEST103C:
5450     PD, LOAD-ENUA(ZTARGET425),         !INSTRS E78 OUTPUT
5451         LOAD-ERROR(TEST103C),         !ERROR DIRECTORY KEY
5452         DCS-CTR(C7.),                 !COMPARE AT TARGET
5453         BUMP-VERIFY,                  !COUNT
5454     NEXT, J/LOADD103C
(5576) DCS[1.00.1.0.0.1] BM[1000..00.11..11.00..010..101...0.0.0..0..0...0.0000...0..0000.0...11.000...001.111.110]

5455 5176: !(FREE)
5456 LOADD103C:
5457     P2-T, D+CSPD(D16), BSEL/B17,      !GET CSP LOC VIA CSPADDR, BASCON FIELD "00"
5458     NEXT, J/GOTEST103C
(5176) DCS[0.00.0.0.0.0] BM[1010..10.00..00.00..000..000...0.1.0..0..0...0.0001...0..0000.0...11.000...001.111.111]

5460 5.77: !(FREE)
5461 GOTEST103C:
5462     SETUP, RETURN/TEST103C,           !GO TO SUBR WHICH:
5463     NEXT,  CALL(DINTOIR-5)           ! D -> DBUF -> IR, THEN BUT(INSTRS)
(5177) DCS[0.00.0.0.0.0] BM[0101..00.11..01.11..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011]

5465
5466
5467
5468
5469
5470 ! - - - - -
5471
5472 !TEST 103D VERIFIES THAT CSPD(07) WAS WRITTEN WITH THE UNIQUE INSTRS PATTERN:
5473 ! (152500), E88 TARGET (405).  LOOKING FOR CSP ADDRESS BIT<03> STUCK ONE/ZERO,
5474 ! OR ERRORS IN DATAPATH FROM EMIT -> CSP -> ALU-B -> D -> DBUF -> IR.
5475 5674:
5476 TEST103D:
5477     PD, LOAD-ENUA(ZTARGET405),         !INSTRS E88 OUTPUT
5478         LOAD-ERROR(TEST103D),         !ERROR DIRECTORY KEY
5479         DCS-CTR(C7.),                 !COMPARE AT TARGET
5480     NEXT, J/LOADD103D
(5674) DCS[1.00.1.0.0.0] BM[1000..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...010.000.000]

5481

```

```

5482 5200: !(FREE)
5483 LOADD1030:
5484 P2-T, D+CSPD(007), BSEL/B17, !GET CSP LOC VIA CSPADDR, BASCON FIELD "00"
5485 NEXT, J/GOTEST1030
(5200) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000!...0.1.0..0..0...0.1000...0..0000.0...11.000...010.000.001)

```

```

5486 5201: !(FREE)
5487 GOTEST1030:
5488 SETUP, RETURN/SCOPE103, !GO TO SUBR WHICH:
5489 NEXT, CALL(DINTOIR-5) ! D -> DBUF -> IR, THEN BUT(INSTR5)
5490 (5201) DCS(0.00.0.0.0.0) BM(0101..00.01..00.00..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

```

```

5491 5202: !(FREE)
5492 SCOPE103:
5493
5494
5495 NEXT, BUTD(SCOPE), !NO ERROR: "TEST104A" (+1. WORDS)
5496 J/TEST104A ! ERROR: "LOAD16-103A" (-16. WORDS)
5497 (5202) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000!...0.0.0..0..0...0.0000...0..0000.0...11.000...101.100.111)
5498

```

```

5499 ! -----
5500
5501 ! THE FOLLOWING SET OF TWO TESTS VERIFIES THAT THE CSP, AND THE CSP ADDRESS FIELD "BASCON"
5502 ! HAS NO STUCK ZERO/ONE BITS, AND THAT THE EMIT -> CSP -> D ; DBUF -> IR DATAPATH
5503 ! IS FULLY FUNCTIONAL.
5504
5505 ! -----
5506
5507
5508
5509
5510

```

```

5511 !TEST 104A VERIFIES THAT CSPB(16) WAS WRITTEN WITH THE UNIQUE INSTR5 PATTERN:
5512 ! (000152), E78 TARGET (425). LOOKING FOR CSP ADDRESS 91T<00> STUCK ONE/ZERO,
5513 ! OR ERRORS IN DATAPATH FROM EMIT -> CSP -> ALU-B -> D -> DBUF -> IR.
5514 ! THIS TEST USES THE "BASCON" ADDRESS MODE FOR THE CSP.
5515 5547:
5516 TEST104A:
5517 PO, LOAD-ENVA(ZTARGET425), !INSTR5 E78 OUTPUT
5518 LOAD-ERROR(TEST104A), !ERROR DIRECTORY KEY
5519 DCS-CTR(L7.), !COMPARE AT TARGET
5520 NEXT, J/LOADD104A
(5547) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..010..101!...0.0.0..0..0...0.0000...0..0000.0...11.000...101.100.000)

```

```

5521 5540:
5522 LOADD104A:
5523 P2-T, D+CSPB(B16), CSPADDR/D17, !GET CSP LOC VIA BASCON, CSPADDR FIELD "0000"
5524 NEXT, J/GOTEST104A
(5540) DCS(0.00.0.0.0.0) BM(1010..11.01..00.00..000..000!...0.1.0.0..0...0.0000...0..0000.0...11.000...010.000.011)

```

```

5526 5203: !(FREE)
5527 GOTEST104A:
5528 SETUP, RETURN/TEST104B, !GO TO SUBR WHICH:
5529

```

```

5530      NEXT, CALL(DINTOIR-5)          ! D -> DBUF -> IR, THEN BUT(INSTR5)
(5203) DCS(0.00.0.0.0.0) BM(0101..00.11..01.00..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

```

```

5531
5532
5533
5534
5535

```

```

5536 ! - - - - -

```

```

5537
5538 !TEST 1048 VERIFIES THAT CSPB(15) WAS WRITTEN WITH THE UNIQUE INSTR5 PATTERN:
5539 ! (000125), E78 TARGET (432). LOOKING FOR CSP ADDRESS BIT(01) STUCK ONE/ZERO,
5540 ! OR ERRORS IN DATAPATH FROM EMIT -> CSP -> ALU-B -> D -> DBUF -> IR.
5541 !THIS TEST USES THE "BASCON" ADDRESS MODE FOR THE CSP.

```

```

5542 5646:
5543 TEST1048:
5544     PO,      LOAD-ENUA(ZTARGET432),      !INSTR5 E78 OUTPUT
5545             LOAD-ERROR(TEST1048),      !ERROR DIRECTORY KEY
5546             DCS-CTR(C7.),              !COMPARE AT TARGET
5547     NEXT,    J/LOADD1048
(5646) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..011..010...0.0.0..0..0...0.0000...0..0000.0...11.000...010.000.100)

```

```

5548
5549 5204: !(FREE)
5550 LOADD1048:
5551     P2-T,    D+CSPB(B15), CSPADDR/D17,   !GET CSP LOC VIA BASCON, CSPADDR FIELD "0000"
5552     NEXT,    J/GOTEST1048
(5204) DCS(0.00.0.0.0.0) BM(1010..11.10..00.00..000..000...0.1.0..0..0...0.0000...0..0000.0...11.000...010.000.101)

```

```

5553
5554 5205: !(FREE)
5555 GOTEST1048:
5556     SETUP,   RETURN/SCOPE104,           !GO TO SUBR WHICH:
5557     NEXT,    CALL(DINTOIR-5)           ! D -> DBUF -> IR, THEN BUT(INSTR5)
(5205) DCS(0.00.0.0.0.0) BM(0101..00.01..00.00..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

```

```

5558
5559
5560
5561
5562 5206: !(FREE)
5563 SCOPE104:
5564     PO,      BUSDIN+EMIT-[I],           !RESET PROC UCON
5565             EN-CLK-IR[15-00],
5566     NEXT,    BUTD(SCOPE),               !NO ERROR: "TEST105A" (+1. WORDS)
5567             J/TEST105A                  ! ERROR: "LOADD104A" (-5. WORDS)
(5206) DCS(0.00.0.1.0.0) BM(0000..00.00..00.01..000..100...0.0.0..0..0...1.1001...0..0000.0...11.000...101.100.001)

```

```

5568
5569
5570

```

```

5571 !.PAGE=====

```

```

5572 .TOC * TEST105: SR CAN LOAD/STORE AS A REGISTER

```

```

5573
5574
5575 !*****
5576 !*

```

```

5577  !*   TESTS: 105                               UWORDS: 023 + 032
5578  !*
5579  !*   FUNCTIONS:
5580  !*
5581  !*   THE FOLLOWING TESTS VERIFY THE VALIDITY OF THE SR AS A TEMPORARY REGISTER,
5582  !*   (IE, IT CAN BE LOADED/READ) IN ALL BIT POSITIONS, AND THAT THE ALU-A SIDE CAN
5583  !*   PASS DATA INTO D.
5584  !*
5585  !*****
5586
5587
5588  !TEST 105A VERIFIES THAT THE SR CAN BE LOADED/READ WITH THE INSTRS PATTERN:
5589  ! (000125), E78 TARGET (432), AND TO VERIFY THE:
5590  ! DATAPATH FROM SR -> ALU-A -> D -> DBUF -> IR.
5591  5541:
5592  TEST105A:
5593      PO,      LOAD-ENUA(ZTARGET432),          !INSTRS E78 OUTPLT
5594              LOAD-ERROR(TEST105A),          !ERROR DIRECTORY KEY
5595              DCS-CTR(C8.),                  !COMPARE AT TARGET
5596              BUMP-VERIFY,                   !COUNT
5597      NEXT     J/LOADSR105A
(5541) DCS[1.00.1.0.0.1] BM[0111..00.11..11.00..011..010...0.0.0..0..0...0.0000...0..0000.0...11.000...101.100.010]

5598
5599  5542:
5600  LOADSR105A:
5601      P2,      RES+CSPD(002),                  !BITS<13:11>=00/0, WHICH IS: SR/LOAD, GUARD/DISABLED
5602      P3-T,    SR+CSPD(002), BSEL/B17.        !DATA IS (000125) = INSTRS E78 (432)
5603      NEXT     J/GOTEST105A
(5542) DCS[0.00.0.0.0.0] BM[1010..10.00..00.00..000..000...1.0.1..0..0...0.1101...0..1000.1...11.000...010.000.111]

5604
5605  5207: !(FREE)
5606  GOTEST105A:
5607      PO,      BUMP-VERIFY,                   !COUNT
5608      SETUP,   RETURN/TEST105A1,             !GO TO SUBR WHICH:
5609      NEXT     CALL[SRINTOIR-5]              ! SR -> D -> DBUF -> IR, THEN BUT(INSTRS)
(5207) DCS[0.00.0.0.0.1] BM[0101..00.11..00.01..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.010]

5610
5611
5612
5613
5614
5615  ! - - - - -
5616
5617  !TEST 105A1 VERIFIES THAT THE BUT(SR3-0) SEES THE "0101" IN THE SR.
5618  5610:
5619  TEST105A1:
5620      PO,      LOAD-ENUA(ZTARGET405),          !BIT<3:0> = "0101"
5621              LOAD-ERROR(TEST105A1),          !ERROR DIRECTORY KEY
5622              DCS-CTR(C3.),                  !COMPARE AT TARGET
5623      NEXT     J/GOBUT105A1
(5610) DCS[1.00.1.0.0.0] BM[1100..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...010.001.000]

5624
5625  5210: !(FREE)

```



```

5626 GOBUT105A1:
5627     PO      BUMP-VERIFY           !COUNT
5628     SETUP,  RETURN/TEST105B,      !RETURN TO START OF NEXT SUBTEST
5629     NEXT,   GOTO-PAGE(7),         !BUT TABLE
5630     J/BUTSR3-0                     !SR<3:0> IN BIT<3:0>
(5210) DCS(0.00.0.0.0.1) BM(0101..00.11..01.01..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.110)

5631
5632
5633
5634
5635
5636 ! - - - - -
5637
5638 !TEST 105B VERIFIES THAT THE SR CAN BE LOADED/READ WITH THE INSTRS PATTERN:
5639 ! (000152), E78 TARGET (425), AND TO VERIFY THE:
5640 ! DATAPATH FROM SR -> ALU-A -> D -> DBUF -> IR.
5641 5652:
5642 TEST105B:
5643     PO,      LOAD-ENUA(ZTARGET425), !INSTRS E78 OUTPUT
5644     LOAD-ERROR(TEST105B),          !ERROR DIRECTORY KEY
5645     DCS-CTR(C8.),                  !COMPARE AT TARGET
5646     NEXT,    J/LOADSR105B
(5652) DCS(1.00.1.0.0.0) BM(0111..00.11..11.00..010..101...0.0.0..0..0...0.0000...0..0000.0...11.000...010.001.001)

5647
5648 5211: !(FREE)
5649 LOADSR105B:
5650     P2-T,   SR+CSPD(D01), BSEL/B17, !DATA IS (000152) = INSTRS E78 (425)
5651     NEXT,   J/GOTEST105B
(5211) DCS(0.00.0.0.0.0) BM(1010..10.00..0C.00..000..000...0.0.1..0..0...0.1110...0..0000.0...11.000...010.001.010)

5652
5653 5212: !(FREE)
5654 GOTEST105B:
5655     SETUP,  RETURN/TEST105B1,      !GO TO SUBR WHICH:
5656     NEXT,   CALL(SRINTOIR-5)       ! SR -> D -> DBUF -> IR, THEN BUT(INSTRS)
(5212) DCS(0.00.0.0.0.0) BM(0101..00.11..00.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.010)

5657
5658
5659
5660
5661 ! - - - - -
5662
5663
5664 !TEST 105B1 VERIFIES THAT THE BUT(SR3-0) SEES THE "1010" IN THE SR.
5665 5600:
5666 TEST105B1:
5667     PO,      LOAD-ENUA(ZTARGET412), !BIT<3:0> = "1010"
5668     LOAD-ERROR(TEST105B1),          !ERROR DIRECTORY KEY
5669     DCS-CTR(C3.),                  !COMPARE AT TARGET
5670     BUMP-VERIFY,                   !COUNT
5671     NEXT,    J/GOBUT105B1
(5600) DCS(1.00.1.0.0.1) BM(1100..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...010.001.011)

5672

```

```

5673 5213: !(FREE)
5674 GOBUT105A1:
5675     SETUP, RETURN/TEST105C,           !RETURN TO START OF NEXT SUBTEST
5676     NEXT,  GOTO-PAGE(7),             !BUT TABLE
5677     J/BUTSR3-0                       !SR<3:0> IN BIT<3:0>
(5213) DCS(0.00.0.0.0.0) BM(0101..00.11..01.00 .010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.1101

```

5678  
5679  
5680  
5681  
5682  
5683  
5684  
5685  
5686  
5687  
5688  
5689  
5690  
5691  
5692  
5693  
5694

! - - - - -

!TEST 105C VERIFIES THAT THE SR CAN BE LOADED/READ WITH THE INSTRS PATTERN:  
! (125200), E88 TARGET (412), AND TO VERIFY THE:  
! DATAPATH FROM SR -> ALU-A -> D -> DBUF -> IR.

```

5642:
5690 TEST105C:
5691     PD,      LOAD-ENUA(ZTARGET412),      !INSTRS E88 OUTPUT
5692     LOAD-ERROR(TEST105C),              !ERROR DIRECTORY KEY
5693     DCS-CTR(CB.),                      !COMPARE AT TARGET
5694     NEXT,    J/LOADSR105C
(5642) DCS(1.00.1.0.0.0) BM(0111..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...010.001.1001

```

5695  
5696  
5697  
5698  
5699

5214: !(FREE)  
LOADSR105C:  
P2-I, SR+CSPD(D04), BSEL/817, !DATA IS (125200) = INSTRS E88 (412)  
NEXT, J/GOTEST105C

```

(5214) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...0.0.1..0..0...0.1011...0..0000.0...11.000...010.001.1011

```

5700  
5701  
5702  
5703  
5704

5215: !(FREE)  
GOTEST105D:  
SETUP, RETURN/TEST105D, !GO TO SUBR WHICH:  
NEXT, CALL[SRINTOIR-5] ! SR -> D -> DBUF -> IR, THEN BUT(INSTRS)

```

(5215) DCS(0.00.0.0.0.0) BM(0101..00.11..00.11..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.0101

```

5705  
5706  
5707  
5708  
5709

! - - - - -

!TEST 105D VERIFIES THAT THE SR CAN BE LOADED/READ WITH THE INSTRS PATTERN:  
! (152500), E88 TARGET (405), AND TO VERIFY THE:  
! DATAPATH FROM SR -> ALU-A -> D -> DBUF -> IR.

```

5636:
5716 TEST105D:
5717     PD,      LOAD-ENUA(ZTARGET405),      !INSTRS E88 OUTPUT
5718     LOAD-ERROR(TEST105D),              !ERROR DIRECTORY KEY
5719     DCS-CTR(CB.),                      !COMPARE AT TARGET
5720     NEXT,    J/LOADSR105D

```

```

5721 (5636) DCS(1.00.1.0.0.0) BM(0111..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...010.001.110)
5722 5216: !(FREE)
5723 LOADSR1050:
5724 PO, BUMP-VERIFY, !COUNT
5725 P2-T, SR=CSPD(010), BSEL/B17, !DATA IS (152500) = INSTR5 E88 (405)
5726 NEXT, J/GOTEST1050
(5216) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...0.0.1..0..0...0.0111...0..0000.0...11.000...010.001.111)
5727 5217: !(FREE)
5728 GOTEST1050:
5729 SETUP, RETURN/TEST105E, !GO TO SUBR WHICH:
5730 NEXT, CALL(SRINTOIR-5) !SR -> D -> DBUF -> IR, THEN BUT(INSTR5)
(5217) DCS(0.00.0.0.0.0) BM(0101..00.11..00.10..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.010)
5732
5733
5734
5735
5736
5737 ! - - - - -
5738
5739 !TEST 105E VERIFIES THE DATAPATH FROM:
5740 ! EMIT -> CSP -> ALU-B -> SR -> ALU-A -> D -> DBUF -> IR,
5741 ! IS A VALID "1101" IN BITS<15:12>. SR VALUE = (152500) FROM PREVIOUS TEST.
5742 5620:
5743 TEST105E:
5744 PO, LOAD-ENUA(ZTARGET415), !BIT<15:12> = "1101"
5745 LOAD-ERROR(TEST105E), !ERROR DIRECTORY KEY
5746 DCS-CTR(C3.), !COMPARE AT TARGET
5747 NEXT, J/GOTEST105E
(5620) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..001..101...0.0.0..0..0...0.0000...0..0000.0...11.000...010.010.000)
5749 5220: !(FREE)
5750 GOTEST105E:
5751 SETUP, RETURN/SCOPE105, !RETURN TO SCOPE LOOP TEST WORD
5752 NEXT, GOTO-PAGE(7), !BUT TABLE
5753 J/BUTIR15-12 !BIT<15:12> TEST
(5220) DCS(0.00.0.0.0.0) BM(0101..00.01..00.10..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.000)
5754
5755
5756
5757 5221: !(FREE)
5758 SCOPE105:
5759 PO, BUSDIN+EMIT-[I], !RESET PROC UCON
5760 EN-CLK-IR[15-00],
5761 NEXT, BUTD[SCOPE], !NO ERROR: "SETUPCSP17A" (+1. WORDS)
5762 J/SETUPCSP17A !ERROR: "LOADSR105A" (-17. WORDS)
(5221) DCS(0.00.0.1.0.0) BM(0000..00.00..00.01..000..100...0.0.0..0..0...1.1001...0..0000.0...11.000...101.100.011)
5763
5764
5765
5766

```

5767  
5768  
5769  
5770  
5771  
5772  
5773  
5774  
5775  
5776  
5777  
5778  
5779  
5780  
5781  
5782  
5783  
5784  
5785  
5786  
5787  
5788  
5789  
5790  
5791  
5792  
5793  
5794  
5795  
5796  
5797  
5798  
5799  
5800  
5801  
5802  
5803  
5804  
5805  
5806  
5807  
5808  
5809  
5810  
5811  
5812  
5813  
5814  
5815  
5816  
5817  
5818  
5819  
5820

! .PAGE=====

.TOC \* TEST114-121: ALU LOGIC TESTS / D(C) TESTS

\*\*\*\*\*  
\*  
\* TESTS: 114 - 121 UWORDS: 250 + 300  
\*  
\* FUNCTIONS:  
\* THESE TESTS TEST THE ALU LOGIC FUNCTIONS.  
\*  
\*\*\*\*\*

SUMMARY OF ALU LOGIC / D(C) TESTS:

TEST NUMB	ALU FUNCTION	OPERANDS A/B=0	D(C) FUNCTION
114A	ZERO	1/1=0	CIN=PS(C)=0
115A	NOT-A	1/1=0	CIN=1
115B	NOT-A	0/1=1	PS(C)=0
115C	NOT-A	0/0=1	
115D	NOT-A	1/0=0	
116A	NOT-A-AND-B	0/1=1	ALU15=1, D(C)=1
116B	ZERO	1/0=0	
116C	NOT-A-AND-B	0/0=0	ALU15=0, D(C)=0
116D	ZERO	0/1=0	
117A	A-AND-NOT-B	1/0=1	CIN=D(C)=0
117B	A-AND-NOT-B	1/1=0	(ALU15=1)
117C	A-AND-NOT-B	0/0=0	CIN=D(C)=1
120A	A-AND-B	0/1=0	
120B	A-AND-B	1/1=1	CIN=0
121A	A-XOR-B	0/0=0	ALU00=1
121B	A-XOR-B	1/0=1	ALU00=0
121C	A-XOR-B	0/1=1	ALU07=1
121D	A-XOR-B	1/1=0	ALU07=0
122A	A-IOR-B	0/0=0	CIN=1
122A3	BUT(D<14-00>=ZERO#D15) W/ D=(000000)		
122A4	BUT(D<14-00>=ZERO#D15) W/ D=(125252)		

```

5821 !
5822
5823 !FOR THE ALU LOGIC TESTS FOLLOWING, THE REQUIRED CONSTANTS
5824 !IN THE CSP ARE:
5825 5543:
5826 SETUPCSP17A:
5827     P3,    CSPD[15]+EMIT, EMIT/000077,    !MASK FOR BITS<05:00>
5828     NEXT,  GOTO-PAGE(7),                !XFR
5829     J/SETUPCSP16A
(5543) DCS[0.00.0.0.0.0] BM[0000..10.00..00.00..111..111...0.0.0..0..0...0.0010...1..0000.0...11.100...000.000.010]

5830
5831 7002: !(FREE)
5832 SETUPCSP16A:
5833     P3,    CSPD[16]+EMIT, EMIT/170000,    !BITS<15:12> SET
5834     NEXT,  J/SETUPCSP15A
(7002) DCS[0.00.0.0.0.0] BM[1111..10.00..00.00..000..000...0.0.0..0..0...0.0001...1..0000.0...11.000...000.001.000]

5835
5836 7010: !(FREE)
5837 SETUPCSP15A:
5838     P3,    CSPD[17]+EMIT, EMIT/007700,    !MASK FOR BITS<11:06>
5839     NEXT,  J/SETUPCSP14A
(7010) DCS[0.00.0.0.0.0] BM[0000..10.11..11.11..000..000...0.0.0..0..0...0.0000...1..0000.0...11.000...000.001.001]

5840
5841 7011: !(FREE)
5842 SETUPCSP14A:
5843     P3,    CSPD[14]+EMIT, EMIT/000100,    !BIT<06> SET
5844     NEXT,  J/SETUPCSP12A
(7011) DCS[0.00.0.0.0.0] BM[0000..10.00..00.01..000..000...0.0.0..0..0...0.0011...1..0000.0...11.000...000.001.010]

5845
5846 7012: !(FREE)
5847 SETUPCSP12A:
5848     P3,    CSPD[11]+EMIT, EMIT/125252,    !PATTERN: "1010 1010 1010 1010"
5849     NEXT,  J/SETUPCSP05A
(7012) DCS[0.00.0.0.0.0] BM[1010..10.10..10.10..101..010...0.0.0..0..0...0.0110...1..0000.0...11.000...000.001.011]

5850
5851 7013: !(FREE)
5852 SETUPCSP05A:
5853     P3,    CSPD[10]+EMIT, EMIT/052525,    !PATTERN: "0101 0101 0101 0101"
5854     NEXT,  J/SETUPCSP07A
(7013) DCS[0.00.0.0.0.0] BM[0101..10.01..01.01..010..101...0.0.0..0..0...0.0111...1..0000.0...11.000...000.001.100]

5855
5856 7014: !(FREE)
5857 SETUPCSP07A:
5858     P3,    CSPD[12]+EMIT, EMIT/177777,    !PATTERN: "1111 1111 1111 1111 "
5859     NEXT,  J/SETUPCSP00A
(7014) DCS[0.00.0.0.0.0] BM[1111..10.11..11.11..111..111...0.0.0..0..0...0.0101...1..0000.0...11.000...000.001.101]

5860
5861 7015: !(FREE)
5862 SETUPCSP00A:
5863     P3,    CSPD[13]+EMIT, EMIT/000000,    !PATTERN: "0000 0000 0000 0000"
5864     NEXT,  GOTO-PAGE(0),                !SAME AS (4)
5865     J/TEST114A

```

(7015) DCS(0.00.0.0.0.0) BM(000C..10.00..00.00..J00..000...0.0.0..0..0...0.0100...1..0000.0...11.100...110.000.111)

5866  
5867  
5868  
5869  
5870  
5871  
5872  
5873  
5874  
5875  
5876  
5877  
5878

! - - - - -  
!VERIFY THAT WITH: ALU=(ZERO), A=(177777), B=(177777), THEN D=(000000), AND D[C]=CIN=PS[C]=(0)

4607:  
TEST114A:  
PO, LOAD-ENVA(ZTARGET434), !INSTRS FOR IR=(000000)  
LOAD-ERROR(TEST114A), !ERROR DIRECTORY KEY  
DCS-CTR(C8.) !COMPARE AT TARGET  
J/GETONES114A

(4607) DCS(1.00.1.0.0.0) BM(0111..00.11..11.00..011..100...0.0.0..0..0...0.0C00...0..0000.0...11.000...110.000.000)

5879  
5880  
5881  
5882  
5883  
5884

4600:  
GETONES114A:  
P2-T, D+CSPD(C177777), D[C]+ALU15, !ALL ONES STORED HERE  
P3, A#BSPHI(C177777)+D, !WRITE INTO ASP,BSP  
NEXT, J/ALU114A

(4600) DCS(0.00.0.0.0.0) BM(1010..10.11..00.01..101..100...0.1.0..0..0...0.J101...0..1111.0...11.000...000.000.011)

5885  
5886  
5887  
5888  
5889  
5890  
5891

4003: !(FREE)  
ALU114A:  
P2-T, D+ZERO, D[C]+CINMUX, !ALU=(ZERO), D[.] = CIN=PS[C]=(0)  
BUS-A+ASP(C177777), !A=(177777)  
BUS-B+CSPD(C177777), !B=(177777)  
NEXT, J/GETZEROS114A !D=(000000)

(4003) DCS(0.00.0.0.0.0) BM(0011..10.00..11.01..101..000...0.1.0..0..0...0.0101...0..0000.0...11.000...000.000.100)

5892  
5893  
5894  
5895  
5896

4004: !(FREE)  
GETZEROS114A:  
P3, A#BSPHI(000000)+D, !ALL ZEROS STORED HERE IN ASP, BSP  
NEXT, J/GOBUT114A

(4004) DCS(0.00.0.0.0.0) BM(0000..00.11..00.01..100.000...0.0.0..0..0...0.0000...0..1111.0...11.000...000.000.101)

5897  
5898  
5899  
5900  
5901  
5902

4005: !(FREE)  
GOBUT114A:  
SETUP, RETURN/TEST114A2, !EXEC SUBR WHICH:  
NEXT, CALL(DZERO) ! (1) D -> IR  
! (2) BUT(INSTRS) INTO ZTARGET---

(4005) DCS(0.00.0.0.0.0) BM(0100..00.11..00.00..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.010)

5903  
5904  
5905  
5906  
5907  
5908  
5909  
5910

!CHECK THAT D[C] GOT A (0) FROM "CINMUX=PS[C]", ABOVE  
4603:  
TEST114A2:  
PO, LOAD-ENVA(ZTARGET402), !BIT <00> = D[C] = (0)  
LOAD-ERROR(TEST114A2), !ERROR DIRECTORY KEY  
DCS-CTR(C3.), !COMPARE AT TARGET  
NEXT, J/GOBUT114A2

(4603) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...000.000.110)

5911

```

5912 4006: !(FREE)
5913 GOBUT114A2:
5914     SETUP, RETURN/SCOPE114A, !RETURN TO SCOPE LOOP TEST WORD
5915     NEXT, GOTO-PAGE(7), !BOT TABLE
5916     J/BUTD(C)A !D(C)H IN BIT<00>
(4006) DCS[0.00.0.0.0.0] BM[0100..00.00..00.00..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.100]

```

```

5917
5918
5919 4007: !(FREE)
5920 SCOPE114A:
5921     P2-T, D+CSPD(C052525), SAVE-D(C), !STORE A CONSTANT
5922     P3, A#BSPHI(C052525)+D, !
5923     NEXT, BUTD(SCOPE), !NO ERROR: "TEST115A1" (+1 WORDS)
5924     J/TEST115A1 !ERROR: "GETONE5114A" (-6 WORDS)
(4007) DCS[0.00.0.1.0.0] BM[1010..10.11..00.01..111..111...0.1.0..0..0...0.0111...0..1111.0...11.000...110.000.001]

```

```

5925
5926
5927
5928
5929
5930 !-----
5931
5932 !THIS NEXT SET OF 12. TESTS EXERCISES THE "NOT-A" ALU FUNCTION
5933
5934 !-----
5935

```

```

5936 !TESTS 115A 1-3 VERIFIES THAT WITH:
5937 !ALU=(NOT-A), A=(052525), B=(177777), THEN D=(125252), AND D(C)=CINMUX=(1)
5938 4601:
5939 TEST115A1:
5940     PO, LOAD-ENUA(ZTARGET412), !BIT<15:12> = "1010"
5941     LOAD-ERROR(TEST115A1), !ERROR DIRECTORY KEY
5942     DCS-CTR(C7.), !COMPARE AT TARGET
5943     NEXT J/ALU115A1
(4601) DCS[1.00.1.0.0.0] BM[1000..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...110.001.000]

```

```

5944
5945 4610:
5946 ALU115A1:
5947     PO, BUNF-VERIFY, !COUNT
5948     P2-T, D+NOT-A, D(C)+CINMUX, !ALU=(NOT-A) D(C)=CINMUX=(1)
5949     BUS-A+ASPHI(C052525), !A=(052525)
5950     BUS-B+CSPD(C177777), !B=(177777)
5951     NEXT J/GETALTN115A1 !D=(125252)
(4610) DCS[0.00.0.0.0.1] BM[0000..10.00..11.01..111..000...0.1.0..0..0...0.0101...0..0000.0...11.000...000.001.000]

```

```

5952
5953 4010: !(FREE)
5954 GETALTN115A1:
5955     P3, A#BSPHI(C125252)+D, !STORE CONSTANT (125252), HOPEFULLY
5956     NEXT J/GOBUT115A1
(4010) DCS[0.00.0.0.0.0] BM[0000..00.11..00.01..110..000...0.0.0..0..0...0.0000...0..1111.0...11.000...000.001.001]

```

```

5957
5958 4011: !(FREE)
5959 GOBUT115A1:

```

```

5960          SETUP, RETURN/TEST115A4,          !EXEC SUBR WHICH:
5961          ! (1) D<15:12> -> IR<15:12>
5962          NEXT, CALL(D15-12)                  ! (2) BUT(IR15-12) INTO ZTARGET---
(4011) DCS(0.00.0.0.0.0) BM(0100..00.11..00.00..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.100)

5963
5964          !CHECK THAT D(C) GOT A (1) FROM "CINMUX=(1)", ABOVE
5965          4604:
5966          TEST115A4:
5967          PO,                                !BIT <00> = D(C) = (1)
5968          LOAD-ENUA(ZTARGET403),             !ERROR DIRECTORY KEY
5969          LOAD-ERROR(TEST115A4),             !COMPARE AT TARGET
5970          DCS-CTR(C3.),                       !COUNT
5971          BUMP-VERIFY,
5972          NEXT, J/GOBUT115A4
(4604) DCS(1.00.1.0.0.1) BM(1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...000.001.010)

5973          4012: !(FREE)
5974          GOBUT115A4:
5975          SETUP, RETURN/TEST115A2,           !RETURN TO START OF NEXT SUBTEST
5976          NEXT, GOTO-PAGE(7),               !BUT TABLE
5977          J/BUTD(C)A                          !D(C)H IN BIT<00>
5978          (4012) DCS(0.00.0.0.0.0) BM(0100..00.11..00.01..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.100)

5979
5980          !CHECK BIT<11:06> = "1010 10"
5981          4616:
5982          TEST115A2:
5983          PO,                                !INSTR5-E78 OUTPUT FOR BIT <11:06>="101 010"
5984          LOAD-ENUA(ZTARGET412),             !ERROR DIRECTORY KEY
5985          LOAD-ERROR(TEST115A2),             !COMPARE AT TARGET
5986          DCS-CTR(C6.),                       !COUNT
5987          BUMP-VERIFY,
5988          NEXT, J/GOBUT115A2
(4616) DCS(1.00.1.0.0.1) BM(1001..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...000.001.011)

5989          4013: !(FREE)
5990          GOBUT115A2:
5991          SETUP, RETURN/TEST115A3,           !EXEC SUBR WHICH:
5992          ! (1) D<11:06> -> IR<11:06>
5993          NEXT, CALL(D11-06)                  ! (2) BUT(INSTR5) INTO ZTARGET---
5994          (4013) DCS(0.00.0.0.0.0) BM(0100..00.11..00.01..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.110)

5995
5996          !CHECK BIT<05:00> = "10 1010"
5997          4615:
5998          TEST115A3:
5999          PO,                                !INSTR5-E88 OUTPUT FOR BIT <05:00> = "101 010"
6000          LOAD-ENUA(ZTARGET425),             !ERROR DIRECTORY KEY
6001          LOAD-ERROR(TEST115A3),             !COMPARE AT TARGET
6002          DCS-CTR(C6.),
6003          NEXT, J/GOBUT115A3
6004
6005

```



```

(4615) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..010..101...0.0.0..0..0...0.0000...0..0000.0...11.000...000.001.100)
6006
6007 4014: !(FREE)
6008 GOBUT115A3:
6009     SETUP, RETURN/TEST115B1,           !EXEC SUBR WHICH:
6010                                     ! (1) D<05:00> -> IR<05:00>
6011     NEXT, CALL(D05-00)                 ! (2) BUT(INSTR5) INTO ZTARGET---
(4014) DCS(0.00.0.0.0.0) BM(0100..00.11..00.01..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.000)
6012
6013
6014
6015
6016 ! - - - - -
6017
6018 !TESTS 115B i-3 VERIFIES THAT WITH:
6019 !ALU=(NOT-A), A=(125252), B=(177777), THEN D=(052525), AND D[C]+PS[C]=(0)
6020 4614:
6021 TEST115B1:
6022     PO, LOAD-ENUA(ZTARGET405),           !BIT <15:12> = "0101"
6023         LOAD-ERROR(TEST115B1),         !ERROR DIRECTORY KEY
6024         DCS-CTR(C6.),                  !COMPARE AT TARGET
6025     NEXT, J/ALU115B1
(4614) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...000.001.101)
6026
6027 4015: !(FREE)
6028 ALU115B1:
6029     PO, BUMP-VERIFY,                   !COUNT
6030         P2-T, D+NOT-A, D[C]+PS[C],     !ALU=(NOT-A), D[C]=PS[C]=(0)
6031         BUS-A+ASPHI(C125252),         !A=(125252)
6032         BUS-B+CSPD(C177777),         !B=(177777)
6033     NEXT, J/GOBUT115B1                 !D=(052525)
(4015) DCS(0.00.0.0.0.0) BM(0000..10.00..11.01..110..001...0.1.0..0..0...0.0101...0..0000.0...11.000...000.001.110)
6034
6035 4016: !(FREE)
6036 GOBUT115B1:
6037     SETUP, RETURN/TEST115B4,         !EXEC SUBR WHICH:
6038                                     ! (1) D<15:12> -> IR<15:12>
6039     NEXT, CALL(D15-12)                 ! (2) BUT(IR15-12) INTO ZTARGET---
(4016) DCS(0.00.0.0.0.0) BM(0100..00.11..00.00..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.100)
6040
6041 !CHECK THAT D[C] GOT A (0) FROM "PS[C]", ABOVE
6042 4602:
6043 TEST115B4:
6044     PO, LOAD-ENUA(ZTARGET402),         !BIT <00> = D[C] = (0)
6045         LOAD-ERROR(TEST115B4),         !ERROR DIRECTORY KEY
6046         DCS-CTR(C3.),                  !COMPARE AT TARGET
6047         BUMP-VERIFY,                   !COUNT
6048     NEXT, J/GOBUT115B4
(4602) DCS(1.00.1.0.0.1) BM(1100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...000.001.111)
6050
6051 4017: !(FREE)

```

```

6052 GOBUT115B4:
6053     SETUP, RETURN/TEST115B2,           !RETURN TO START OF NEXT SUBTEST
6054     NEXT,  GOTO-PAGE(7),              !BUT TABLE
6055     J/BUTD(C)A                          !D(C)H IN BIT<00>
(4017) DCS[0.00.0.0.0.0] BM[0100..00.11..00.01..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.100]

6056
6057
6058 !CHECK BIT<11:06> = "0101 01"
6059 4613:
6060 TEST115B2:
6061     PO,      LOAD-ENVA(ZTARGET405),      !INSTRS-E78 OUTPUT FOR BIT <11:06> = "010 101"
6062     LOAD-ERROR(TEST115B2),              !ERROR DIRECTORY KEY
6063     DCS-CTR(C6.),                       !COMPARE AT TARGET
6064     BUMP-VERIFY,                         !COUNT
6065     NEXT,   J/GOBUT115B2
(4613) DCS[1.00.1.0.0.1] BM[1001..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...000.010.000]

6066
6067 4020: !(FREE)
6068 GOBUT115B2:
6069     SETUP, RETURN/TEST115B3,           !EXEC SUBR WHICH:
6070     NEXT,  CALL[011-06]                ! (1) D<11:06> -> IR<11:06>
6071     ! (2) BUT(INSTRS) INTO ZTARGET---
(4020) DCS[0.00.0.0.0.0] BM[0100..00.11..00.01..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.110]

6072
6073
6074
6075 !CHECK BIT<05:00> = "01 0101"
6076 4612:
6077 TEST115B3:
6078     PO,      LOAD-ENVA(ZTARGET432),      !INSTRS-E88 OUTPUT FOR BIT<05:00> = "010 101"
6079     LOAD-ERROR(TEST115B3),              !ERROR DIRECTORY KEY
6080     DCS-CTR(C6.),                       !COMPARE AT TARGET
6081     NEXT,   J/GOBUT115B3
(4612) DCS[1.00.1.0.0.0] BM[1001..00.11..11.00..011..010...0.0.0..0..0...0.0000...0..0000.0...11.000...000.010.001]

6083
6084 4021: !(FREE)
6085 GOBUT115B3:
6086     SETUP, RETURN/SCOPE115B,          !EXEC SUBR WHICH:
6087     NEXT,  CALL[005-00]                ! (1) D<05:00> -> IR<05:00>
6088     ! (2) BUT(INSTRS) INTO ZTARGET---
(4021) DCS[0.00.0.0.0.0] BM[0100..00.00..00.10..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.000]

6089
6090
6091 4022: !(FREE)
6092 SCOPE115B:
6093     NEXT,  BUTD(SCOPE),                 !NO ERROR: "TEST115C1" (+1.WORDS)
6094     J/TEST115C1                          ! ERROR: "ALJ115A1" (-17.WORDS)
(4022) DCS[0.00.0.1.0.0] BM[0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.001.001]

6095
6096

```

6097  
6098

6099  
6100

6101  
6102

6103  
6104

6105  
6106

6107  
6108

! - - - - -  
! TESTS 115C 1-3 VERIFIES THAT WITH:  
! ALU=(NOT-A), A=(052525), B=(000000), THEN D=(125252)

4611:  
TEST115C1:  
PO, LOAD-ENUA(ZTARGET412), !BIT <15:12> = "1010"  
LOAD-ERROR(TEST115C1), !ERROR DIRECTORY KEY  
DCS-CTR(C6.), !COMPARE AT TARGET  
J/ALU115C1

(4611) DCS[1.00.1.0.0.0] BM[1001..00.11..11 00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...110.010.010]

6109  
6110

6111  
6112

6113  
6114

6115  
6116

4622:  
ALU115C1:  
PO, BUMP-VERIFY, !COUNT  
P2-T, D=NOT-A, SAVE-DIC, !ALU=(NOT-A)  
BUS-A+ASPHI(C052525), !A=(052525)  
BUS-B+CSPO(C000000), !B=(000000)  
J/GOBUT115C1 !D=(125252)

(4622) DCS[0.00.0.0.0.1] BM[0000..10.00..11.01..111..111...0.1.0..0..0...0.0100...0..0000.0...11.000...000.010.011]

6117  
6118

6119  
6120

6121  
6122

4023: !(FREE)  
GOBUT115C1:  
SETUP, RETURN/TEST115C2, !EXEC SUBR WHICH:  
NEXT, CALL[D15-12] ! (1) D<15:12> -> IR<15:12>  
! (2) BUT(IR15-12) INTO ZTARGET---

(4023) DCS[0.00.0.0.0.0] BM[0100..00.11..00.01..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.100]

6123  
6124

6125  
6126

6127  
6128

6129  
6130

6131

!CHECK BIT<11:06> = "1010 10"  
4617:  
TEST115C2:  
PO, LOAD-ENUA(ZTARGET412), !INSTRS-E78 OUTPUT FOR BIT <11:06> = "101 010"  
LOAD-ERROR(TEST115C2), !ERROR DIRECTORY KEY  
DCS-CTR(C6.), !COMPARE AT TARGET  
BUMP-VERIFY, !COUNT  
J/GOBUT115C2

(4617) DCS[1.00.1.0.0.1] BM[1001..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...000.010.100]

6132  
6133

6134  
6135

6136  
6137

4024: !(FREE)  
GOBUT115C2:  
SETUP, RETURN/TEST115C3, !EXEC SUBR WHICH:  
NEXT, CALL[D11-06] ! (1) D<11:06> -> IR<11:06>  
! (2) BUT(INSTRS) INTO ZTARGET---

(4024) DCS[0.00.0.0.0.0] BM[0100..00.11..00.10..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.110]

6138  
6139

6140  
6141

6142  
6143

!CHECK BIT<05:00> = "10 1010"  
4627:  
TEST115C3:

```

6144
6145      PO,      LOAD-ENUA(ZTARGET425),      !INSTRS-E88 OUTPUT FOR BIT<05:00> = "101 010"
6146      LOAD-ERROR(TEST115C3),      !ERROR DIRECTORY KEY
6147      DCS-CTR(C6.),      !COMPARE AT TARGET
6148      NEXT,      J/GOBUT115C3
(4627) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..010..101...0.0.0..0..0...0.0000...0..0000.0...11.000...000.010.101)
6149
6150      4025:      !(FREE)
6151      GOBUT115C3:
6152      SETUP,      RETURN/TEST115D1,      !EXEC SUBR WHICH:
6153      !          ! (1) D<05:00> -> IR<05:00>
6154      NEXT,      CALL(D05-00)      ! (2) BUT(INSTRS) INTO ZTARGET---
(4025) DCS(0.00.0.0.0.0) BM(0100..00.11..00.10...10..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.000)
6155
6156
6157
6158
6159      ! - - - - -
6160
6161      !TESTS 115D 1-3 VERIFIES THAT WITH:
6162      !ALU=(NOT-A, A=(125252), B=(000000), THEN D=(052525)
6163      4626:
6164      TEST115D1:
6165      PO,      LOAD-ENUA(ZTARGET405),      !BIT<15:12> = "0101"
6166      LOAD-ERROR(TEST115D1),      !ERROR DIRECTORY KEY
6167      DCS-CTR(C6.),      !COMPARE AT TARGET
6168      NEXT,      J/ALU115D1
(4626) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...000.010.110)
6169
6170      4026:      !(FREE)
6171      ALU115D1:
6172      PO,      BUMP-VERIFY,      !COUNT
6173      P2-T,      D+NOT-A, SAVE-D(C),      !ALU=(NOT-A), D(C)+0
6174      BUS-A+ASPHI(C125252),      !A=(125252)
6175      BUS-B+CSPD(C000000),      !B=(000000)
6176      NEXT,      J/GOBUT115D1      !D=(052525)
(4026) DCS(0.00.0.0.0.1) BM(0000..10.00..11.01..110..111...0.1.0..0..0...0.0100...0..0000.0...11.000...000.010.111)
6177
6178      4027:      !(FREE)
6179      GOBUT115D1:
6180      SETUP,      RETURN/TEST115D2,      !EXEC SUBR WHICH:
6181      !          ! (1) D<15:12> -> IR<15:12>
6182      NEXT,      CALL(D15-12)      ! (2) BUT(IR15-12) INTO ZTARGET---
(4027) DCS(0.00.0.0.0.0) BM(0100..00.11..00.10..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.100)
6183
6184      !CHECK BIT<11:06> = "0101 01"
6185      4625:
6186      TEST115D2:
6187      PO,      LOAD-ENUA(ZTARGET405),      !INSTRS-E78 OUTPUT FOR BIT<11:06> = "010 101"
6188      LOAD-ERROR(TEST115D2),      !ERROR DIRECTORY KEY
6189      DCS-CTR(C6.),      !COMPARE AT TARGET
6190      BUMP-VERIFY,      !COUNT

```

K011-K MICRO V00A-1 00:00:03 12-MAR-77

```

6191      NEXT, J/GOBUT115D2
(4625) DCS(1.00.1.0.0.1) BM(1001..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...000.011.000)
6192
6193      4030: !(FREE)
6194      GOBUT115D2:
6195      SETUP, RETURN/TEST115D3,
6196
6197      NEXT, CALL(D11-06)
(4030) DCS(0.00.0.0.0.0) BM(0100..00.11..00.10..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.110)
6198
6199
6200
6201      !CHECK BIT<05:00> = "01 0101"
6202      4624:
6203      TEST115D3:
6204
6205      PO, LOAD-ENUA(ZTARGET432),
6206      LOAD-ERROR(TEST115D3),
6207      DCS-CTR(C6.),
6208      NEXT, J/GOBUT115D3
(4624) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..011..010...0.0.0..0..0...0.0000...0..0000.0...11.000...000.011.001)
6209
6210      4031: !(FREE)
6211      GOBUT115D3:
6212      SETUP, RETURN/SCOPE115D,
6213
6214      NEXT, CALL(D05-00)
(4031) DCS(0.00.0.0.0.0) BM(0100..00.00..00.11..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.000)
6215
6216      4032: !(FREE)
6217      SCOPE115D:
6218      PO, BUMP-VERIFY,
6219      NEXT, BUTD(SCOPE),
6220      J/TEST116A1
(4032) DCS(0.00.0.1.0.1) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.010.011)
6221
6222
6223
6224
6225      !-----
6226
6227      !THIS NEXT SET OF 12. TESTS EXERCISES THE ALU FUNCTIONS
6228      !"ZERO" AND "NOT-A-AND-B", AND THE D(C) INPUTS "ALU15" AND "D(C)"
6229
6230      !-----
6231
6232      !TESTS 116A 1-5 VERIFIES THAT WITH:
6233      !ALU=(NOT-A-AND-B), A=(000000), B=(125252), THEN D=(125252)
6234      4623:
6235      TEST116A1:
6236      PO, LOAD-ENUA(ZTARGET412),
6237      LOAD-ERROR(TEST116A1),

```

```

!BIT<15:12> = "1010"
!ERROR DIRECTORY KEY

```

```

6238          DCS-CTR(C6.),          !COMPARE AT TARGET
6239      NEXT      J/ALU116A1
(4623) DCS[1.00.1.0.0.0] BM[1001..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...110.011.100]
6240
6241      4634:
6242      ALU116A1:
6243          PO,          BUMP-VERIFY,          !COUNT
6244          P2-T,        D=NOT-A-AND-B, D[C]=ALU15, !ALU=(NOT-A-AND-B), D[C]=(1)
6245          BUS-A=ASPH1(C000000),          !A=(000000)
6246          BUS-B=CSPD(C125252),          !B=(125252)
6247      NEXT      J/GOBUT116A1          !D=(125252)
(4634) DCS[0.00.0.0.0.1] BM[0010..10.00..11.01..100..100...0.1.0..0..0...0.0110...0..0000.0...11.000...000.011.011]
6248
6249      4033: !(FREE)
6250      GOBUT116A1:
6251          SETUP, RETURN/TEST116A2,          !EXEC SUBR WHICH:
6252          NEXT      CALL(D15-12)          ! (1) D<15:12> -> IR<15:12>
6253      (4033) DCS[0.00.0.0.0.0] BM[0100..00.11..00.10..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.100]
        ! (2) BUT (IR15-12) INTO ZTARGET---
6254      !CHECK THAT D[C] GOT A (1) FROM "ALU15," ABOVE
6255
6256      4621:
6257      TEST116A2:
6258          PO,          LOAD-ENUA(ZTARGET403),          !BIT <00> = D[C] = (1)
6259          LOAD-ERROR(TEST116A2),          !ERROR DIRECTORY KEY
6260          DCS-CTR(C3.),          !COMPARE AT TARGET
6261          BUMP-VERIFY,          !COUNT
6262      NEXT      J/GOBUT116A2
(4621) DCS[1.00.1.0.0.1] BM[1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...000.011.100]
6263      4034: !(FREE)
6264      GOBUT116A2:
6265          SETUP, RETURN/TEST116A3,          !RETURN TO START OF NEXT SUBTEST
6266          NEXT,      GOTO-PAGE(7),          !BUT TABLE
6267          J/BUTD[C]A          !D[C]H IN BIT <00>
(4034) DCS[0.00.0.0.0.0] BM[0100..00.11..00.10..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.100]
6268
6269      !CHECK BIT<11:06> = "1010 10"
6270      4620:
6271      TEST116A3:
6272          PO,          LOAD-ENUA(ZTARGET412),          !INSTR5-E78 OUTPUT FOR BIT <11:06> = "101 010"
6273          LOAD-ERROR(TEST116A3),          !ERROR DIRECTORY KEY
6274          DCS-CTR(C6.),          !COMPARE AT TARGET
6275          BUMP-VERIFY,          !COUNT
6276      NEXT      J/GOBUT116A3
(4620) DCS[1.00.1.0.0.1] BM[1001..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...000.011.101]
6277
6278      4035: !(FREE)
6279      GOBUT116A3:
6280          SETUP, RETURN/TEST116A4,          !EXEC SUBR WHICH:
6281          NEXT,      CALL(D11-06)          ! (1) D<11:06> -> IR<11:06>
6282          ! (2) BUT(INSTR5) INTO ZTARGET---

```

(4035) DCS(0.00.0.0.0.0) BM(0100..00.11..00.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.1101

6283  
6284  
6285  
6286  
6287  
6288  
6289  
6290  
6291  
6292  
6293

!CHECK BIT<05:00> = "10 1010"

4637:  
TEST116A4:

PO, LOAD-ENUA(ZTARGET425),  
LOAD-ERROR(TEST116A4),  
DCS-CTR(C6.),  
NEXT J/GOBUT116A4

!INSTRS-EBB OUTPUT FOR BIT <05:00> ="101 010"  
!ERROR DIRECTORY KEY  
!COMPARE AT TARGET

(4637) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..010..101...0.0.0..0..0...0.0000...0..0000.0...11.000...000.011.1101

6294  
6295  
6296  
6297  
6298  
6299

4036: !(FREE)  
GOBUT116A4:

SETUP, RETURN/TEST116A5,  
NEXT CALL(D05-00)

!EXEC SUBR WHICH:  
!(1) D<05:00> -> IR<05:00>  
!(2) BUT(INSTRS) INTO ZTARGET---

(4036) DCS(0.00.0.0.0.0) BM(0100..00.11..00.11..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.0001

6300  
6301  
6302  
6303  
6304  
6305  
6306  
6307  
6308  
6309

!CHECK THAT D[C] WAS PROPOGATED UNCHANGED AS A (1), VIA D[C]+D[C]

4636:  
TEST116A5:

PO, LOAD-ENUA(ZTARGET403),  
LOAD-ERROR(TEST116A5),  
DCS-CTR(C3.),  
NEXT J/GOBUT116A5

!BIT<01> = D[C] = (1)  
!ERROR DIRECTORY KEY  
!COMPARE AT TARGET

(4636) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...000.011.1111

6310  
6311  
6312  
6313  
6314  
6315

4037: !(FREE)  
GOBUT116A5:

SETUP, RETURN/TEST116B,  
NEXT, GOTO-PAGE(7),  
J/BUTD[C]B

!RETURN TO START OF NEXT SUBTEST  
!BUT TABLE  
!D[C]H IN BIT <01>

(4037) DCS(0.00.0.0.0.0) BM(0100..00.11..00.00..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.101.0001

6316  
6317  
6318  
6319  
6320  
6321  
6322  
6323  
6324  
6325  
6326  
6327  
6328  
6329

!TEST 116B VERIFIES THAT WITH:  
!ALU=(ZERO), A=(125252), B=(052525), THEN D=(000000)

4606:  
TEST116B:

PO, LOAD-ENUA(ZTARGET434),  
LOAD-ERROR(TEST116B),  
DCS-CTR(C6.),

!INSTRS FOR IR=(000000)  
!ERROR DIRECTORY KEY  
!COMPARE AT TARGET

```

6330      NEXT      J/ALU1168
(4606) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...000.100.000)
6331
6332      4040: !(FREE)
6333      ALU1168:
6334      P2-T,      D+ZERO, SAVE-D(C),           !ALU=(ZERO), D(C)=D(C)=(1)
6335      BUS-A+ASPHI(C125252),           !A=(125252)
6336      BUS-B+CSPD(C052525),           !B=(052525)
6337      NEXT      J/GOBUT1168           !D=(000000)
(4040) DCS(0.00.0.0.0.0) BM(0011..10.00..11.01..110..111...0.1.0..0..0...0.0111...0..0000.0...11.000...000.100.001)
6338
6339      4041: !(FREE)
6340      GOBUT1168:
6341      SETUP,    RETURN/SCOPE116C,      !EXEC SUBR WHICH:
6342      !          ! (1) D-> IR
6343      NEXT      CALL(DZERO)             ! (2) BUT(INSTR5) INTO ZTARGET---
(4041) DCS(0.00.0.0.0.0) BM(0100..00.00..01.00..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.010)
6344
6345
6346
6347      4042: !(FREE)
6348      SCOPE116C:
6349      NEXT,      EUTD(SCOPE),           !NO ERROR: "TEST116C1" (+1. WORDS)
6350      J/TEST116C1,                       ! ERROR: "ALU116A1" (-13. WORDS)
(4042) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.011.101)
6351
6352
6353
6354      ! - - - - -
6355
6356      !TEST 116C 1-5 VERIFIES THAT WITH:
6357      !ALU=(NOT-A-AND-B), A=(000000), B=(052525), THEN D=(052525)
6358      4635:
6359      TEST116C1:
6360      PD,      LOAD-ENVA(ZTARGET405),      !BIT<15:12> = "0101"
6361      LOAD-ERROR(TEST116C1),           !ERROR DIRECTORY KEY
6362      DCS-CTR(C6.),                     !COMPARE AT TARGET
6363      NEXT      J/ALU116C1
(4635) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...110.100.110)
6364
6365      4646:
6366      ALU116C1:
6367      PD,      BUMP-VERIFY,               !COUNT
6368      P2-T,      D+NOT-A-AND-B, D(C)+ALU15, !ALU=(NOT-A-AND-B), D(C)=(0)
6369      BUS-A+ASPHI(C000000),           !A=(000000)
6370      BUS-B+CSPD(C052525),           !B=(052525)
6371      NEXT      J/GOBUT116C1           !D=(052525)
(4646) DCS(0.00.0.0.0.1) BM(0010..10.00..11.01..100..100...0.1.0..0..0...0.0111...0..0000.0...11.000...000.100.011)
6372
6373      4043: !(FREE)
6374      GOBUT116C1:
6375      SETUP,    RETURN/TEST116C2,      !EXEC SUBR WHICH:
6376      !          ! (1) D<15:12> -> IR<15:12>

```



KD11-K MICRO VO0A-1 00:00:03 12-MAR-77

```

6377      NEXT, CALL(D15-12)                !(2) BUT(IR15-12) INTO ZTARGET---
(4043) DCS(0 00.0.0.0.0) BM(0100..00.11..00.11..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.100)
6378
6379      !CHECK THAT D(C) GOT A (0) FROM "ALU15", ABOVE
6380      4633:
6381      TEST116C2:
6382          PO,          LOAD-ENVA(ZTARGET402),          !BIT <00> = D(C) = (0)
6383          LOAD-ERROR(TEST116C2),          !ERROR DIRECTORY KEY
6384          DCS-CTR(C3.),          !COMPARE AT TARGET
6385          BUMP-VERIFY,          !COUNT
6386          J/GOBUT116C2
(4633) DCS(1.00.1.0.0.1) BM(1100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...000.100.100)
6387
6388      4044: !(FREE)
6389      GOBUT116C2:
6390          SETUP, RETURN/TEST116C3,          !RETURN TO START OF NEXT SUBTEST
6391          NEXT, GOTO-PAGE(7),          !BUT TABLE
6392          J/BUT(D)A          !D(C) IN BIT<00>
(4044) DCS(0.00.0.0.0.0) BM(0100..00.11..00.11..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.100)
6393
6394      !CHECK BIT<11:06> = "0101 01"
6395      4632:
6396      TEST116C3:
6397          PO,          LOAD-ENVA(ZTARGET405),          !INSTRS-E78 OUTPUT FOR BIT<11:06>="010 101"
6398          LOAD-ERROR(TEST116C3),          !ERROR DIRECTORY KEY
6399          DCS-CTR(C6.),          !COMPARE AT TARGET
6400          BUMP-VERIFY,          !COUNT
6401          J/GOBUT116C3
(4632) DCS(1.00.1.0.0.1) BM(1001..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...000.100.101)
6402
6403      4045: !(FREE)
6404      GOBUT116C3:
6405          SETUP, RETURN/TEST116C4,          !EXEC SUBR WHICH:
6406          NEXT, CALL(D11-06)          !!(1) D<11:06>-> IR<11:06>
6407          !!(2) BUT(INSTRS) INTO ZTARGET---
(4045) DCS(0.00.0.0.0.0) BM(0100..00.11..00.11..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.110)
6408
6409
6410      !CHECK BIT <05:00> = "01 0101"
6411      4631:
6412      TEST116C4:
6413
6414          PO,          LOAD-ENVA(ZTARGET432),          !INSTRS-E88 OUTPUT FOR BIT <05:00> = "010 101"
6415          LOAD-ERROR(TEST116C4),          !ERROR DIRECTORY KEY
6416          NEXT, J/GOBUT116C4
(4631) DCS(1.00.0.0.0.0) BM(0000..00.11..11.00..011..010...0.0.0..0..0...0.0000...0..0000.0...11.000...000.100.110)
6417
6418
6419      4046: !(FREE)
6420      GOBUT116C4:
6421          SETUP, RETURN/TEST116C5,          !EXEC SUBR WHICH:

```

```

6422                                     !(1) D<05:00> -> IR<05:00>
6423     NEXT, CALL(D05-00)                !(2) BUT(INSTRS) INTO ZTARGET---
(4046) DCS(0.00.0.0.0.0) BM(0100..00.11..00 11..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.000)
6424
6425
6426
6427     !CHECK THAT D(C) WAS PROPOGATED UNCHANGED AS A (0), VIA D(C)+D(C)
6428     4630:
6429     TEST116C5:
6430         PO,          LOAD-ENUA(ZTARGET401),          !BIT<01> = D(C) = (0)
6431         LOAD-ERROR(TEST116C5),          !ERROR DIRECTORY KEY
6432         DCS-CTR(C3.),          !COMPARE AT TARGET
6433     NEXT, J/GOBUT116C5
(4630) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...000.100.111)
6434
6435     4047: !(FREE)
6436     GOBUT116C5:
6437
6438         SETUP, RETURN/TEST1160,          !RETURN TO START OF NEXT SUBTEST
6439         NEXT, GOTO-PAGE(7),          !BUT TABLE
6440         J/BUTD(C)8          !D(C)H IN BIT <01>
(4047) DCS(0.00.0.0.0.0) BM(0100..00.11..01.00..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.101.000)
6441
6442
6443
6444
6445
6446
6447     ! - - - - -
6448
6449     !TEST 1160 VERIFIES THAT WITH:
6450     !ALU=(ZERO), A=(052525), B=(125252), THEN D=(000000)
6451     4645:
6452     TEST1160:
6453         PO,          LOAD-ENUA(ZTARGET434),          !INSTRS FOR IR=(000000)
6454         LOAD-ERROR(TEST1160),          !ERROR DIRECTORY KEY
6455         DCS-CTR(C6.),          !COMPARE AT TARGET
6456     NEXT, J/ALU1160
(4645) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...000.101.000)
6457
6458     4050: !(FREE)
6459     ALU1160:
6460         P2-T, D+ZERO, SAVE-D(C),          !ALU=(ZERO), D(C)=D(C)=(0)
6461         BUS-A+ASPHI(C052525),          !A=(052525)
6462         BUS-B+CSPD(C125252),          !B=(125252)
6463     NEXT, J/GOBUT1160          !D=(000000)
(4050) DCS(0.00.0.0.0.0) BM(0011..10.00..11.01..111..111...0.1.0..0..0...0.0110...0..0000.0...11.000...000.101.001)
6464
6465     4051: !(FREE)
6466     GOBUT1160:
6467         SETUP, RETURN/SCOPE1160,          !EXEC SUBR WHICH:
6468                                     !(1) D -> IR

```

```

6469      NEXT, CALL(DZERO)                !(2) BUT(INSTR5) INTO ZTARGET---
(4051) DCS(0.00.0.0.0.0) BM(0100..00.00..01.01..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.010)
6470
6471
6472
6473      4052: !(FREE)
6474      SCOPE1160:
6475      NEXT, BUTD(SCOPE),                !NO ERROR: "TEST117A1" (+1. WORDS)
6476      J/TEST117A1                       ! ERROR: "ALU116C1" (-13. WORDS)
(4052) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.100.111)
6477
6478
6479
6480
6481
6482      ! -----
6483
6484      ! THIS NEXT SET OF 9. TESTS EXERCIZES THE ALU FUNCTION
6485      ! "A-AND-NOT-B", AND THE CARRYOUT OF "CINMUX=D(C)" INTO D(C)
6486
6487
6488
6489
6490      ! -----
6491
6492      ! TESTS 117A 1-4 VERIFIES THAT WITH:
6493      ! ALU=(A-AND-NOT-B), A=(177777), B=(125252), THEN D=(052525)
6494      4647:
6495      TEST117A1:
6496      PO,          LOAD-ENUA(ZTARGET405),    !BIT<15:12> = "0101"
6497      P2-T,        LOAD-ERROR(TEST117A1),    !ERROR DIRECTORY KEY
6498      NEXT,        DCS-CTR(C6.),             !COMPARE AT TARGET
6499      J/ALU117A1
(4647) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...110.110.110)
6500
6501      4666:
6502      ALU117A1:
6503      PO,          BUMP-VERIFY,              !COUNT
6504      P2-T,        D+A-AND-NOT-B, D(C)+CINMUX, !ALU=(A-AND-NOT-B), D(C)=CIN=D(C)=(0)
6505      BUS-A+ASPHI(C177777),                 !A=(177777)
6506      BUS-B+CSPD(C125252),                 !B=(125252)
6507      NEXT,        J/GOBUT117A1             !D=(052525)
(4666) DCS(0.00.0.0.0.1) BM(0111..10.00..11.01..101..000...0.1.0..0..0...0.0110...0..0000.0...11.000...000.101.011)
6508
6509      4053: !(FREE)
6510      GOBUT117A1:
6511      SETUP,      RETURN/TEST117A2,        !EXEC SUBR WHICH:
6512      NEXT,        CALL(D15-12)            ! (1) D<15:12> -> IR<15:12>
6513      (4053) DCS(0.00.0.0.0.0) BM(0100..00.11..01.00..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.100)
6514
6515      !CHECK BIT<11:06> = "0101 01"

```

```

6516 4644:
6517 TEST117A2:
6518     PO,          LOAD-ENVA(ZTARGET405),          !INSTR5-E78 OUTPUT FOR BIT<11:06>="010 101"
6519     LOAD-ERROR(TEST117A2),          !ERROR DIRECTORY KEY
6520     DCS-CTR(C6.),          !COMPARE AT TARGET
6521     BUMP-VERIFY,          !COUNT
6522     NEXT,         J/GOBUT117A2
(4644) DCS[1.00.1.0.0.1] BM[1001..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...000.101.100]

6523 4054: !(FREE)
6524 GOBUT117A2:
6525     SETUP,       RETURN/TEST117A3,          !EXEC SUBR WHICH:
6526     NEXT,         CALL[D11-06]              ! (1) D<11:06> -> IR<11:06>
6527     ! (2) BUT(INSTR5) INTO ZTARGET---
6528 (4054) DCS[0.00.0.0.0.0] BM[0100..00.11..01.00..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.110]

6529
6530
6531 !CHECK BIT<05:00> = "01 0101"
6532 4643:
6533 TEST117A3:
6534     PO,          LOAD-ENVA(ZTARGET432),          !INSTR5-E88 OUTPUT FOR BIT <05:00>="010 101"
6535     LOAD-ERROR(TEST117A3),          !ERROR DIRECTORY KEY
6536     DCS-CTR(C6.),          !COMPARE AT TARGET
6537     BUMP-VERIFY,
6538     NEXT,         J/GOBUT117A3
(4643) DCS[1.00.1.0.0.0] BM[1001..00.11..11.00..011..010...0.0.0..0..0...0.0000...0..0000.0...11.000...000.101.101]

6539
6540 4055: !(FREE)
6541 GOBUT117A3:
6542     SETUP,       RETURN/TEST117A4,          !EXEC SUBR WHICH:
6543     NEXT,         CALL[D05-00]              ! (1) D<05:00> -> IR<05:00>
6544     ! (2) BUT(INSTR5) INTO ZTARGET---
6545 (4055) DCS[0.00.0.0.0.0] BM[0100..00.11..01.00..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.000]

6546 !CHECK THAT D[C] GOT A (0) FROM CINMUX, ABOVE
6547 4642:
6548 TEST117A4:
6549     PO,          LOAD-ENVA(ZTARGET413),          !BIT<02> = D[C] = (0)
6550     LOAD-ERROR(TEST117A4),          !ERROR DIRECTORY KEY
6551     DCS-CTR(C3.),          !COMPARE AT TARGET
6552     BUMP-VERIFY,          !COUNT
6553     NEXT,         J/GOBUT117A4
(4642) DCS[1.00.1.0.0.1] BM[1100..00.11..11.00..001..011...0.0.0..0..0...0.0000...0..0000.0...11.000...000.101.110]

6554
6555 4056: !(FREE)
6556 GOBUT117A4:
6557     SETUP,       RETURN/TEST117B1,          !RETURN TO START OF NEXT SUBTEST
6558     NEXT,         GOTO-PAGE(7),            !BUT TABLE
6559     J/BUTD[C]C          !D[C]H IN BIT<02>
6560 (4056) DCS[0.00.0.0.0.0] BM[0100..00.11..01.00..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.000]

```

```

6561
6562
6563
6564
6565 ! - - - - -
6566
6567 ! TESTS 117B 1-3 VERIFIES THAT WITH:
6568 ! ALU=(A-AND-NOT-B), A=(177777), B=(052525), THEN D=(125252)
6569 4641:
6570 TEST117B1:
6571     PO,      LOAD-ENVA(ZTARGET412),      ! BIT<15:12> = "1010"
6572     LOAD-ERROR(TEST117B1),      ! ERROR DIRECTORY KEY
6573     DCS-CTR(C6.),      ! COMPARE AT TARGET
6574     NEXT,     J/ALU117B1
(4641) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...000.101.1111)

6575
6576 4057:  !(FREE)
6577 ALU117B1:
6578     PO,      BUMP-VERIFY,      ! COUNT
6579     P2-T,    D+A-AND-NOT-B, D(C)+ALU15, ! ALU=(A-AND-NOT-B), D(C)=ALU15=(1)
6580     BUS-A+ASPHI(C177777),      ! A=(177777)
6581     BUS-B+CSPD(C052525),      ! B=(052525)
6582     NEXT,     J/GOBUT117B1      ! D=(125252)
(4057) DCS(0.00.0.0.0.1) BM(0111..10.00..11.01..101..100...0.1.0..0..0...0.0111...0..0000.0...11.000...000.110.000)

6583
6584 4060:  !(FREE)
6585 GOBUT117B1:
6586     SETUP,   RETURN/TEST117B2,      ! EXEC SUBR WHICH:
6587     ! (1) D<15:12> -> IR<15:12>
6588     NEXT,    CALL(D15-12)           ! (2) BUT(IR15-12) INTO ZTARGET---
(4060) DCS(0.00.0.0.0.0) BM(0100..00.11..01.00..000...111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.100)

6589
6590 ! CHECK BIT <11:06> = "1010 10"
6591 4640:
6592 TEST117B2:
6593     PO,      LOAD-ENVA(ZTARGET412),      ! INSTR5-E7B OUTPUT FOR BIT <11:06>="101 010"
6594     LOAD-ERROR(TEST117B2),      ! ERROR DIRECTORY KEY
6595     DCS-CTR(C6.),      ! COMPARE AT TARGET
6596     BUMP-VERIFY,      ! COUNT
6597     NEXT,     J/GOBUT117B2
(4640) DCS(1.00.1.0.0.1) BM(1001..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...000.110.001)

6598
6599 4061:  !(FREE)
6600 GOBUT117B2:
6601     SETUP,   RETURN/TEST117B3,      ! EXEC SUBR WHICH:
6602     ! (1) D<11:06> -> IR<11:06>
6603     NEXT,    CALL(D11-06)           ! (2) BUT(INSTR5) INTO ZTARGET---
(4061) DCS(0.00.0.0.0.0) BM(0100..00.11..01.01..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.110)

6604
6605
6606
6607 ! CHECK BIT<05:00> = "10 1010"

```

```

6608 4657:
6609 TEST117B3:
6610
6611 PO, LOAD-ENUA(ZTARGET425), !INSTRS-EBB OUTPUT FOR BIT <05:00> = "101 010"
6612 LOAD-ERROR(TEST117B3), !ERROR DIRECTORY KEY
6613 DCS-CTR(C6.), !COMPARE AT TARGET
6614 NEXT, J/GOBUT117B3
(4657) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..010..101...0.0.0..0..0...0.0000...0..0000.0...11.000...000.110.010)

6615 4062: !(FREE)
6616 GOBUT117B3:
6617 SETUP, RETURN/TEST117C1, !EXEC SUBR WHICH:
6618 ! (1) D<05:00> -> IR<05:00>
6619 ! (2) BUT (INSTRS) INTO ZTARGET---
6620 NEXT, CALL(D05-00)
(4062) DCS(0.00.0.0.0.0) BM(0100..00.11..01.01..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.000)

6621
6622
6623
6624
6625
6626
6627
6628 ! - - - - -
6629
6630 !TESTS 117C 1-2 VERIFIES THAT WITH:
6631 !ALU=(A-AND-NOT-B), A=(000000), B=(000000), THEN D=(000000)
6632 4656:
6633 TEST117C1:
6634 PO, LOAD-ENUA(ZTARGET434), !INSTRS FOR IR=(000000)
6635 LOAD-ERROR(TEST117C1), !ERROR DIRECTORY KEY
6636 DCS-CTR(C6.), !COMPARE AT TARGET
6637 NEXT, J/ALU117C1
(4656) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...000.110.011)

6638 4063: !(FREE)
6639 ALU117C1:
6640 P2-T, D+A-AND-NOT-B, D(C)+CINMUX, !ALL=(A AND-NOT-B), D(C)=CINMUX=D(C)=(1)
6641 BUS-A+ASPHI(000000), !H=(000000)
6642 BUS-B+CSPD(000000), !B=(000000)
6643 NEXT, J/GOBUT117C1
(4063) DCS(0.00.0.0.0.0) BM(0111..10.00..11.01..100..000...0.1.0..0..0...0.0100...0..0000.0...11.000...000.110.100)

6645 4064: !(FREE)
6646 GOBUT117C1:
6647 SETUP, RETURN/TEST117C2, !EXEC SUBR WHICH:
6648 ! (1) D -> IR
6649 ! (2) BUT (INSTRS) INTO ZTARGET---
6650 NEXT, CALL(DZERO)
(4064) DCS(0.00.0.0.0.0) BM(0100..00.11..01.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.010)

6651
6652 !CHECK THAT D(C) GOT A (1) FROM CINMUX, ABOVE
6653 4677:
6654

```

```

6655 TEST117C2:
6656     PO,      LOAD-ENUA(ZTARGET417),      !BIT<02> = D[C] = (1)
6657     LOAD-ERROR(TEST117C2),      !ERROR DIRECTORY KEY
6658     DCS-CTR(C3.),      !COMPARE AT TARGET
6659     NEXT     J/GOBUT117C2
(4677) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..001..111...0.0.0..0..0...0.0000...0..0000.0...11.000...000.110.101)

6660
6661     4065:  !(FREE)
6662     GOBUT117C2:
6663     SETUP,   RETURN/SCOPE117C,      !RETURN TO SCOPE LOOP TEST WORD
6664     NEXT,    GOTO-PAGE(7),      !GOTO TABLE
6665     J/BUTD[C]C      !BIT<02> = D[C]H
(4065) DCS(0.00.0.0.0.0) BM(0100..00.00..01.10..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.000)

6666
6667
6668
6669     4066:  !(FREE)
6670     SCOPE117C:
6671     NEXT,    BUTD[SCOPE],      !NO ERROR: "TEST120A1" (+1. WORDS)
6672     J/TEST120A1      !ERROR: "ALU117A1" (-20. WORDS)
(4066) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.110.111)

6673
6674
6675     ! -----
6676
6677     !THIS NEXT SET OF 7. TESTS EXERCIZES THE ALU FUNCTION
6678     !"A-AND-B", AND THE CARRYOUT OF "CINMUX =(0) INTO D[C]
6679
6680     ! -----
6681
6682     !TESTS 120A 1-3 VERIFIES THAT WITH:
6683     !ALU=(A-AND-B), A=(125252), B=(177777), THEN D=(125252)
6684     4667:
6685     TEST120A1:
6686     PO,      LOAD-ENUA(ZTARGET412),      !BIT<15:12> = "1010"
6687     LOAD-ERROR(TEST120A1),      !ERROR DIRECTORY KEY
6688     DCS-CTR(C6.),      !COMPARE AT TARGET
6689     NEXT     J/ALU120A1
(4667) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...110.110.100)

6690
6691     4664:
6692     ALU120A1:
6693     PO,      BUMP-VERIFY,      !COUNT
6694     P2-T,    D+A-AND-B, D[C]+ALU15,      !ALU=(A-AND-B), D[C]=(1)
6695     BUS-A+ASPHI(C125252),      !A=(125252)
6696     BUS-B+CSPO(C177777),      !B=(177777)
6697     NEXT     J/GOBUT120A1      !D=(125252)
(4664) DCS(0.00.0.0.0.1) BM(1011. 10.00..11.01..110..100...0.1.0..0..0...0.0101...0..0000.0...11.000...000.110.111)

6698
6699     4067:  !(FREE)
6700     GOBUT120A1:
6701     SETUP,   RETURN/TEST120A2,      !EXEC SUBR WHICH:
6702     ! (1) D<15:12> -> IR<15:12>

```

```

6703      NEXT      CALL(D15-12)                !(2) BUT(IR15-12) INTO ZTARGET---
(4067) DCS(0.00.0.0.0.0) BM(0100..00.11..01.01..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.100)
6704
6705      !CHECK BIT <11:06> = "1010 10"
6706      4655:
6707      TEST120A2:
6708          PO,          LOAD-ENUA(ZTARGET412),          !INSTRS-E78 OUTPUT FOR BIT <11:06> = "101 010"
6709          LOAD-ERROR(TEST120A2),          !ERROR DIRECTORY KEY
6710          DCS-CTR(C6.),          !COMPARE AT TARGET
6711          BUMP-VERIFY,          !COUNT
6712          J/GOBUT120A2
(4655) DCS(1.00.1.0.0.1) BM(1001..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...000.111.000)
6713
6714      4070: !(FREE)
6715      GOBUT120A2:
6716          SETUP, RETURN/TEST120A3,          !EXEC SUBR WHICH:
6717          ! (1) D<11:06> -> IR<11:06>
6718      NEXT      CALL(D11-06)                ! (2) BUT(INSTRS) INTO ZTARGET---
(4070) DCS(0.00.0.0.0.0) BM(0100..00.11..01.01..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.110)
6719
6720
6721      !CHECK BIT<05:00> = "10 1010"
6722      4654:
6723      TEST120A3:
6724
6725          PO,          LOAD-ENUA(ZTARGET425),          !INSTRS-E88 OUTPUT FOR BIT<05:00>="101 010"
6726          LOAD-ERROR(TEST120A3),          !ERROR DIRECTORY KEY
6727          DCS-CTR(C6.),          !COMPARE AT TARGET
6728          J/GOBUT120A3
(4654) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..010..101...0.0.0..0..0...0.0000...0..0000.0...11.000...000.111.001)
6729
6730
6731      4071: !(FREE)
6732      GOBUT120A3:
6733          SETUP, RETURN/TEST120B1,          !EXEC SUBR WHICH:
6734          ! (1) D<05:00> -> IR<05:00>
6735      NEXT      CALL(D05-00)                ! (2) BUT(INSTRS) INTO ZTARGET---
(4071) DCS(0.00.0.0.0.0) BM(0100..00.11..01.01..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.000)
6736
6737
6738
6739
6740      ! - - - - -
6741
6742      !TESTS 120B 1-4 VERIFIES THAT WITH:
6743      !ALU=(A-AND-B), A=(052525), B=(177777), THEN D=(052525)
6744      4653:
6745      TEST120B1:
6746          PO,          LOAD-ENUA(ZTARGET405),          !BIT<15:12> = "0101"
6747          LOAD-ERROR(TEST120B1),          !ERROR DIRECTORY KEY
6748          DCS-CTR(C6.),          !COMPARE AT TARGET

```



```

6749      NEXT      J/ALU12081
(4653) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...000.111.0101)
6750
6751      4072:  !(FREE)
6752      ALU12081:
6753
6754      P2-T,      D=A-AND-B, D(C)=CINMUX,      !ALU=(A-AND-B), D(C)=CIN=(0)
6755      BUS-A=ASPHI(C052525),      !A=(052525)
6756      BUS-B=CSPD(C177777),      !B=(177777)
6757      NEXT      J/GOBUT12081      !D=(052525)
(4072) DCS(0.00.0.0.0.0) BM(1011..10.00..11.01..111..000...0.1.0..0..0...0.0101...0..0000.0...11.000...000.111.0111)
6758
6759      4073:  !(FREE)
6760      GOBUT12081:
6761      SETUP,      RETURN/TEST12082,      !EXEC SUBR WHICH:
6762      NEXT      CALL(D15-12)      ! (1) D<15:12> -> IR<15:12>
6763      (4073) DCS(0.00.0.0.0.0) BM(0100..00.11..01.01..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.1001)
6764      !CHECK BIT<11:06> = "0101 01"
6765      4652:
6766      TEST12082:
6767      PD,      LOAD-ENUA(ZTARGET405),      !INSTRS-E78 OUTPUT FOR BIT<11:06>="010 101"
6768      LOAD-ERROR(TEST12082),      !ERROR DIRECTORY KEY
6769      DCS-CTR(C6.),      !COMPARE AT TARGET
6770      BUMP-VERIFY,      !COUNT
6771      NEXT      J/GOBUT12082
(4652) DCS(1.00.1.0.0.1) BM(1001..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...000.111.1001)
6773
6774      4074:  !(FREE)
6775      GOBUT12082:
6776      SETUP,      RETURN/TEST12083,      !EXEC SUBR WHICH:
6777      NEXT      CALL(D11-06)      ! (1) D<11:06> -> IR<11:06>
6778      (4074) DCS(0.00.0.0.0.0) BM(0100..00.11..01.01..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.1101)
6779
6780      !CHECK BIT<05:00> = "01 0101"
6781      4651:
6782      TEST12083:
6783
6784      PD,      LOAD-ENUA(ZTARGET432),      !INSTRS-E88 OUTPUT FOR BIT <05:00>="010 101"
6785      LOAD-ERROR(TEST12083),      !ERROR DIRECTORY KEY
6786      DCS-CTR(C6.),      !COMPARE AT TARGET
6787      NEXT      J/GOBUT12083
(4651) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..011..010...0.0.0..0..0...0.0000...0..0000.0...11.000...000.111.1011)
6790
6791      4075:  !(FREE)
6792      GOBUT12083:
6793      SETUP,      RETURN/TEST12084,      !EXEC SUBR WHICH:
6794      (1) D<05:00> -> IR<05:00>

```

6795 NEXT CALL(D05-00) !(? ) BUT(INSTRS) INTO ZTARGET---  
 (4075) DCS[0.00.0.0.0.0] BM[0100..00.11..01.01..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.000]

6796  
 6797 !CHECK THAT D[C] GOT A (0) FROM "CINMUX" = (0)  
 6798 4650:

6799 TEST12084:  
 6800 PO, LOAD-ENUA(ZTARGET402), !BIT<00> = D[C] = (0)  
 6801 LOAD-ERROR(TEST12084), !ERROR DIRECTORY KEY  
 6802 DCS-CTR(C3.), !COMPARE AT TARGET  
 6803 BUMP-VERIFY, !COUNT  
 6804 J/GOBUT12084

(4650) DCS[1.00.1.0.0.1] BM[1100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...000.111.110]

6805  
 6806 4076: !(FREE)  
 6807 GOBUT12084:

6808 SETUP, RETURN/SCOPE1208, !RETURN TO SCOPE LOOP TEST WORD  
 6809 NEXT, GOTO-PAGE(7), !BUT TABLE  
 6810 J/BUTD[C]A !D[C]H IN BIT <00>

(4076) DCS[0.00.0.0.0.0] BM[0100..00.00..01.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.100]

6811  
 6812 4077: !(FREE)  
 6813 SCOPE1208:

6814 NEXT, BUTD[SCOPE], !NO ERROR: "TEST121A1" (+1. WORDS)  
 6815 J/TEST121A1 ! ERROR: "ALU120A1" (-15. WORDS)

(4077) DCS[0.00.0.1.0.0] BM[0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.110.101]

6816  
 6817  
 6818  
 6819  
 6820  
 6821 !-----  
 6822

6823 !THIS NEXT SET OF 16. TESTS EXERCIZES THE ALU FUNCTION  
 6824 !"A-XOR-B", AND THE CARRYOUT FUNCTIONS OF "ALU00" AND "ALU07" INTO D[C]  
 6825  
 6826 !-----  
 6827

6828  
 6829 !TESTS 121A 1-4 VERIFIES THAT WITH:  
 6830 !ALU=(A-XOR-B), A=(000000), B=(052525), THEN D=(052525)

6831 4665:  
 6832 TEST121A1:  
 6833 PO, LOAD-ENUA(ZTARGET405), !BIT<15:12> = "0101"  
 6834 LOAD-ERROR(TEST121A1), !ERROR DIRECTORY KEY  
 6835 DCS-CTR(C6.), !COMPARE AT TARGET  
 6836 J/ALU121A1

(4665) DCS[1.00.1.0.0.0] BM[1001..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...111.000.110]

6837  
 6838 4706:  
 6839 ALU121A1:  
 6840 PO, BUMP-VERIFY, !COUNT  
 6841 P2-T, D+A-XOR-B, D[C]+ALU00, !ALU=(A-XOR-B), D[C]=ALU00=(1)

```

6842          BUS-A+ASPHI(C000000),          !A=(000000)
6843          BUS-B+CSPD(C052525),          !B=(052525)
6844          NEXT, J/GOBUT121A1           !D=(052525)
(4706) DCS(0.00.0.0.0.0.1) BM(0110..10.00..11.01..100..010...0.1.0..0..0...0.0111...0..0000.0...11.000...001.000.011)

6845          4103: !(FREE)
6846          GOBUT121A1:
6847          SETUP, RETURN/TEST121A2,      !EXEC SUBR WHICH:
6848          !!(1) D<15:12> -> IR<15:12>
6849          !!(2) BUT(IR15-12) INTO ZTARGET---
6850          NEXT, CALL(015-12)
(4103) DCS(0.00.0.0.0.0.0) BM(0100..00.11..01.11..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.100)

6851          !CHECK BIT<11:06> = "0101 01"
6852          4676:
6853          TEST121A2:
6854          PO,
6855          LOAD-ENUA(ZTARGET405),          !INSTR5-E78 OUTPUT FOR BIT<11:06>="010 101"
6856          LOAD-ERROR(TEST121A2),        !ERROR DIRECTORY KEY
6857          DCS-CTR(C6.),                  !COMPARE AT TARGET
6858          BUMP-VERIFY,                   !COUNT
6859          NEXT, J/GOBUT121A2
(467) DCS(1.00.1.0.0.0.1) BM(1001..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...001.000.100)

6860          4104: !(FREE)
6861          GOBUT121A2:
6862          SETUP, RETURN/TEST121A3,      !EXEC SUBR WHICH:
6863          !!(1) D<11:06> -> IR<11:06>
6864          !!(2) BUT(INSTR5) INTO ZTARGET---
6865          NEXT, CALL(011-06)
(4104) DCS(0.00.0.0.0.0.0) BM(0100..00.11..01.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.110)

6866          !CHECK BIT<05:00> = "01 0101"
6867          4675:
6868          TEST121A3:
6869          PO,
6870          LOAD-ENUA(ZTARGET432),          !INSTR5-E88 OUTPUT FOR BIT<05:00>="010 101"
6871          LOAD-ERROR(TEST121A3),        !ERROR DIRECTORY KEY
6872          DCS-CTR(C6.),                  !COMPARE AT TARGET
6873          NEXT, J/GOBUT121A3
(4675) DCS(1.00.1.0.0.0.0) BM(1001..00.11..11.00..011..010...0.0.0..0..0...0.0000...0..0000.0...11.000...001.000.101)

6876          4105: !(FREE)
6877          GOBUT121A3:
6878          SETUP, RETURN/TEST121A4,      !EXEC SUBR WHICH:
6879          !!(1) D<05:00> -> IR<05:00>
6880          !!(2) BUT(INSTR5) INTO ZTARGET---
6881          NEXT, CALL(005-00)
(4105) DCS(0.00.0.0.0.0.0) BM(0100..00.11..01.11..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.000)

6882          !CHECK THAT D(C) GOT A (1) FROM "ALUD0"
6883          4674:
6884          TEST121A4:
6885          PO,
6886          LOAD-ENUA(ZTARGET403),          !BIT<01> = D(C) = (1)

```

```

6887          LOAD-ERROR(TEST121A4),          !ERROR DIRECTORY KEY
6888          DCS-CTR(C3.),                    !COMPARE AT TARGET
6889          BUMP-VERIFY,                      !COUNT
6890          NEXT, J/GOBUT121A4
(4674) DCS(1.00.1.0.0.1) BM(1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...001.000.110)

6891          4106: !(FREE)
6892          GOBUT121A4:
6893          SETUP, RETURN/TEST121B1,          !RETURN TO START OF NEXT SUBTEST
6894          NEXT, GOTO-PAGE(7),              !BUT TABLE
6895          J/BUTD(C)B                         !D(C)H IN BIT<01>
6896          (4106) DCS(0.00.0.0.0.0) BM(0100..00.11..01.10..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.101.000)

6897
6898
6899
6900
6901          ! - - - - -
6902
6903          !TESTS 121B 1-4 VERIFIES THAT WITH:
6904          !ALU=(A-XOR-B), A=(177777), B=(052525), THEN D=(125252)
6905          4663:
6906          TEST121B1:
6907          PO, LOAD-ENUA(ZTARGET412),          !BIT<15:12> = "1010"
6908          LOAD-ERROR(TEST121B1),            !ERROR DIRECTORY KEY
6909          DCS-CTR(C6.),                      !COMPARE AT TARGET
6910          NEXT, J/ALU121B1
(4663) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...001.000.111)

6911          4107: !(FREE)
6912          ALU121B1:
6913          PO, BUMP-VERIFY,                  !COUNT
6914          P2-T, D+A-XOR-B, D(C)+ALU00,       !ALU=(A-XOR-B), D(C)=ALU00=(0)
6915          BUS-A+ASPHI(C177777),            !A=(177777)
6916          BUS-B+CSPO(C052525),             !B=(052525)
6917          NEXT, J/GOBUT121B1                !D=(125252)
6918          (4107) DCS(0.00.0.0.0.1) BM(0110..10.00..11.01..101..010...0.1.0..0..0...0.0111...0..0000.0...11.000...001.001.000)

6919
6920          4110: !(FREE)
6921          GOBUT121B1:
6922          SETUP, RETURN/TEST121B2,          !EXEC SUBR WHICH:
6923          NEXT, CALL(D15-12)                 ! (1) D<15:12> -> IR<15:12>
6924          (4110) DCS(0.00.0.0.0.0) BM(0100..00.11..01.10..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.100)

6925          !CHECK BIT<11:06> = "1010 10"
6926          4662:
6927          TEST121B2:
6928          PO, LOAD-ENUA(ZTARGET412),          !INSTR5-E78 OUTPUT FOR BIT<11:06>="101 010"
6929          LOAD-ERROR(TEST121B2),            !ERROR DIRECTORY KEY
6930          DCS-CTR(C6.),                      !COMPARE AT TARGET
6931          BUMP-VERIFY,                      !COUNT
6932          NEXT, J/GOBUT121B2
6933

```

```

(4662) DCS(1.00.1.0.0.1) BM(1001..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...001.001.001)
6934
6935 4111: !(FREE)
6936 GOBUT12182:
6937     SETUP, RETURN/TEST12183,
6938
6939     NEXT, CALL(D11-06)
(4111) DCS(0.00.0.0.0.0) BM(0100..00.11..01.10..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.110)
6940
6941
6942
6943 !CHECK BIT<05:00> = "10 1010"
6944 4661:
6945 TEST12183:
6946
6947     PO,     LOAD-ENVA(ZTARGET425),
6948             LOAD-ERROR(TEST12183),
6949             DCS-CTR(C6.),
6950             NEXT, J/GOBUT12183
(4661) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..010..101...0.0.0..0..0...0.0000...0..0000.0...11.000...001.001.010)
6951
6952 4112: !(FREE)
6953 GOBUT12183:
6954     SETUP, RETURN/TEST12184,
6955
6956     NEXT, CALL(D05-00)
(4112) DCS(0.00.0.0.0.0) BM(0100..00.11..01.10..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.000)
6957
6958 !CHECK THAT D(C) GOT A (0) FROM "ALU00"
6959 4660:
6960 TEST12184:
6961     PO,     LOAD-ENVA(ZTARGET401),
6962             LOAD-ERROR(TEST12184),
6963             DCS-CTR(C3.),
6964             BUMP-VERIFY,
6965             NEXT, J/GOBUT12184
(4660) DCS(1.00.1.0.0.1) BM(1100..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...001.001.011)
6966
6967 4113: !(FREE)
6968 GOBUT12184:
6969     SETUP, RETURN/SCOPE121C,
6970     NEXT,  GOTO-PAGE(7),
6971             J/BUTD(C)B
(4113) DCS(0.00.0.0.0.0) BM(0100..00.00..10.01..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.101.000)
6972
6973
6974
6975 4114: !(FREE)
6976 SCOPE121C:
6977     NEXT,  BUTD(SCOPE),
6978             J/TEST121C1
!EXEC SUBR WHICH:
!(1) D<11:06> -> IR<11:06>
!(2) BUT(INSTRS) INTO ZTARGET---
!INSTRS-E88 OUTPUT FOR BIT<05:00>="101 010"
!ERROR DIRECTORY KEY
!COMPARE AT TARGET
!EXEC SUBR WHICH:
!(1) D<05:00> -> IR<05:00>
!(2) BUT(INSTRS) INTO ZTARGET---
!BIT<01> = D(C) = (0)
!ERROR DIRECTORY KEY
!COMPARE AT TARGET
!COUNT
!RETURN TO SCOPE LOOP TEST WORD
!BUT TABLE
!D(C)H IN BIT<01>
!NO ERROR: "TEST121C1" (+1. WORDS)
!ERROR: "ALU121A1" (-17. WORDS)

```

(4114) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...111.000.111)

6979  
6980  
6981  
6982  
6983  
6984  
6985  
6986  
6987  
6988  
6989  
6990  
6991

! TESTS 121C 1-4 VERIFIES THAT WITH:  
! ALU=(A-XOR-B), A=(000000), B=(125252), THEN D=(125252)

4707:  
TEST121C1:  
PO, LOAD-ENUA(ZTARGET412), !BIT<15:12> = "1010"  
LOAD-ERROR(TEST121C1), !ERROR DIRECTORY KEY  
DCS-CTR(C6.), !COMPARE AT TARGET  
NEXT, J/ALU121C1

(4707) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...110.111.000)

6992  
6993  
6994  
6995  
6996  
6997  
6998  
6999

4670:  
ALU121C1:  
PO, BUMP-VERIFY, !COUNT  
P2-T, D+A-XOR-B, D(C)+ALU07, !ALU=(A-XOR-B), D(C)=ALU07=(1)  
BUS-A+ASPHI(C000000), !A=(000000)  
BUS-B+CSPD(C125252), !B=(125252)  
NEXT, J/GOBUT121C1 !D=(125252)

(4670) DCS(0.00.0.0.0.1) BM(0110..10.00..11.01..100..011...0.1.0..0..0...0.0110...0..0000.0...11.000...001.001.101)

7000  
7001  
7002  
7003  
7004  
7005

4115: !(FREE)  
GOBUT121C1:  
SETUP, RETURN/TEST121C2, !EXEC SUBR WHICH:  
NEXT, CALL(D15-12) ! (1) D<15:12> -> IR<15:12>  
! (2) BUT(IR15-12) INTO ZTARGET---

(4115) DCS(0.00.0.0.0.0) BM(0100..00.11..00.00..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.100)

7006  
7007  
7008  
7009  
7010  
7011  
7012  
7013  
7014

!CHECK BIT<11:06> = "1010 10"  
4605:  
TEST121C2:  
PO, LOAD-ENUA(ZTARGET412), !INSTR5-E78 OUTPUT FOR BIT<11:06>="101 010"  
LOAD-ERROR(TEST121C2), !ERROR DIRECTORY KEY  
DCS-CTR(C6.), !COMPARE AT TARGET  
BUMP-VERIFY, !COUNT  
NEXT, J/GOBUT121C2

(4605) DCS(1.00.1.0.0.1) BM(1001..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...001.001.110)

7015  
7016  
7017  
7018  
7019  
7020

4116: !(FREE)  
GOBUT121C2:  
SETUP, RETURN/TEST121C3, !EXEC SUBR WHICH:  
NEXT, CALL(D11-06) ! (1) D<11:06> -> IR<11:06>  
! (2) BUT(INSTR5) INTO ZTARGET---

(4116) DCS(0.00.0.0.0.0) BM(0100..00.11..10.01..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.110)

7021  
7022  
7023  
7024

!CHECK BIT<05:00> = "10 1010"  
4717:



```

7072          BUS-B+CSPD(C125252),          !B=(125252)
7073          NEXT      J/GOBUT121D1        !D=(052525)
(4121) DCS(0.00.0.0.0.1) BM(0110..10.00..11.01..101..011...0.1.0..0..0...0.0110...0..0000.0...11.000...001.010.010)

7074          4122: !(FREE)
7075          GOBUT121D1:
7076          SETUP,   RETURN/TEST121D2,          !EXEC SUBR WHICH:
7077          !          !          !          !          !          !
7078          !          !          !          !          !          !
7079          !          !          !          !          !          !
          NEXT      CALL(D15-12)          !          !
(4122) DCS(0.00.0.0.0.0) BM(0100..00.11..10.10..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.100)

7080          !CHECK BIT<11:06> = "0101 01"
7081          4726:
7082          TEST121D2:
7083          PO,      LOAD-ENUA(ZTARGET405),          !INSTR5-E78 OUTPUT FOR BIT<11:06>="010 101"
7084          !          LOAD-ERROR(TEST121D2),          !ERROR DIRECTORY KEY
7085          !          DCS-CTR(C6.),          !COMPARE AT TARGET
7086          !          BUMP-VERIFY,          !COUNT
7087          !          NEXT      J/GOBUT121D2
7088          !          !          !          !          !          !
(4726) DCS(1.00.1.0.0.1) BM(1001..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...001.010.011)

7089          4123: !(FREE)
7090          GOBUT121D2:
7091          SETUP,   RETURN/TEST121D3,          !EXEC SUBR WHICH:
7092          !          !          !          !          !          !
7093          !          !          !          !          !          !
7094          !          !          !          !          !          !
          NEXT      CALL(D11-06)          !          !
(4123) DCS(0.00.0.0.0.0) BM(0100..00.11..10.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.110)

7095          !CHECK BIT<05:00> = "01 0101"
7096          4737:
7097          TEST121D3:
7098          PO,      LOAD-ENUA(ZTARGET432),          !INSTR5-E88 OUTPUT FOR BIT<05:00>="010 101"
7099          !          LOAD-ERROR(TEST121D3),          !ERROR DIRECTORY KEY
7100          !          DCS-CTR(C6.),          !COMPARE AT TARGET
7101          !          NEXT      J/GOBUT121D3
7102          !          !          !          !          !          !
(4737) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..011..010...0.0.0..0..0...0.0000...0..0000.0...11.000...001.010.100)

7103          4124: !(FREE)
7104          GOBUT121D3:
7105          SETUP,   RETURN/TEST121D4,          !EXEC SUBR WHICH:
7106          !          !          !          !          !          !
7107          !          !          !          !          !          !
7108          !          !          !          !          !          !
          NEXT      CALL(D05-00)          !          !
(4124) DCS(0.00.0.0.0.0) BM(0100..00.11..10.11..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.000)

7109          !CHECK THAT D(C) GOT A (0) FROM "ALU07"
7110          4736:
7111          TEST121D4:
7112          PO,      LOAD-ENUA(ZTARGET413),          !BIT<02> = D(C) = (0)
7113          !          !          !          !          !          !
7114          !          !          !          !          !          !
7115          !          !          !          !          !          !
7116          !          !          !          !          !          !

```



```

7117          LOAD-ERROR(TEST121D4),          !ERROR DIRECTORY KEY
7118          DCS-CTR(C3.),                    !COMPARE AT TARGET
7119          BUMP-VERIFY,                      !COUNT
7120          NEXT, J/GOBUT121D4
(4736) DCS(1.00.1.0.0.1) BM(1100..00.11..11.00..001..011...0.0.0..0..0...0.0000...0..0000.0...11.000...001.010.101)

7121          4125: !(FREE)
7122          GOBUT121D4:
7123          SETUP, RETURN/SCOPE121D,          !RETURN TO SCOPE LOOP TEST WORD
7124          NEXT, GOTO-PAGE(7),              !BUT TABLE
7125          J/BUTD(C)                          !D(C)H IN BIT<02>
7126          (4125) DCS(0.00.0.0.0.0) BM(0100..00.00..10.10..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.000)

7127          4126: !(FREE)
7128          SCOPE121D:
7129          PO, BUMP-VERIFY,                  !COUNT
7130          NEXT, BUTD(SCOPE),                !NO ERROR: "TEST122A1" (+1. WORDS)
7131          J/TEST122A1                          !ERROR: "ALU121C1" (-17. WORDS)
7132          (4126) DCS(0.00.0.1.0.1) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.111.001)

7133
7134
7135
7136
7137
7138
7139
7140
7141          ! -----
7142          ! THIS NEXT SET OF 8. TESTS CHECK THE ALU FUNCTIONS "A"
7143          ! AND "A-IOR-B" THE FUNCTION D(C)=CINMUX=(1) AND ALSO
7144          ! THE BUT(D<14:00>=ZERO#D<15>) BUT FOR D=(0)(00000) AND
7145          ! D=(1)(25252)
7146          4671:
7147          TEST122A1:
7148          PO, LOAD-ENVA(ZTARGET434),        !INSTRS FOR IR=(000000)
7149          LOAD-ERROR(TEST122A1),          !ERROR DIRECTORY KEY
7150          DCS-CTR(C6.),                    !COMPARE AT TARGET
7151          NEXT, J/ALU122A1
7152          (4671) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...110.111.010)

7153          4672:
7154          ALU122A1:
7155          PO, BUMP-VERIFY,                  !COUNT
7156          P2-T, D+A-IOR-B, D(C)+CINMUX,     !ALU=(A-IOR-B), D(C)=CINMUX=(1)
7157          BUS-A+ASPHI(000000),            !A=(000000)
7158          BUS-B+BSPHI(000000),            !B=(000000)
7159          NEXT, J/GOBUT122A1              !D=(000000)
7160          (4672) DCS(0.00.0.0.0.1) BM(1110..01.11..11.01..100..000...0.1.0..0..0...0.0000...0..0000.0...11.000...001.010.111)

7161          4127: !(FREE)
7162          GOBUT122A1:
7163          SETUP, RETURN/TEST122A2,          !EXEC SUBR WHICH:
7164

```

```

7165                                     !(1) D -> IR
7166      NEXT, CALL(DZERO)                !(2) BUT(INSTR5) INTO ZTARGET---
(4127) DCS(0.00.0.0.0.0) BM(0100..00.10..11.11..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.010)
7167
7168
7169
7170      !CHECK THAT D(C) GOT A (1) FROM CINMUX=(1)
7171      4570:
7172      TEST122A2:
7173      PO,          LOAD-ENVA(ZTARGET403),      !BIT<00> = D(C) = (1)
7174                  LOAD-ERROR(TEST122A2),      !ERROR DIRECTORY KEY
7175                  DCS-CTR(C3.),                !COMPARE AT TARGET
7176                  J/GOBUT122A2
(4570) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...001.011.000)
7177
7178      4130: !(FREE)
7179      GOBUT122A2:
7180      SETUP,      RETURN/TEST122A3,           !RETURN TO START OF NEXT SUBTEST
7181      NEXT,       GOTO-PAGE(7),              !BUT TABLE
7182                  J/BUTD(C)A                !D(C)H IN BIT <00>
(4130) DCS(0.00.0.0.0.0) BM(0100..00.11..10.11..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.100)
7183
7184
7185
7186      !CHECK THAT D<15>=0, [D<14:00>=ZERO]=1 WHEN D=(000000)
7187      4734:
7188      TEST122A3:
7189      PO,          LOAD-ENVA(ZTARGET402),      !SETUP FOR D<15:00>=ZERO
7190                  LOAD-ERROR(TEST122A3),      !ERROR DIRECTORY KEY
7191                  DCS-CTR(C3.),                !COMPARE AT TARGET
7192                  J/GOBUT122A3
(4734) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...001.011.001)
7193
7194      4131: !(FREE)
7195      GOBUT122A3:
7196      SETUP,      RETURN/TEST122A4,           !RETURN TO START OF NEXT SUBTEST
7197      NEXT,       GOTO-PAGE(7),              !BUT TABLE
7198                  J/BUTD-15-ZERO            !BIT<1:0> = D<15># D<14:00>=ZERO
(4131) DCS(0.00.0.0.0.0) BM(0100..00.11..10.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)
7199
7200      !CHECK THAT D<15>=(1), [D<14:00>=ZERO]=(0) WHEN D=(125252)
7201      4735:
7202      TEST122A4:
7203      PO,          LOAD-ENVA(ZTARGET401),      !SETUP FOR D<15:00>=(125252)
7204                  LOAD-ERROR(TEST122A4),      !ERROR DIRECTORY KEY
7205                  DCS-CTR(C4.),                !COMPARE AT TARGET
7206                  BUMP-VERIFY,                 !COUNT
7207                  J/SETD122A4
(4735) DCS(1.00.1.0.0.1) BM(1011..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...001.011.010)
7208
7209      4132: !(FREE)

```

```

7210 SETD122A4:
7211 P2-T, D+CSPD(C125252), D(C)+0, !SETUP D FOR TEST
7212 NEXT, J/GOBUT122A4
(4132) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...0.1.0..0..0...0.0110...0..0000.0...11.000...001.011.011)

7213
7214 4133: !(FREE)
7215 GOBUT122A4:
7216 SETUP, RETURN/SCOPE122A, !RETURN TO SCOPE LOOP TEST WORD
7217 NEXT, GOTO-PAGE(7) !BUT TABLE
7218 J/BUTD-IS-ZERO !BIT<1:0> = D15*(D<14:00>=ZERO)
(4133) DCS(0.00.0.0.0.0) BM(0100..00.00..10.11..100. 111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

7219
7220
7221
7222 4134: !(FREE)
7223 SCOPE122A:
7224 PD, BUSDIN+EMIT-[I], !RESET PROC UCON
7225 EN-CLK-IR[15-00],
7226 NEXT, BUTD[SCOPE], !NO ERROR: "TEST130A1" (+1. WORDS)
7227 J/TEST130A1 !ERROR: "ALU122A1" (-8. WORDS)
(4134) DCS(0.00.0.1.0.0) BM(0000..00.00..00.01..000..100...0.0.0..0..0...1.1001...0..0000.0...11.000...110.111.011)

```

7228  
7229  
7230  
7231  
7232  
7233 !.PAGE=====

7236 .TOC \* TEST130-136: ALU ARITHMETIC FUNCTION/CARRY LOOKAHEAD TESTS

```

7238 !*****
7239 !*
7240 !* TESTS: 130 - 136 UWORDS: 127 + 160
7241 !*
7242 !* FUNCTIONS:
7243 !*
7244 !* ALU ARITHMETIC FUNCTION DECODE, INTERNAL CARRIES, CARRYOUTS, CARRY LOOKAHEAD.
7245 !*
7246 !*****

```

7248 :  
7249 SUMMARY OF ALU ARITHMETIC / CARRY LOOKAHEAD TESTS:

TEST NUMB	OPERANDS EMPLOYED: (A/B)+(B/A)+(CIN)=(COUT)(D)	ALU FUNCTION
130A	(0101)+(0101)+(0)=(0)(1010)	A-PLUS-B-PLUS-0
130B	(1010)+(1010)+(1)=(1)(0101)	A-PLUS-B-PLUS-1
131A	(1010)+(0101)+(0)=(0)(1111)	A-PLUS-B-PLUS-PS(C)
131B	(0101)+(1010)+(0)=(0)(1111)	DIVIDE/D(C)=0/A-PLUS-B-PLUS-0

KD11-K

MICRO V00A-1 00:00:03 12-MAR-77

```

7260
7261      132A      (1010)-(1010)-(0)=(1)(0000)      A-MINUS-B-MINUS-0
7262      132B      (0101)-(0101)-(0)=(1)(0000)      DIVIDE/D(C)=1/A-MINUS-B-MINUS-0
7263
7264      133A      (1000)+(1000)+(0)=(1)(0000)      A-PLUS-B-PLUS-D(C)
7265      133B      (0111)+(0111)+(1)=(0)(1111)      A-PLUS-NOT-B-PLUS-D(C)
7266
7267      134A      (0100)+(1100)+(0)=(1)(0000)      A-PLUS-B-PLUS-D(C)
7268      134B      (1011)+(0011)+(1)=(0)(1111)      A-PLUS-NOT-B-PLUS-D(C)
7269
7270      135A      (1010)+(0110)+(0)=(1)(0000)      A-PLUS-B-PLUS-D(C)
7271      135B      (0101)+(1001)+(1)=(0)(1111)      A-PLUS-NOT-B-PLUS-D(C)
7272
7273      136A      (0101)+(1011)+(0)=(1)(0000)      A-PLUS-B-PLUS-D(C)
7274      136B      (1010)+(0100)+(1)=(0)(1111)      A-PLUS-NOT-B-PLUS-D(C)
7275
7276
7277
7278
7279      !CHECK INTERNAL ALU CARRIES WITH:      (052525)+(052525)+(0)=(125252)
7280      !ALSO CHECK ALU FUNCTION "A-PLUS-B-PLUS-0", D(C)+COUT15=(0)
7281
7282      4673:
7283      TEST130A1:
7284      PO,      LOAD-ENL(A(ZTARGET434),      !SETUP FOR IR=(000000)/BUTINSTRS TEST
7285      LOAD-ERROR(TEST130A1),      !ERROR DIRECTORY KEY
7286      DCS-CTR(C8.),      !COMPARE AT TARGET
7287      NEXT,      J/ARITH130A1
(4673) DCS(1.00.1.0.0.0) BM(0111..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...101.100.000)

7288
7289      4540:
7290      ARITH130A1:
7291      P2-T,      D+A-PLUS-B-PLUS-0,      !ALU=(A-PLUS-B), CIN=(0)
7292      D(C)+COUT15,      !GET CARRYOUT
7293      BUS-A+ASPHI(C052525),      !A=(052525)
7294      BUS-B+BSPHI(C052525),      !B=(052525)
7295      SR+A-PLUS-B-PLUS-0,      !D=(125252), COUT15=(0)
7296      NEXT,      J/COMP130A1
(4540) DCS(0.00.0.0.0.0) BM(1001..01.11..11.01..111..110...0.1.1..0..0...0.0000...0..0000.0...11.000...001.011.101)

7297
7298      4135:      !(FREE)
7299      COMP130A1:
7300      PO,      BUMP-VERIFY,      !COUNT
7301      P2-T,      D+SR-XOR-CSPD(C125252),      !COMPARE RECEIVED: EXPECTED
7302      SAVE-D(C),      !SAVE CARRY
7303      NEXT,      J/GOBUT130A1
(4135) DCS(0.00.0.0.0.0) BM(0110..10.00..00.00..000..111...0.1.0..0..0...0.0110...0..0000.0...11.000...001.011.110)

7304
7305      4136:      !(FREE)
7306      GOBUT130A1:
7307      SETUP,      RETURN/TEST130A2,      !RETURN TO START OF NEXT SUBTEST
7308      NEXT,      CALL(JINTOIR-5)      !GO PUT D -> IR, BUT(INSTR5)
(4136) DCS(0.00.0.0.0.0) BM(0100..00.10..11.01..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

```

```

7309
7310
7311
7312 !CHECK THAT CARRYOUT OF BIT15 (COUT15) WAS CORRECT
7313 4550:
7314 TEST130A2:
7315     PO,          LOAD-ENUA(ZTARGET402),          !BIT<00> CLEAR
7316                 LOAD-ERROR(TEST130A2),          !ERROR DIRECTORY KEY
7317                 DCS-CTR(C3.),                  !COMPARE AT TARGET
7318     NEXT,        J/GOBUT130A2
(4550) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...001.011.111)

7319
7320 4137: !(FREE)
7321 GOBUT130A2:
7322     SETUP,       RETURN/TEST130B1,              !RETURN TO START OF NEXT SUBTEST
7323     NEXT,        GOTO-PAGE(7),                  !BUT TABLE
7324                 J/BUTD(C)A                      !D(C)H= COUT15 H IN BIT<00>
(4137) DCS(0.00.0.0.0.0) BM(0100..00.10..10.11..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.100)

7325
7326
7327
7328 ! - - - - -
7329
7330 !CHECK INTERNAL ALU CARRIES WITH: (125252)+(125252)+(1)=(052525)
7331 !ALSO CHECK ALU FUNCTION "A-PLUS-B-PLUS-1", D(C)+COUT15=(1)
7332
7333 4530:
7334 TEST130B1:
7335     PO,          LOAD-ENUA(ZTARGET434),          !SETUP FOR IR=(000000)/BUTINSTRS TEST
7336                 LOAD-ERROR(TEST130B1),          !ERROR DIRECTORY KEY
7337                 DCS-CTR(C8.),                  !COMPARE AT TARGET
7338     NEXT,        J/ARITH130B1
(4530) DCS(1.00.1.0.0.0) BM(0111 00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...001.100.000)

7339
7340 4140: !(FREE)
7341 ARITH130B1:
7342     P2-T,        D+A-PLUS-B-PLUS-1,              !ALU=(A-PLUS-B), CIN=(1)
7343                 D(C)+COUT15,                  !GET CARRYOUT
7344                 BUS-A+ASPHI(C125252),          !A=(125252)
7345                 BUS-B+BSPHI(C125252),          !B=(125252)
7346                 SR+A-PLUS-B-PLUS-1,          !D=(052525), COUT15=(1)
7347     NEXT,        J/COMP130B1
(4140) DCS(0.00.0.0.0.0) BM(1100..01.11..11.01..110..110...0.1.1..0..0...0.0000...0..0000.0...11.000...001.100.001)

7348
7349 4141: !(FREE)
7350 COMP130B1:
7351     PO,          BUMP-VERIFY,                    !COUNT
7352     P2-T,        D+SR-XOR-CSPD(C052525),        !COMPARE RECEIVED: EXPECTED
7353                 SAVE-D(C),                      !SAVE CARRY
7354     NEXT,        J/GOBUT130B1
(4141) DCS(0.00.0.0.0.1) BM(0110. 10.00..00.00..000..111. .0.1.0..0..0...0.0111...0..0000.0...11.000...001.100.010)

7355
7356 4142: !(FREE)

```

```

7357 GOBUT13081:
7358     SETUP, RETURN/TEST13082,           !RETURN TO START OF NEXT SUBTEST
7359     NEXT,  CALL(DINTOIR-5)             !GO PUT D -> IR, BUT(INSTRS)
(4142) DCS(0.00.0.0.0.0) BM(0100..00.10..10.11..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

```

```

7360
7361
7362
7363 !CHECK THAT CARRYOUT OF BIT15 (COUT15) WAS CORRECT
7364 4532:
7365 TEST13082:

```

```

7366     PO,      LOAD-ENVA(ZTARGET403),      !BIT<00> SET
7367     LOAD-ERROR(TEST13082),              !ERROR DIRECTORY KEY
7368     DCS-CTR(C3.),                       !COMPARE AT TARGET
7369     NEXT,    J/GOBUT13082
(4532) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...001.100.011)

```

```

7370
7371 4143: !(FREE)
7372 GOBUT13082:
7373     SETUP, RETURN/SCOPE1308,           !RETURN TO SCOPE LOOP TEST WORD
7374     NEXT,  GOTO-PAGE(7),              !BUT TABLE
7375     J/BUTD(C)A                          !D(C)H= COUT15 H IN BIT<00>

```

```

(4143) DCS(0.00.0.0.0.0) BM(0100..00.00..11.00..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.100)

```

```

7376
7377
7378
7379 4144: !(FREE)
7380 SCOPE1308:

```

```

7381     NEXT,    BUTD(SCOPE),              !NO ERROR: "TEST131A1" (+1.WORDS)
7382     J/TEST131A1                        !ERROR: "ARITH130A1" (-11.WORDS)
(4144) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...101.100.001)

```

```

7383
7384
7385
7386 ! - - - - -
7387

```

```

7388 !CHECK INTERNAL ALU CARRIES WITH: (125252)+(052525)+(0)=(177777)
7389 !ALSO CHECK ALU FUNCTION "A-PLUS-B-PLUS-PS(C)", D(C)+COUT15=(0)
7390 !CIN=PS(C)=(0) FROM INITIALIZATION ROUTINE
7391 4541:

```

```

7392 TEST131A1:
7393     PO,      LOAD-ENVA(ZTARGET434),      !SETUP FOR IR=(000000)/BUTINSTRS TEST
7394     LOAD-ERROR(TEST131A1),              !ERROR DIRECTORY KEY
7395     DCS-CTR(C8.),                       !COMPARE AT TARGET
7396     NEXT,    J/ARITH131A1

```

```

(4541) DCS(1.00.1.0.0.0) BM(0111..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...101.010.000)

```

```

7397
7398 4520:
7399 ARITH131A1:
7400     P2-T,    D+A-PLUS-B-PLUS-PS(C),      !ALU=(A-PLUS-B), CIN=PS(C)=(0)
7401     D(C)+COUT15,                          !GET CARRYOUT
7402     BUS-A+ASPHI(C125252),                 !A=(125252)
7403     BUS-B+CSPD(C052525),                  !B=(052525)

```

```

7404          SR+A-PLUS-B-PLUS-PS(C),          !D=(177777), COUT15=(0)
7405          NEXT      J/COMP131A1
(4520) DCS(0.00.0.0.0.0) BM(0001..10.00..11.01..110..110...0.1.1..0..0...0.0111...0..0000.0...11.000...001.100.101)

7406          4145:  !(FREE)
7407          COMP131A1:
7408          PO,          BUMP-VERIFY,          !COUNT
7409          P2-T,        D+SR-XOR-CSPD(C177777), !COMPARE RECEIVED: EXPECTED
7410          SAVE-D(C),   !SAVE CARRY
7411          NEXT      J/GOBUT131A1
7412          (4145) DCS(0.00.0.0.0.1) BM(0110..10.00..00.00..000..111...0.1.0..0..0...0.0101...0..0000.0...11.000...001.100.110)

7413          4146:  !(FREE)
7414          GOBUT131A1:
7415          SETUP,      RETURN/TEST131A2,      !RETURN TO START OF NEXT SUBTEST
7416          NEXT      CALL(DINTOIR-5)         !GO PUT D -> IR, BUT(INSTR5)
7417          (4146) DCS(0.00.0.0.0.0) BM(0100..00.10..11.00..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

7418          !CHECK THAT CARRYOUT OF BIT15 (COUT15) WAS CORRECT
7419          4544:
7420          TEST131A2:
7421          PO,          LOAD-ENUA(ZTARGET401), !BIT<01> CLEAR
7422          LOAD-ERROR(TEST131A2), !ERROR DIRECTORY KEY
7423          DCS-CTR(C3.), !COMPARE AT TARGET
7424          NEXT      J/GOBUT131A2
7425          (4544) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...001.100.111)

7426          4147:  !(FREE)
7427          GOBUT131A2:
7428          SETUP,      RETURN/TEST131B1,      !RETURN TO START OF NEXT SUBTEST
7429          NEXT      GOTO-PAGE(7),           !BUT TABLE
7430          J/BUTD(C)B   !D(C)H= COUT15 H IN BIT<01>
7431          (4147) DCS(0.00.0.0.0.0) BM(0100..00.10..11.00..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...C11.101.000)

7432          ! - - - - -
7433          !CHECK INTERNAL ALU CARRIES WITH:          (052525)+(125252)+(0)=(177777)
7434          !ALSO CHECK ALU FUNCTION "DIVIDE-STEP" = "A-PLUS-B" SINCE D(C)H=(0)
7435          !FROM ABOVE. D(C)+COUT15=(0)
7436          4546:
7437          TEST131B1:
7438          PO,          LOAD-ENUA(ZTARGET434), !SETUP FOR IR=(000000)/BUTINSTRS TEST
7439          LOAD-ERROR(TEST131B1), !ERROR DIRECTORY KEY
7440          DCS-CTR(C8.), !COMPARE AT TARGET
7441          NEXT      J/ARITH131B1
7442          (4546) DCS(1.00.1.0.0.0) BM(0111..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...001.101.000)

7443          4150:  !(FREE)

```

KD11-K

MICRO

VOOA-1 00:00:03 12-MAR-77

```

7450 ARITH131B1:
7451     P2-T,    D+DIVIDE-STEP,          !ALU=(A-PLUS-B), CIN=(0)
7452     D(C)+COUT15,          !GET CARRYOUT
7453     BUS-A+ASPHI(C052525),    !A=(052525)
7454     BUS-B+CSPD(C125252),    !B=(125252)
7455     SR+DIVIDE-STEP,          !D=(177777), COUT15=(0)
7456     NEXT,    J/COMP131B1
(4150) DCS(0.00.0.0.0.0) BM(1000..10.00..11.01..111..110...0.1.1..0..0...0.0110...0..0000.0...11.000...001.101.001)

7457
7458 4151: !(FREE)
7459 COMP131B1:
7460     P0,    BUMP-VERIFY,          !COUNT
7461     P2-T,    D+SR-XOR-CSPD(C177777),    !COMPARE RECEIVED: EXPECTED
7462     SAVE-D(C)          !SAVE CARRY
7463     NEXT,    J/GOBUT131B1
(4151) DCS(0.00.0.0.0.0) BM(0110..10.00..00.00..000..111...0.1.0..0..0...0.0101...0..0000.0...11.000...001.101.010)

7464
7465 4152: !(FREE)
7466 GOBUT131B1:
7467     SETUP,    RETURN/TEST131B2,    !RETURN TO START OF NEXT SUBTEST
7468     NEXT,    CALL(DINTOIR-5)    !GO PUT D -> IR, BUT(INSTR5)
(4152) DCS(0.00.0.0.0.0) BM(0100..00.11..11.01..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

7469
7470
7471
7472 !CHECK THAT CARRYOUT OF BIT15 (COUT15) WAS CORRECT
7473 4750:
7474 TEST131B2:
7475     P0,    LOAD-ENVA(ZTARGET413),    !BIT<02> CLEAR
7476     LOAD-ERROR(TEST131B2),    !ERROR DIRECTORY KEY
7477     DCS-CTR(C3.),    !COMPARE AT TARGET
7478     NEXT,    J/GOBUT131B2
(4750) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..001..011...0.0.0..0..0...0.0000...0..0000.0...11.000...001.101.011)

7479
7480 4153: !(FREE)
7481 GOBUT131B2:
7482     SETUP,    RETURN/SCOPE131B,    !RETURN TO SCOPE LOOP TEST WORD
7483     NEXT,    GOTO-PAGE(7),    !BUT TABLE
7484     J/BUTD(C)          !D(C)H= COUT15 H IN BIT<02>
(4153) DCS(0.00.0.0.0.0) BM(0100..00.00..11.01..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.000)

7485
7486
7487
7488 4154: !(FREE)
7489 SCOPE131B:
7490     P3,    CSPD(05)+EMIT, EMIT/170360,    !CONSTANT FOR USE BELOW
7491     NEXT,    BUTD(SCOPE),    !NO ERROR: "TEST132A1" (+1. WORDS)
7492     J/TEST132A1    !ERROR: "ARITH131A1" (-11. WORDS)
(4154) DCS(0.00.0.1.0.0) BM(1111..10.00..00.11..110..000...0.0.0..0..0...0.1010...1..0000.0...11.000...101.010.001)

7493
7494
7495

```



```

7496 ! - - - - -
7497
7498 !CHECK INTERNAL ALU CARRIES WITH: (125252)-(125252)-(0)=(000000)
7499 !ALSO CHECK ALU FUNCTION "A-MINUS-B", D(C)+COUT15=(1)
7500
7501 4521:
7502 TEST132A1:
7503     PO,          LOAD-ENVA(ZTARGET434),          !SETUP FOR IR=(000000)/BUTINSTRS TEST
7504                 LOAD-ERROR(TEST132A1),          !ERROR DIRECTORY KEY
7505                 DCS-CTR(C7.),                    !COMPARE AT TARGET
7506                 NEXT, J/ARITH132A1
(4521) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...101.010.010)

7507
7508 4522:
7509 ARITH132A1:
7510     P3-T,        D+A-MINUS-B,                    !ALU=(A-MINUS-B-MINUS-1), CIN=(1)
7511                 D(C)+COUT15,                    !GET CARRYOUT
7512                 BUS-A+ASPHI(C125252),            !A=(125252)
7513                 BUS-B+BSPHI(C125252),            !B=(125252)
7514                 NEXT, J/GOBUT132A1               !D=(000000), COUT15=(1)
(4522) DCS(0.00.0.0.0.0) BM(1101..01.11..11.01..110..110...1.1.0..0..0...0.0000...0..0000.0...11.000...001.101.101)

7516
7517 4155: !(FREE)
7518 GOBUT132A1:
7519     PO,          BUMP-VERIFY,                    !COUNT
7520     SETUP,       RETURN/TEST132A2,                !RETURN TO START OF NEXT SUBTEST
7521     NEXT,        CALL(DINTOIR-5)                  !GO PUT D -> IR, BUT(INSTR5)
(4155) DCS(0.00.0.0.0.1) BM(0100..00.10..10.01..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

7522
7523
7524
7525 !CHECK THAT CARRYOUT OF BIT15 (COUT15) WAS CORRECT
7526 4516:
7527 TEST132A2:
7528     PO,          LOAD-ENVA(ZTARGET403),          !BIT<01> SET
7529                 LOAD-ERROR(TEST132A2),          !ERROR DIRECTORY KEY
7530                 DCS-CTR(C3.),                    !COMPARE AT TARGET
7531                 NEXT, J/GOBUT132A2
(4516) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...001.101.110)

7532
7533 4156: !(FREE)
7534 GOBUT132A2:
7535     SETUP,       RETURN/TEST132B1,                !RETURN TO START OF NEXT SUBTEST
7536     NEXT,        GOTO-PAGE(7),                    !BUT TABLE
7537                 J/BUTD(C)B                        !D(C)H= COUT15 H IN BIT<01>
(4156) DCS(0.00.0.0.0.0) BM(0100..00.10..10.01..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.101.000)

7538
7539
7540
7541 ! - - - - -
7542

```

```

7543 !CHECK INTERNAL ALU CARRIES WITH: (052525)-(052525)-(0)=(000000)
7544 !ALSO CHECK ALU FUNCTION "DIVIDE-STEP" = "A-MINUS-B" SINCE D(C)=(1)
7545 !FROM ABOVE. D(C)+COUT15=(1)
7546 4157:
7547 TEST13281:
7548 PO, LOAD-ENUA(ZTARGET434), !SETUP FOR IR=(000000)/BUTINSTRS TEST
7549 LOAD-ERROR(TEST13281), !ERROR DIRECTORY KEY
7550 DCS-CTR(C7.), !COMPARE AT TARGET
7551 NEXT, J/ARITH13281
(4157) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...001.101.111)

7552 4157: !(FREE)
7553 ARITH13281:
7554 P3-T, D+DIVIDE-STEP, !ALU=(A-MINUS-B-MINUS-1), CIN=(1)
7555 D(C)+COUT15, !GET CARRYOUT
7556 BUS-A+ASPHI(C052525), !A=(052525)
7557 BUS-B+BSPHI(C052525), !B=(052525)
7558 !D=(000000), COUT15=(1)
7559 NEXT, J/GOBUT13281
(4157) DCS(0.00.0.0.0.0) BM(1000..01.11..11.01..111..110...1.1.0..0..0...0.0000...0..0000.0...11.000...001.110.000)

7561 4160: !(FREE)
7562 GOBUT13281:
7563 PO, BUMP-VERIFY !COUNT
7564 SETUP, RETURN/TEST13282, !RETURN TO START OF NEXT SUBTEST
7565 NEXT, CALL(DINTOIR-5) !GO PUT D -> IR, BUT(INSTR5)
7566 (4160) DCS(0.00.0.0.0.1) BM(0100..00.10..01.11..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

7567
7568
7569 !CHECK THAT CARRYOUT OF BIT15 (COUT15) WAS CORRECT
7570 4472:
7571 TEST13282:
7572 PO, LOAD-ENUA(ZTARGET417), !BIT<02> SET
7573 LOAD-ERROR(TEST13282), !ERROR DIRECTORY KEY
7574 DCS-CTR(C3.), !COMPARE AT TARGET
7575 NEXT, J/GOBUT13282
(4472) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..001..111...0.0.0..0..0...0.0000...0..0000.0...11.000...001.110.001)

7577 4161: !(FREE)
7578 GOBUT13282:
7579 SETUP, RETURN/SCOPE1328, !RETURN TO SCOPE LOOP TEST WORD
7580 NEXT, GOTO-PAGE(7), !BUT TABLE
7581 J/BUTD(C) !D(C)H=COUT15 H IN BIT <02>
7582 (4161) DCS(0.00.0.0.0.0) BM(0100..00.00..11.10..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.000)

7583
7584
7585 4162: !(FREE)
7586 SCOPE1328:
7587 P3, CSPD(06)+EMIT, EMIT/007417, !CONSTANT FOR USE BELOW
7588 NEXT, BUTD(SCOPE), !NO ERROR: "TEST133A1" (+1. WORDS)
7589

```

7590 J/TEST133A1 ! ERROR: "ARITH132A1" (-9. WORDS)  
 (4162) DCS(0.00.0.1.0.0) BM(0000..10.11..11.00..001..111...0.0.0..0..0...0.1001...1..0000.0...11.000...101.010.011)

7591  
7592  
7593  
7594  
7595

! -----

7596  
7597 !CHECK CARRY PROPOGATE/GENERATE LOGIC WITH:  
7598 !(103607)-PLUS-(103607)-PLUS-(1)=(007417), COUT15=(1)

7599 4523:  
7600 TEST133A1:  
7601 PO, LOAD-ENUA(ZTARGET434), !SETUP FOR IR=(000000)/BUTINSTRS COMPARE  
7602 LOAD-ERROR(TEST133A1), !ERROR DIRECTORY KEY  
7603 DCS-CTR(C10.), !COMPARE AT TARGET  
7604 NEXT, J/OPB133A1  
 (4523) DCS(1.00.1.0.0.0) BM(0101..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...101.001.000)

7605  
7606 4510:  
7607 OPB133A1:  
7608 P3, CSPD(16)+EMIT, !A B-SIDE OPERANDS:  
7609 EMIT/103607, !"1000 0111 1000 0111"  
7610 NEXT, J/DOPA133A1  
 (4510) DCS(0.00.0.0.0.0) BM(1000..10.01..11.10..000..111...0.0.0..0..0...0.0001...1..0000.0...11.000...001.110.011)

7611  
7612 4163: !(FREE)  
7613 DOPA133A1:  
7614 P2-T, D+CSPD(D16), D(C)+ALU15, !OP-A INTO D, D(C)+1  
7615 SR+CSPD(D16) !OP-A INTO SR TOO  
7616 NEXT, J/GOTEST133A1 ! (OPA=OPB)  
 (4163) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..100...0.1.1..0..0...0.0001...0..0000.0...11.000...001.110.100)

7617  
7618 4164: !(FREE)  
7619 GOTEST133A1:  
7620 PO BUMP-VERIFY !COUNT  
7621 SETUP, RETURN/TEST133A2, !EXEC SUBR WHICH:  
7622 ! (1) D+OPA-PLUS-OPB-PLUS-1  
7623 ! (2) D+007417-XOR-D (EQUAL?)  
7624 NEXT, CALL(ALUCARRY1) ! (3) J/BUTINSTRS/(000000) (CHECK ANSWER)  
 (4164) DCS(0.00.0.0.0.1) BM(0100..00.10..10.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.000.111)

7625  
7626  
7627  
7628 !CHECK THAT CARRYOUT COUT15 ABOVE GENERATED CORRECTLY AS A (1)

7629 4500:  
7630 TEST133A2:  
7631 PO, LOAD-ENUA(ZTARGET403), !BIT<00>SET  
7632 LOAD-ERROR(TEST133A2), !ERROR DIRECTORY KEY  
7633 DCS-CTR(C3.), !COMPARE AT TARGET  
7634 BUMP-VERIFY !COUNT  
7635 NEXT, J/GOBUT133A2  
 (4500) DCS(1.00.1.0.0.1) BM(1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...001.110.101)

7636

```

7637 4165: !(FREE)
7638 GOBUT133A2:
7639     SETUP, RETURN/TEST13381, !RETURN TO START OF NEXT SUBTEST
7640     NEXT, GOTO-PAGE(7), !BUT TABLE
7641     J/BUTD(C)A !D(C)H = COUT15 IN BIT<00>
(4165) DCS(0.00.0.0.0.0) BM(0100..00.10..10.00..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.100)

7642
7643
7644
7645
7646
7647 ! - - - - -
7648
7649 !CHECK CARRY PROPOGATE/GENERATE LOGIC WITH COMPLEMENT OF ABOVE:
7650 !(074170)-PLUS-(074170)-PLUS-(0)=(170360), COUT15=(0)
7651 4501:
7652 TEST13381:
7653     PO, LOAD-ENUA(ZTARGET434), !SETUP FOR IR=(000000)/BUTINSTRS COMPARE
7654     LOAD-ERROR(TEST13381), !ERROR DIRECTORY KEY
7655     DCS-CTR(C9.), !COMPARE AT TARGET
7656     NEXT, J/OPA13381
(4501) DCS(1.00.1.0.0.0) BM(0110..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...001.110.110)

7657
7658 4166: !(FREE)
7659 OPA13381:
7660     P2-T, D+NOT-CSPD(D16), !
7661     D(C)+ALU15, !OP-A INTO D, D(C)+0
7662     SR+NOT-CSPD(D16), !OP-A INTO SR TOO
7663     !A-SIDE OPERAND:
7664     !"0111 1000 0111 1000"
7665     !B-SIDE OPERAND WILL BE:
7666     !NEXT, J/GOTEST13381 ! (SAME AS OP-A)
(4166) DCS(0.00.0.0.0.0) BM(0111..10.00..11.01..101..100...0..1..0..0...0.0001...0..0000.0...11.000...001.110.111)

7667
7668 4167: !(FREE)
7669 GOTEST13381:
7670     SETUP, RETURN/TEST13382, !EXEC SUBR WHICH:
7671     ! (1) D+(NOT-OPA)-PLUS-(NOT-OPB)
7672     ! (2) D+170360-XOR-D (EQUAL?)
7673     ! (3) J/BUTINSTRS/(000000) (CHECK ANSWER)
(4167) DCS(0.00.0.0.0.0) BM(0100..00.10..10.00..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.010.000)

7674
7675
7676
7677 !CHECK THAT CARRYOUT COUT15 ABOVE GENERATED CORRECTLY AS A (0)
7678 4502:
7679 TEST13382:
7680     PO, LOAD-ENUA(ZTARGET402), !BIT<00> CLEAR
7681     LOAD-ERROR(TEST13382), !ERROR DIRECTORY KEY
7682     NEXT, J/GOBUT13382
(4502) DCS(1.00.0.0.0.0) BM(0000..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...001.!11.000)

7683

```

```

7684 4170: !(FREE)
7685 GOBUT13382:
7686     SETUP, RETURN/SCOPE1338, !RETURN TO SCOPE LOOP TEST WORD
7687     NEXT, GOTO-PAGE(7), !BUT TABLE
7688     J/BUTD(C)A !D(C)H = COUT15 H IN BIT<00>
(4170) DCS(0.00.0.0.0.0) BM(0100..00.00..11.11..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.100)

7689
7690
7691
7692 4171: !(FREE)
7693 SCOPE1338:
7694     PO, BUMP-VERIFY, !COUNT
7695     NEXT, BUTD(SCOPE), !NO ERROR: "TEST134A1" (+1. WORDS)
7696     J/TEST134A1 !ERROR: "OPA133A1" (-11. WORDS)
(4171) DCS(0.00.0.1.0.1) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...101.001.001)

7697
7698
7699
7700 ! - - - - -
7701
7702 !CHECK CARRY PROPOGATE/GENERATE LOGIC WITH:
7703 !(045513)-PLUS-(141703)-PLUS-(1)=(007417), COUT15=(1)
7704 4511:
7705 TEST134A1:
7706     PO, LOAD-ENVA(ZTARGET434), !SETUP FOR IB=(000000)/BUTINSTRS COMPARE
7707     LOAD-ERROR(TEST134A1), !ERROR DIRECTORY KEY
7708     DCS-CTR(C11.), !COMPARE AT TARGET
7709     NEXT, J/OPA134A1
(4511) DCS(1.00.1.0.0.0) BM(0100..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...101.001.010)

7710
7711 4512:
7712 OPA134A1:
7713     PO, BUMP-VERIFY, !COUNT
7714     P3, CSPD(17)+EMIT, !A-SIDE OPERAND:
7715     EMIT/045513, !"0100 1011 0100 1011"
7716     NEXT, J/OPB134A1
(4512) DCS(0.00.0.0.0.1) BM(0100..10.10..11.01..001..011...0.0.0..0..0...0.0000...1..0000.0...11.000...001.111.010)

7717
7718 4172: !(FREE)
7719 OPB134A1:
7720     P3, CSPD(16)+EMIT, !B-SIDE OPERAND:
7721     EMIT/141703, !"1100 0011 1100 0011"
7722     NEXT, J/DOPA134A1
(4172) DCS(0.00.0.0.0.0) BM(1100..10.00..11.11..000..011...0.0.0..0..0...0.0001...1..0000.0...11.000...001.111.011)

7723
7724 4173: !(FREE)
7725 DOPA134A1:
7726     P2-T, D+CSPD(D17), D(C)+ALU00, !OP-A INTO D, D(C)+1
7727     SR+CSPD(D17), !OP-A INTO SR TOO
7728     NEXT, J/GOTEST134A1
(4173) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..010...0.1.1..0..0...0.0000...0..0000.0...11.000...001.111.100)

7729
7730 4174: !(FREE)

```

```

7731 GOTEST134A1:
7732     PD,      BUMP-VERIFY,          !COUNT
7733     SETUP,   RETURN/TEST134A2,     !EXEC SUBR WHICH:
7734                                     ! (1) D+OPA-PLUS-OPA-PLUS-1
7735                                     ! (2) D+007417-XOR-D (EQUAL?)
7736     NEXT,    CALL(ALUCARRY1)       ! (3) J/BUTINSTRS/(00000) (CHECK ANSWER)
(4174) DCS(0.00.0.0.0.1) BM(0100..00.10..10.00..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.000.111)

7737
7738
7739
7740 !CHECK THAT CARRYOUT COUT15 ABOVE GENERATED CORRECTLY AS A (1)
7741 4503:
7742 TEST134A2:
7743     PD,      LOAD-ENUA(ZTARGET403), !BIT<01> SET
7744             LOAD-ERROR(TEST134A2), !ERROR DIRECTORY KEY
7745             DCS-CTR(C3.),          !COMPARE AT TARGET
7746             BUMP-VERIFY,          !COUNT
7747     NEXT,    J/GOBUT134A2
(4503) DCS(1.00.1.0.0.1) BM(1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...001.111.101)

7748
7749 4175: !(FREE)
7750 GOBUT134A2:
7751     SETUP,   RETURN/TEST134B1,     !RETURN TO START OF NEXT SUBTEST
7752     NEXT,    GOTO-PAGE(7),         !BUT TABLE
7753             J/BUTD(C18)           !D[C] = COUT15 IN BIT<01>
(4175) DCS(0.00.0.0.0.0) BM(0100..00.10..10.00..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.101.000)

7754
7755
7756
7757
7758
7759 !-----
7760
7761 !CHECK CARRY PROPOGATE/GENERATE LOGIC WITH COMPLEMENT OF ABOVE:
7762 !((132264)-PLUS-(036074)-PLUS-(0))=(170360), COUT15=(0)
7763 4504:
7764 TEST134B1:
7765     PD,      LOAD-ENUA(ZTARGET434), !SETUP FOR IR=(00000)/BUTINSTRS COMPARE
7766             LOAD-ERROR(TEST134B1), !ERROR DIRECTORY KEY
7767             DCS-CTR(C9.),          !COMPARE AT TARGET
7768     NEXT,    J/OPA134B1
(4504) DCS(1.00.1.0.0.0) BM(0110..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...001.111.110)

7769
7770 4176: !(FREE)
7771 OPA134B1:
7772     P2-T,    D+NOT-CSPD(D17),       !
7773             D[C]+ALU00,             !OP-A INTO D, D[C]+(0)
7774             SR+NOT-CSPD(D17),      !OP-A INTO SR TOO
7775                                     !A-SIDE OPERAND:
7776                                     !"1011 0100 1011 0100"
7777                                     !B-SIDE OPERAND WILL BE:
7778     NEXT,    J/GOTEST134B1        !"0011 1100 0011 1100"

```

(4176) DCS(0.00.0.0.0.0) BM(0111..10.00..11.01..101..010...0.1.1..0..0...0.0000...0..0000.0...11.000...001.111.111)

7779  
7780  
7781  
7782  
7783  
7784  
7785

4177: !(FREE)  
GOTEST13481:  
SETUP, RETURN/TEST13482,

!EXEC SUBR WHICH:  
!(1) D+(NOT-OPA)-PLUS-(NOT-OPB)  
!(2) D+170360-XOR-D (EQUAL?)  
!(3) J/BUTINSTRS/(000000) (CHECK ANSWER)

NEXT, CALL(ALUCARRY2)  
(4177) DCS(0.00.0.0.0.0) BM(0100..00.10..10.00..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.010.000)

7786  
7787  
7788  
7789  
7790  
7791  
7792  
7793  
7794  
7795

!CHECK THAT CARRYOUT COUT15 ABOVE GENERATED CORRECTLY AS A (0)

4505:  
TEST13482:  
PO, LOAD-ENUA(ZTARGET401),  
LOAD-ERROR(TEST13482),  
DCS-CTR(C3.),  
NEXT, J/GOBUT13482

!BIT<01> CLEAR  
!ERROR DIRECTORY KEY  
!COMPARE AT TARGET

(4505) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...010.000.000)

7796  
7797  
7798  
7799  
7800  
7801

4200: !(FREE)  
GOBUT13482:  
SETUP, RETURN/SCOPE1348,  
NEXT, GOTO-PAGE(7),  
J/BUTD(C)B

!RETURN TO SCOPE LOOP TEST WORD  
!BUT TABLE  
!D(C)H = COUT15 H IN BIT<01>

(4200) DCS(0.00.0.0.0.0) BM(0100..00.01..00.00..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.101.000)

7802  
7803  
7804  
7805  
7806  
7807  
7808  
7809

4201: !(FREE)  
SCOPE1348:  
PO, BUMP-VERIFY,  
NEXT, BUTD(SCOPE),  
J/TEST135A1

!COUNT  
!NO ERROR: "TEST135A1" (+1. WORDS)  
!ERROR: "OPA134A1" (-11. WORDS)

(4201) DCS(0.00.0.1.0.1) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...101.001.011)

7810  
7811  
7812  
7813  
7814  
7815  
7816  
7817  
7818  
7819  
7820  
7821  
7822

!CHECK CARRY PROPOGATE/GENERATE LOGIC WITH:  
!(122645)-PLUS-(064551)-PLUS-(1)=(007417), COUT15=(1)

4513:  
TEST135A1:  
PO, LOAD-ENUA(ZTARGET434),  
LOAD-ERROR(TEST135A1),  
DCS-CTR(C11.),  
NEXT, J/OPA135A1

!SETUP FOR IR=(000000)/BUTINSTRS COMPARE  
!ERROR DIRECTORY KEY  
!COMPARE AT TARGET

(4513) DCS(1.00.1.0.0.0) BM(0100..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...101.001.100)

7823  
7824

4514:

```

7825 OPA135A1:
7826 PO, BUMP-VERIFY, !COUNT
7827 P3, CSPD(17)+EMIT, !A-SIDE OPERAND:
7828 EMIT/122645, !"1010 0101 1010 0101"
7829 NEXT, J/OPB135A1
(4514) DCS(0.00.0.0.0.1) BM(1010..10.01..01.10..100..101...0.0.0..0..0...0.0000...1..0000.0...11.000...010.000.0101)

7830
7831 4202: !(FREE)
7832 OPB135A1:
7833 P3, CSPD(16)+EMIT, !B-SIDE OPERAND:
7834 EMIT/064551, !"0110 1001 0110 1001"
7835 NEXT, J/DOPA135A1
(4202) DCS(0.00.0.0.0.0) BM(0110..10.10..01.01..101..001...0.0.0..0..0...0.0001...1..0000.0...11.000...010.000.0111)

7836
7837 4203: !(FREE)
7838 DOPA135A1:
7839 P2-T, D+CSPD(D17), D(C)+ALU15, !OP-A INTO D, D(C)+1
7840 SR+CSPD(D17), !OP-A INTO SR TOO
7841 NEXT, J/GOTEST135A1
(4203) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..100...0.1.1..0..0...0.0000...0..0000.0..11.000...010.000.1001)

7842
7843 4204: !(FREE)
7844 GOTEST135A1:
7845 PO, BUMP-VERIFY, !COUNT
7846 SETUP, RETURN/TEST135A2, !EXEC SUBR WHICH:
7847 ! (1) D+OPA-PLUS-OPB-PLUS-1
7848 ! (2) D+007417-XOR-D (EQUAL?)
7849 NEXT, CALL[ALUCARRY1] ! (3) J/BUTINSTRS/(000000) (CHECK ANSWER)
(4204) DCS(0.00.0.0.0.1) BM(0100..00.10..10.00..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.000.1111)

7850
7851
7852 !CHECK THAT CARRYOUT COUT15 ABOVE GENERATED CORRECTLY AS A (1)
7853 4506:
7854 TEST135A2:
7855 PO, LOAD-ENLUA(ZTARGET417), !BIT<02> SET
7856 LOAD-ERROR(TEST135A2), !ERROR DIRECTORY KEY
7857 DCS-CTR(C3.), !COMPARE AT TARGET
7858 BUMP-VERIFY, !COUNT
7859 NEXT, J/GOBUT135A2
(4506) DCS(1.00.1.0.0.1) BM(1100..00.11..11.00..001..111...0.0.0..0..0...0.0000...0..0000.0...11.000...010.000.1011)

7861
7862 4205: !(FREE)
7863 GOBUT135A2:
7864 SETUP, RETURN/TEST135B1, !RETURN TO START OF NEXT SUBTEST
7865 NEXT, GOTO-PAGE(7), !BUT TABLE
7866 J/BUTD(C)C !D(C)H = COUT15 IN BIT<02>
(4205) DCS(0.00.0.0.0.0) BM(0100..00.10..10.00..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.0001)

7867
7868
7869
7870

```



```

7871
7872 ! - - - - -
7873
7874 !CHECK CARRY PROPOGATE/GENERATE LOGIC WITH COMPLEMENT OF ABOVE:
7875 ! (055132)-PLUS-(113226)-PLUS-(0)=(170360), COUT15=(0)
7876 4507:
7877 TEST135B1:
7878     PO,          LOAD-ENUA(ZTARGET434),          !SETUP FOR IR=(000000)/BUTINSTRS COMPARE
7879                LOAD-ERROR(TEST135B1);          !ERROR DIRECTORY KEY
7880                DCS-CTR(C9.),                    !COMPARE AT TARGET
7881     NEXT        J/OPA135B1
(4507) DCS(1.00.1 0.0.0) BM(0110..00.11..1..00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...010.000.110)

7882
7883 4206: !(FREE)
7884 OPA135B1:
7885     P2-T,       D+NOT-CSPD(D17),
7886                D(C)+ALU15,
7887                SR+NOT-CSPD(D17),
7888
7889
7890
7891     NEXT        J/GOTEST135B1
(4206) DCS(0.00.0.0.0.0) BM(0111..10.00..11.01..101..100. .0.1.1..0..0...0.0000...0..0000.0...11.000...010.000.111)

7892
7893 4207: !(FREE)
7894 GOTEST135B1:
7895     SETUP,      RETURN/TEST135B2,
7896
7897
7898     NEXT        CALL[ALUCARRY2]
(4207) DCS(0.00.0.0.0.0) BM(0100..00.10..01.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.010.000)

7899
7900
7901
7902 !CHECK THAT CARRYOUT COUT15 ABOVE GENERATED CORRECTLY AS A (0)
7903 4477:
7904 TEST135B2:
7905     PO,          LOAD-ENUA(ZTARGET413),          !BIT<02> CLEAR
7906                LOAD-ERROR(TEST135B2);          !ERROR DIRECTORY KEY
7907                DCS-CTR(C3.),                    !COMPARE AT TARGET
7908     NEXT        J/GOBUT135B2
(4477) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..001..011...0.0.0..0..0...0.0000...0..0000.0...11.000...010.001.000)

7909
7910 4210: !(FREE)
7911 GOBUT135B2:
7912     SETUP,      RETURN/SCOPE135B,
7913     NEXT,       GOTO-PAGE(7),
7914                J/BUTD(C)
(4210) DCS(0.00.0.0.0.0) BM(0100..00.01..00.01..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.000)

7915
7916
7917

```

```

7918 4211: !(FREE)
7919 SCOPE1358:
7920 PO, BUMP-VERIFY, !COUNT
7921 NEXT, BUTD(SCOPE), !NO ERROR: "TEST136A1" (+1. WORDS)
7922 J/TEST136A1 !ERROR: "OPA135A1" (-11. WORDS)
(4211) DCS(0.00.0.1.0.1) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...101.001.101)

```

```

7923
7924
7925
7926 ! - - - - -
7927

```

```

7928 !CHECK CARRY PROPOGATE/GENERATE LOGIC WITH:
7929 !(055132)-PLUS-(132264)-PLUS-(1)=(007417), COUT15=(1)
7930 4515:
7931 TEST136A1:
7932 PO, LOAD-ENUA(ZTARGET434), !SETUP FOR IR=(000000)/BUTINSTRS COMPARE
7933 LOAD-ERROR(TEST136A1), !ERROR DIRECTORY KEY
7934 DCS-CTR(C10.), !COMPARE AT TARGET
7935 NEXT, J/OPB136A1
(4515) DCS(1.00.1.0.0.0) BM(0101..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...101.100.010)

```

```

7936
7937 4542:
7938 OPB136A1:
7939 P3, CSPD(16)+EMIT, !B-SIDE OPERAND:
7940 NEXT, EAIT/132264, !"1011 0100 1011 0100"
7941 J/DOPA136A1
(4542) DCS(0.00.0.0.0.0) BM(1011..10.01..00.10..110..100...0.0.0..0..0...0.0001...1..0000.0...11.000...010.001.010)

```

```

7942
7943 4212: !(FREE)
7944 DOPA136A1:
7945 P2-T, D+NOT-CSPD(D17), D(C)+COUT07, !OP-A INTO D, D(C)+1
7946 SR+NOT-CSPD(D17), !OP-A INTO SR TOO
7947 NEXT, J/GOTEST136A1 !"0101 1010 0101 1010"
(4212) DCS(0.00.0.0.0.0) BM(0111..10.00..11.01..101..101...0.1.1..0..0...0.0000...0..0000.0...11.000...010.001.011)

```

```

7948
7949 4213: !(FREE)
7950 GOTEST136A1:
7951 PO, BUMP-VERIFY, !COUNT
7952 SETUP, RETURN/TEST136A2, !EXEC SUBR WHICH:
7953 ! (1) D+OPA-PLUS-OPB-PLUS-1
7954 ! (2) D+007417-XOR-D EQUAL?
7955 NEXT, CALL(ALUCARRY1) ! (3) J/BUTINSTRS/(000000) (CHECK ANSWER)
(4213) DCS(0.00.0.0.0.1) BM(0100..00.10..01.11..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.000.111)

```

```

7956
7957
7958 !CHECK THAT CARRYOUT COUT15 ABOVE GENERATED CORRECTLY AS A (1)
7959 4476:
7960 TEST136A2:
7961 PO, LOAD-ENUA(ZTARGET403), !BIT<00> SET
7962 LOAD-ERROR(TEST136A2), !ERROR DIRECTORY KEY
7963 DCS-CTR(C3.), !COMPARE AT TARGET
7964 BUMP-VERIFY, !COUNT
7965

```

```

7966      NEXT      J/GOBUT136A2
(4476) DCS[1.00.1.0.0.0] BM[1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...010.001.100]
7967
7968      4214:  !(FREE)
7969      GOBUT136A2:
7970      SETUP,    RETURN/TEST136B1,      !RETURN TO START OF NEXT SUBTEST
7971      NEXT,     GOTO-PAGE(7),          !BUT TABLE
7972      J/BUTD(C)A      !D(C)H = COUT15 IN BIT<00>
(4214) DCS[0.00.0.0.0.0] BM[0100..00.10..01.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.100]
7973
7974
7975
7976
7977
7978      ! - - - - -
7979
7980      !CHECK CARRY PROPOGATE/GENERATE LOGIC WITH COMPLEMENT OF ABOVE:
7981      !((122645)-PLUS-(132264)-PLUS-(0))=(170360), COUT15=(0)
7982      4475:
7983      TEST136B1:
7984      PD,        LOAD-ENUA(ZTARGET434),      !SETUP FOR IR=(000000)/BUTINSTRS COMPARE
7985      LOAD-ERROR(TEST136B1),                !ERROR DIRECTORY KEY
7986      DCS-CTR(C9.),                          !COMPARE AT TARGET
7987      NEXT,     J/OPA136B1
(4475) DCS[1.00.1.0.0.0] BM[0110..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...010.001.101]
7988
7989      4215:  !(FREE)
7990      OPA136B1:
7991      P2-T,    D+CSPD(D17),                !OP-A INTO D, D(C)+0
7992      D(C)+0,                               !OP-A INTO SR TOO
7993      SR+CSPD(D17),                          !A-SIDE OPERAND:
7994      "1010 0101 1010 0101"                !"1010 0101 1010 0101"
7995      "0100 1011 0100 1011"                !B-SIDE OPERAND WILL BE:
7996      "0100 1011 0100 1011"                !"0100 1011 0100 1011"
7997      NEXT,     J/GOTEST136B1
(4215) DCS[0.00.0.0.0.0] BM[1010..10.00..00.00..000..000...0.1.1..0..0...0.0000...0..0000.0...11.000...010.001.110]
7998
7999      4216:  !(FREE)
8000      GOTEST136B1:
8001      SETUP,    RETURN/TEST136B2,          !EXEC SUBR WHICH:
8002      ! (1) D+(NOT-OPA)-PLUS-(NOT-OPB)
8003      ! (2) D+170360-XOR-D (EQUAL?)
8004      ! (3) J/BUTINSTRS/(000000) (CHECK ANSWER)
      NEXT,     CALL[ALUCARRY2]
(4216) DCS[0.00.0.0.0.0] BM[0100..00.10..01.11..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.010.000]
8005
8006
8007
8008      !CHECK THAT CARRYOUT COUT15 ABOVE GENERATED CORRECTLY AS A (0)
8009      4473:
8010      TEST136B2:
8011      PD,        LOAD-ENUA(ZTARGET402),      !BIT<00> CLEAR
8012      LOAD-ERROR(TEST136B2),                !ERROR DIRECTORY KEY

```

```

8013          DCS-CTR(C3.),          !COMPARE AT TARGET
8014      NEXT,      J/GOBUT13682
(4473) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...010.001.1111)

8015
8016      4217:  !(FREE)
8017      GOBUT13682:
8018          SETUP,      RETURN/SCOPE1368,          !RETURN TO SCOPE LOOP TEST WORD
8019          NEXT,      GOTO-PAGE(7),          !BUT TABLE
8020          J/BUTD(C)A          !D(C)H = COUT15 H IN BIT<00>
(4217) DCS(0.00.0.0.0.0) BM(0100..00.01..00.10..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.1001)

8021
8022
8023
8024      4220:  !(FREE)
8025      SCOPE1368:
8026          PD,      BUMP-VERIFY,          !COUNT
8027          NEXT,      BUTD(SCOPE),          !NO ERROR: "TEST320A" (+1. WORDS)
8028          J/TEST320A          !ERROR: "OPA136A1" (-11. WORDS)
(4220) DCS(0.00.0.1.0.1) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...101.100.0111)

8029
8030
8031
8032
8033
8034
8035      ! -----
8036      !
8037      ! THE FOLLOWING TWO SUBROUTINES ARE USED IN THE ABOVE CARRY LOOKAHEAD/INTERNAL CARRIES TESTS:
8038
8039
8040      7007:  !(FREE)
8041      ALUCARRY1:
8042          P3-T,      D+A-PLUS-B-PLUS-D(C),          !D (- SUM OF A, B; D(C) WAS SET PREVIOUSLY
8043          D(C)+COUT15,          !GET CARRYOUT FOR EXAMINATION LATER
8044          SR+A-PLUS-B-PLUS-D(C),          !ALSO GET IT INTO THE SR
8045          BUS-A+SR          !A-SIDE OPERAND WAS IN THE SR
8046          BUS-B+CSPD(D16),          !B-SIDE OPERAND WAS IN CSP(16)
8047          NEXT,      J/ALUCARRY1A
(7007) DCS(0.00.0.0.0.0) BM(0100..10.00..00.00..000..110...1.1.1..0..0...0.0001...0..0000.0...11.000...000.001.1111)

8048
8049      7017:  !(FREE)
8050      ALUCARRY1A:
8051          P2-T,      D+SR-XOR-CSPD(D06), SAVE-D(C),          !COMPARE RECEIVED : (007417)
8052          NEXT,      J/DINTOIRS          !GO PUT D -> IR, BUT(INSTR5) FOR (000000)
(7017) DCS(0.00.0.0.0.0) BM(0110..10.00..00.00..000..111...0.1.0..0..0...0.1001...0..0000.0...11.000...010.111.0111)

8053
8054
8055
8056
8057      7020:  !(FREE)
8058      ALUCARRY2:
8059          P3-T,      D+A-PLUS-NOT-B-PLUS-D(C),          !D (- DIFF OF A, B; D(C) WAS SET PREVIOUSLY
8060          D(C)+COUT15,          !GET CARRYOUT FOR EXAMINATION LATER

```

```

8061          SR←A-PLUS-NOT-B-PLUS-D(C),          !ALSO GET IT INTO THE SR
8062          BUS←A←SR,                            !A-SIDE OPERAND WAS IN THE SR
8063          BUS←B←CSPD(D16),                      !B-SIDE OPERAND WAS IN CSP(16)
8064          NEXT J/ALUCARRY2A
(7020) DCS(0.00.0.0.0.0) BM(0101..10.00..00.00..000..110...1.1.1..0..0...0.0001...0..0000.0...11.000...000.010.001)

```

```

8065          7021: !(FREE)
8066          ALUCARRY2A:
8067          P2←T, D←SR←XOR←CSPD(D05), SAVE←D(C),    !COMPARE RECEIVED : (170360)
8068          NEXT J/DINTOIRS                        !GO PUT C → IR, BUT(INSTR5) FOR (000000)
8069          (7021) DCS(0.00.0.0.0.0) BM(0110..10.00..00.00..000..111...0.1.0..0..0...0.1010...0..0000.0...11.000...010.111.011)

```

```

8070
8071
8072
8073
8074
8075          !.PAGE=====
8076
8077
8078          .TOC * TEST320: D(C) SELECTION / COUT07-DOUT07 / D<14:00>=ZERO2BIT<00>
8079
8080

```

```

8081          !*****
8082          !*
8083          !* TESTS: 320 A - F                                UWORDS: 026 + 014
8084          !*
8085          !* FUNCTIONS:
8086          !*
8087          !* THESE SIX TESTS CHECK THE D(C) ADDRESS SELECTION LOGIC, COUT07 CARRY
8088          !* AND DOUT07 BITS, AND THE D<14:00>=ZERO BUT, WHEN ONLY BIT<00>=1 (NOT
8089          !* CHECKED IN TEST410).
8090          !*
8091          !*****

```

```

8092
8093
8094
8095          !TEST-320-A SETS ONLY THE D(C) INPUT "ALU00" (CODE=010), AND THEN CHECKS THAT D(C)←ALU00
8096          !RESULTS IN A "1".
8097          4543:
8098          TEST320A:
8099          PO, LOAD←ENLUA(ZTARGET403),          !BIT<00> SET
8100          LOAD←ERPOR(TEST320A),              !ERROR DIRECTORY KEY
8101          DCS←CTR(C6.),                      !COMPARE AT TARGET
8102          NEXT J/SETONE320A
(4543) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...111.000.000)

```

```

8103
8104          4700:
8105          SETONE320A:
8106          PO, BUMP←VERIFY,                   !COUNT
8107          P3, CSPD(17)←EMIT, EMIT/000001,    !A ONE
8108          NEXT J/SET0320A
(4700) DCS(0.00.0.0.0.1) BM(0000..10.00..00.00..000..001...0.0.0..0..0...0.0000...1..0000.0...11.000...010.010.001)

```

```

8109          4221: !(FREE)
8110

```

```

8111 SETD320A:
8112     P2-T,  D←CSPD(D17), D(C)←0,           !D GETS (000001)
8113         SR←CSPD(D17)                       !SO DOES SR
8114     P3,   A←BSPHI(111)←0,                 !STORE THE CONSTANT
8115     NEXT  J/GETDC320A
(4221) DCS(0.00.0.0.0.0) BM(1010..10.11..00.01..000..000...0.1.1..0..0...0.0000...0..10:1.0...11.000...010.010.010)

8116
8117 4222: !(FREE)
8118 GETDC320A:
8119     P2-T,  D←A-PLUS-B-PLUS-0, D(C)←ALU00,   !D(C) CODE (010)
8120         BUS←A←SR                             !A=(000001)
8121         BUS←B←CSPD(C000000),                 !B=(000000)
8122     NEXT  J/GOBUT320A                       !D=(000001), ONLY D(C)=ALU00 SET
(4222) DCS(0.00.0.0.0.0) BM(1001..10.00..00.00..000..010...0.1.0..0..0...0.0100...0..0000.0...11.000...010.010.011)

8123
8124 4223: !(FREE)
8125 GOBUT320A:
8126     PO,   BUMP-VERIFY,                       !COUNT
8127     SETUP, RETURN/TEST320B,                 !RETURN TO START OF NEXT SUBTEST
8128     NEXT,  GOTO-PAGE(7),                     !BUT TABLE
8129     J/BUTD(C)A                               !D(C)H IN BIT<00>
(4223) DCS(0.00.0.0.0.1) BM(0100..00.10..11.01..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.100)

8130
8131
8132
8133
8134 ! - - - - -
8135
8136 !TEST-320-B CHECKS THAT D<14:00>=ZERO IS NOT SET WHEN D=(000001)
8137 4552:
8138 TEST320B:
8139     PC,   LOAD-ENUA(ZTARGET400),             !BIT<01> CLEAR
8140         LOAD-ERROR(TEST320B),               !ERROR DIRECTORY KEY
8141         DCS-CTR(C3.),                       !COMPARE AT TARGET
8142         BUMP-VERIFY,                         !COUNT
8143     NEXT  J/GOBUT320B
(4552) DCS(1.00.1.0.0.1) BM(1100..00.11..11.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...010.010.100)

8144
8145 4224: !(FREE)
8146 GOBUT320B:
8147     SETUP, RETURN/TEST320C,                 !RETURN TO START OF NEXT SUBTEST
8148     NEXT,  GOTO-PAGE(7),                     !BUT TABLE
8149     J/BLD-IS-ZERO                             !CHEK ON IT
(4224) DCS(0.00.0.0.0.0) BM(0100..00.10..11.10..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

8150
8151
8152
8153 ! - - - - -
8154
8155 !TEST-320-C CHECKS THAT THE COUT15/COUT07 SIGNALS DON'T TRACK EACH OTHER
8156 4562:
8157 TEST320C:

```

```

8158          PO,      LOAD-ENUA(ZTARGET413),      !BIT<02> CLEAR
8159          LOAD-ERROR(TEST320C),                !ERROR DIRECTORY KEY
8160          DCS-CTR(C6.),                          !COMPARE AT TARGET
8161          NEXT      J/SETONE320C
(4562) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..001..011...0.0.0..0..0...0.0000...0..0000.0...11.000...010.010.101)

```

```

8162          4225:  !(FREE)
8163          SETONE320C:
8164          P3,      CSPD(16)+EMIT, EMIT/100000,    !A ONE IN B15
8165          NEXT      J/SETD320C
(4225) DCS(0.00.0.0.0.0) BM(1000..10.00..00.00..000..000...0.0.0..0..0...0.0001...1..0000.0...11.000...010.010.110)

```

```

8167          4226:  !(FREE)
8168          SETD320C:
8169          P2-T,    D+CSPD(D16), D(C)+ALU15,        !D GETS (100000)
8170          SR+CSPD(D16),                          !SO DOES SR
8171          P3,      A#BSPHI(13)+0,                 !STORE THE CONSTANT
8172          NEXT      J/GETDC320C
(4226) DCS(0.00.0.0.0.0) BM(1010..10.11..00.01..001..100...0.1.1..0..0...0.0001...0..1011.0...11.000...010.010.111)

```

```

8174          4227:  !(FREE)
8175          GETDC320C:
8176          PO,      BUMP-VERIFY,                   !COUNT
8177          P2-T,    D+A-PLUS-B-PLUS-0, D(C)+COUT07, !D(C) GETS COUT07=0, COUT15=1
8178          BUS-A+SR,                                  !A=(100000)
8179          BUS-B+CSPD(C125252),                      !B=(125252)
8180          NEXT      J/GOBUT320C                    !D=(025252)
(4227) DCS(0.00.0.0.0.1) BM(1001..10.00..00.00..000..101...0.1.0..0..0...0.0110...0..0000.0...11.000...010.011.000)

```

```

8182          4230:  !(FREE)
8183          GOBUT320C:
8184          SETUP,   RETURN/TEST3200,                !RETURN TO START OF NEXT SUBTEST
8185          NEXT,    GOTO-PAGE(7),                    !BUT TABLE
8186          J/BUTD(C)                                  !D(C)H IN BIT<02>
(4230) DCS(0.00.0.0.0.0) BM(0100..00.10..10.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.000)

```

```

8188
8189
8190
8191
8192
8193          ! - - - - -
8194

```

```

8195          !TEST-320-D CHECKS THAT THE BUTR(COUT07#DOUT07) SEES THE (01*) THAT WAS GENERATED
8196          4537:
8197          TEST3200:
8198          PO,      LOAD-ENUA(ZTARGET403),          !BIT<2:i> = "01"
8199          LOAD-ERROR(TEST3200),                    !ERROR DIRECTORY KEY
8200          DCS-CTR(C3.),                              !COMPARE AT TARGET
8201          NEXT      J/GOBUT3200
(4537) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...010.011.001)

```

```

8202          4231:  !(FREE)
8203          GOBUT3200:
8204

```

```

8205          SETUP, RETURN/TEST320E,          !RETURN TO START OF NEXT SUBTEST
8206          NEXT,  GOTO-PAGE(7),             !BUT TABLE
8207          J/BUTCOUT7DOU7                    !COUT07#007 IN BIT<2:1>
(4231) DCS(0.00.0.0.0.0) BM(0100..00.10..11.01..100..111...0.0.0..C..0...0.0000...0..0000.0...11.100...011.001.001)

8208
8209
8210
8211
8212          ! - - - - -
8213
8214          !TEST-320-E SETS ONLY THE D(C) INPUT "COUT07" (CODE=101), AND THEN CHECKS THAT D(C)+COUT07
8215          ! RESULTS IN A "1".
8216          4554:
8217          TEST320E:
8218          PO,          LOAD-ENUA(ZTARGET403),          !BIT<01> SET
8219          LOAD-ERROR(TEST320E),          !ERROR DIRECTORY KEY
8220          DCS-CTR(C6.),          !COMPARE AT TARGET
8221          NEXT,        J/SETONE320E
(4554) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...010.011.010)

8222
8223          4232: !(FREE)
8224          SETONE320E:
8225          PO,          BUMP-VERIFY          !COUNT
8226          P3,          CSPD(15)+EMIT, EMIT/000200,    !A ONE IN BIT07
8227          NEXT,        J/SETD320E
(4232) DCS(0.00.0.0.0.1) BM(0000..10.00..00.10..000..000...0.0.0..0..0...0.0010...1..0000.0...11.000...010.011.011)

8228
8229          4233: !(FREE)
8230          SETD320E:
8231          P2-T,        D+CSPD(015), D(C)+0,          !D GETS (000200)
8232          SR+CSPD(015)          !SO DOES SR
8233          P3,          A#BSPHI(15)+0,          !STORE THE CONSTANT
8234          NEXT,        J/GETDC320E
(4233) DCS(0.00.0.0.0.0) BM(1010..10.11..00.01..010..000...0.1.1..0..0...0.0010...0..1011.0...11.000...010.011.100)

8235
8236          4234: !(FREE)
8237          GETDC320E:
8238          P2-T,        D+A-PLUS-B-PLUS-0, D(C)+COUT07,    !D(C) CODE (101)
8239          BUS-A+SR          !A=(000200)
8240          BUS-B+CSPD(015),    !B=(000200)
8241          NEXT,        J/GOBUT320E          !D=(000400), ONLY D(C)=COUT07 SET
(4234) DCS(0.00.0.0.0.0) BM(1001..10.00..00.00..000..101...0.1.0..0..0...0.0010...0..0000.0...11.000...010.011.101)

8242
8243          4235: !(FREE)
8244          GOBUT320E:
8245          SETUP, RETURN/TEST320F,          !RETURN TO START OF NEXT SUBTEST
8246          NEXT,  GOTO-PAGE(7),             !BUT TABLE
8247          J/BUTD(C)B          !D(C)H IN BIT<01>
(4235) DCS(0.00.0.0.0.0) BM(0100..00.10..11.01..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.101.000)

8248
8249
8250

```



```

8251
8252 ! - - - - -
8253
8254 !TEST-320-F CHECKS THAT THE BUTR(COUT07#DO07) SEES THE (10*) THAT WAS GENERATED
8255 4556:
8256 TEST320F:
8257     PO,          LOAD-ENUA(ZTARGET405),          !BIT<2:1> = "10"
8258                 LOAD-ERROR(TEST320F),          !ERROR DIRECTORY KEY
8259                 DCS-CTR(C3.),                  !COMPARE AT TARGET
8260     NEXT,        J/GOBUT320F
(4556) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...010.011.110)

8261
8262 4236: !(FREE)
8263 GOBUT320F:
8264     SETUP,      RETURN/SCOPE320,          !RETURN TO SCOPE LOOP TEST WORD
8265     NEXT,        GOTO-PAGE(7),           !BUT TABLE
8266                 J/BUTCOUT7DO07         !COUT07#DO7 IN BIT<2:1>
(4236) DCS(0.00.0.0.0.0) BM(0100..00.01..00.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.001)

8267
8268
8269 4237: !(FREE)
8270 SCOPE320:
8271     PO,          BUMP-VERIFY,             !COUNT
8272                 BUSDIN+EMIT-[[I],        !RESET PROC UCONS
8273                 EN-CLK-IR[15-00],
8274     NEXT,        BUTD[SCOPE],            !NO ERROR: "TEST350" (+1. WORDS)
8275                 J/TEST350                !ERROR: "SETONE320A" (-20. WORDS)
(4237) DCS(0.00.0.1.0.1) BM(0000..00.00..00.01..000..100...0.0.0..0..0...1.1001...0..0000.0...11.000...111.000.001)

```

```

8276
8277
8278
8279
8280
8281
8282
8283 !.PAGE=====
8284
8285

```

```

8286 .TOC * TEST350-352: ASP/BSP HI/LO ADDRESSING MODES, DATA VALIDITY
8287
8288 !*****
8289 !*
8290 !* TESTS: 350 - 352                                UWORDS: 075 + 125
8291 !*
8292 !* FUNCTIONS:
8293 !*
8294 !* VERIFIES THE ASP/BSP ADDRESSING MODES.
8295 !*
8296 !*****
8297
8298
8299

```

```

8300 !TEST 350 A-D USES THE A/B SP "SF" AND "DF" ADDRESS MODES TO WRITE A UNIQUE PATTERN TO

```

8301 ! LOCATIONS (00:07) OF ASPHI, ASPL0, BSPHI, BSPL0, THRU USE OF A COUNT LOOP  
 8302 ! THE GENERATED PATTERN IS THEN READ BACK, AND COMPARED TO A REGENERATED ORIGINAL  
 8303 ! TO CHECK FOR ADDRESSING CORRECTNESS, ABILITY TO READ/WRITE, AND DATAPATH VALIDITY.

8304  
 8305 ! FIRST FILL UP THE SCRATCHPADS:

8306  
 8307 AT THE END OF THE FILLUP LOOP, A/B SP HI/LO LOOK LIKE THIS:

8308	LOCTN	ASPHI	BSPHI	ASPL0	BSPL0
8309	-----	-----	-----	-----	-----
8310					
8311					
8312	00	050700	050700	057077	057077
8313	01	051611	051611	056166	056166
8314	02	052522	052522	055255	055255
8315	03	053433	053433	054344	054344
8316	04	054344	054344	053433	053433
8317	05	055255	055255	052522	052522
8318	06	056166	056166	051611	051611
8319	07	057077	057077	050700	050700
8320	10	-----	-----	-----	-----
8321	11	000001	000001	-----	-----
8322	12	000152	000152	-----	-----
8323	13	100000	100000	000125	000125
8324	14	-----	-----	-----	-----
8325	15	000200	000200	-----	-----
8326	16	-----	-----	-----	-----
8327	17	-----	-----	-----	-----

8328  
 8329 NOTE: CONSTANTS FOR DCS IN ASPHI/BSPHI 01/03/05/07 WERE DESTROYED,  
 8330 AND \*MUST\* BE RESTORED AFTER LEAVING THIS TEST SEQUENCE.

8331  
 8332  
 8333  
 8334  
 8335  
 8336 4701:

8337 TEST350:

8338 PO, LOAD-ERROR(TEST350), ! ERROR DIRECTORY KEY  
 8339 P3, CSPD(15)+EMIT, EMIT/177067, ! ADDED TO PATTERN "N" TO GET "N+1"  
 8340 NEXT, J/START350

(4701) DCS(1.00.0.0.0.0) BM(1111..10.11..10.00..110..111...0.0.0..0..0...0.0010...1..0000.0...11.000...010.100.000)

8341  
 8342 4240: !(FREE)

8343 START350:

8344 PO, BUMP-VERIFY, ! COUNT  
 8345 P3, CSPD(14)+EMIT, EMIT/057077, ! THE INITIAL PATTERN  
 8346 ! SF=0, DM=7, DF=7

8347 NEXT, J/LOADSR0350  
 (4240) DCS(0.00.0.0.0.1) BM(0101..10.11..10.00..111..111...0.0.0..0..0...0.0011...1..0000.0...11.000...010.100.001)

8348  
 8349 4241: !(FREE)

8350 LOADSR0350:

8351 P2-T, D+CSPD(014), ! INITIAL PATTERN  
 8352 SR+CSPD(014), ! D->IR, SR-TEMP

```

8353      NEXT,      GOTO-PAGE(7),      |
8354      J/DINTOIR350      |
(4241) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..111...0.1.1..0..0...0.0011...0..0000.0...11.100...011.101.011)
8355
8356      !*** LOOP BACK POINT FOR SP FILLUP ***
8357
8358      7353:
8359      DINTOIR350:
8360      SETUP,      RETURN/WRTDF350,      |CALL SUBR FOR D -> IR
8361      NEXT,      CALL(DINTOIR)      |
(7353) DCS(0.00.0.0.0.0) BM(0100..00.00..00.00..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.110.111)
8362
8363      4002:      !(FREE)
8364      WRTDF350:
8365      P0,      DCS-CTR(C15.),      |HOLD UP
8366      P3,      A#BSPHI(DF)+D,      |USE DF TO WRITE A/B HI, W/ WR(AB,H,A)
8367      NEXT,      GOTO-PAGE(7),      |
8368      J/WRTSF350      |
(4002) DCS(0.00.1.0.0.0) BM(0000..00.00..00.10..000..111...0.0.0..0..0...0.0000...0..1011.0...11.100...000.001.110)
8369
8370      7016:      !(FREE)
8371      WRTSF350:
8372      P3,      A#BSPLO(SF)+D,      |USE SF TO WRITE A/B LO, W/ WR(AB,L,B)
8373      NEXT,      J/NEXTPAT350      |
(7016) DCS(0.00.0.0.0.0) BM(0000..00.01..00.11..000..000...0.0.0..0..0...0.0000...0..0011.0...11.000...000.010.011)
8374
8375      7023:      !(FREE)
8376      NEXTPAT350:
8377      P2-T,      SR#D+SR-PLUS-CSPD(D15), SAVE-D(C),      |INCREMENT PATTERN
8378      NEXT,      BUTR(DMO),      |IF TRUE: "LOADHI350" (+1. WORDS) EXIT LOOP
8379      J/DINTOIR350      |IF FALSE: "DINTOIR350" (-3. WORDS) KEEP GOING
(7023) DCS(0.00.0.0.0.0) BM(1001..10.00..00.00..000..111...0.1.1..0..0...0.0010...0..0000.0...01.001...011.101.011)
8380
8381      !COME HERE IF DONE LOOPING
8382      7357:
8383      LOADHI350:
8384
8385      |*** IF ANY ASP/BSP HI/LO BIT<03> IN ADDR STUCK ZERO,
8386      |ONE OF THE FOLLOWING TWO WORDS WILL OVERWRITE
8387      |A PREVIOUSLY WRITTEN LOCATION (IE, 02 OR 03) ***
8388      P2-T,      D+CSPD(D01), SAVE-D(C),      |PATTERN OF (000152)
8389      P3,      A#BSPHI(12)+D,      |WRITE SPHI WITH ADDRS BIT<03> SET, DF=0, W/ WR(AB,H,A)
8390      NEXT,      J/LOADHIA350      |FOR USE IN TEST373
(7357) DCS(0.00.0.0.0.0) BM(1010..10.10..00.00..001..111...0.1.0..0..0...0.1110...0..1011.0...11.000...000.010.100)
8391
8392      7024:      !(FREE)
8393      LOADHIA350:
8394      P2-T,      D+CSPD(D02), SAVE-D(C),      |PATTERN OF (000125)
8395      P3,      A#BSPLO(13)+D,      |WRITE SPHI WITH ADDRS BIT<03> SET, DF=0, W/ WR(AB,L,B)
8396      NEXT,      J/AGAINSRO350      |
(7024) DCS(0.00.0.0.0.0) BM(1010..10.11..00.01..001..111...0.1.0..0..0...0.1101...0..0011.0...11.000...000.010.101)
8397

```

```

8398 7025: !(FREE)
8399 AGAINSR0350:
8400 P2-T, D+CSP0(D14), !AGAIN RESET D, SR TO INITIAL PATTERN
8401 SR+CSP0(D14), !OF (057077)
8402 NEXT, GOTO-PAGE(4),
8403 J/DINTOIRA350
(7025) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..100...0.1.1..0..0...0.0011...0..0000.0...11.100...101.110.011)

8404
8405 !*** LOOP BACK ENTRY POINT FOR TESTS ***
8406
8407 4563:
8408 DINTOIRA350:
8409 SETUP, RETURN/TEST350A, !GO TO SUBR WHICH PUTS SR -> IR FOR SF/DF/DM
8410 NEXT, CALL(SRINTOIR)
(4563) DCS(0.00.0.0.0.0) BM(0100..00.11..11.01..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.110.110)

8411
8412 ! - - - - -
8413
8414 !TEST 350A CHECKS BSP-ADDRS/SF, BSPLO ADDR FOR ERRORS
8415 4752:
8416 TEST350A:
8417 PO, LOAD-ENVA(ZTARGET434), !INSTRS FOR IR=(000000)
8418 LOAD-ERROR(TEST350A), !ERROR DIRECTORY KEY
8419 DCS-CTR(C7.), !COMPARE AT TARGET
8420 NEXT, J/COMP350A
(4752) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...111.000.010)

8422
8423 4702:
8424 COMP350A:
8425 P2-T, D+SR-XOR-BSPLO(SF), SAVE-D(C), !COMPARE EXPECTED:RECEIVED, BITWISE
8426 NEXT, J/GOBUT350A
(4702) DCS(0.00.0.0.0.0) BM(0110..00.01..00.00..000..111...0.1.0..0..0...0.0000...0..0000.0...11.000...010.100.011)

8427
8428 4243: !(FREE)
8429 GOBUT350A:
8430 PO, BUMP-VERIFY, !COUNT
8431 SETUP, RETURN/RESETIR350A, !GO TO SUBR WHICH:
8432 NEXT, CALL(DINTOIR-5) ! D -> IR, BUT(INSTRS)
(4243) DCS(0.00.0.0.0.1) BM(0111..00.00..00.10..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

8433
8434 7022: !(FREE)
8435 RESETIR350A:
8436 SETUP, RETURN/TEST350B, !GO TO SUBR WHICH PUTS SR -> IR FOR SF/DF/DM
8437 NEXT, CALL(SRINTOIR)
(7022) DCS(0.00.0.0.0.0) BM(0100..00.11..11.01..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.110.110)

8438
8439 ! - - - - -
8440
8441 !TEST 350B CHECKS BSP-ADDRS/DF, BSPHI ADDR FOR ERRORS
8442

```

```

8443 4753:
8444 TEST350B:
8445     PO,      LOAD-ENUA(ZTARGET434),      !INSTRS FOR IR=(000000)
8446           LOAD-ERROR(TEST350B),        !ERROR DIRECTORY KEY
8447           DCS-CTR(C7.),                !COMPARE AT TARGET
8448           NEXT,      J/COMP350B
(4753) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...010.100.100)

8449 4244:  !(FREE)
8450 COMP350B:
8451     P2-T,    D+SR-XOR-BSPHI(DF), SAVE-D(C), !COMPARE EXPECTED:RECEIVED, BITWISE
8452     NEXT,    J/GOBUT350B
(4244) DCS(0.00.0.0.0.0) BM(0110..01.00..00.00..000..111...0.1.0..0..0...0.0000...0..0000.0...11.000...010.100.101)

8454 4245:  !(FREE)
8455 GOBUT350B:
8456     SETUP,   RETURN/RESETIR350B,        !GO TO SUBR WHICH:
8457     NEXT,    CALL(DINTOIR-5)           ! D -> IR, BUT(INSTRS)
(4245) DCS(0.00.0.0.0.0) BM(0111..00.00..00.10..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

8459 7026:  !(FREE)
8460 RESETIR350B:
8461     SETUP,   RETURN/TEST350C,          !GO TO SUBR WHICH PUTS SR -> IR FOR SF/DF/DM
8462     NEXT,    CALL(SRINTOIR)
(7026) DCS(0.00.0.0.0.0) BM(0100..00.11..11.00..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.110.110)

8464 ! - - - - -
8465 !
8466 !
8467 !
8468 !
8469 !TEST 350C CHECKS ASP-ADDRS/DF, ASPLO ADDRS FOR ERRORS
8470 4742:
8471 TEST350C:
8472     PO,      LOAD-ENUA(ZTARGET434),      !INSTRS FOR IR=(000000)
8473           LOAD-ERROR(TEST350C),        !ERROR DIRECTORY KEY
8474           DCS-CTR(C7.),                !COMPARE AT TARGET
8475           BUMP-VERIFY,                 !COUNT
8476           NEXT,      J/COMP350C
(4742) DCS(1.00.1.0.0.1) BM(1000..00.11..11.00..011..100...0.0.0..0..0...0.0000.. 0..0000.0...11.000...010.100.110)

8477 4246:  !(FREE)
8478 COMP350C:
8479     P2-T,    D+ASPLO(DF)-XOR-BSPHI(SF), SAVE-D(C), !COMPARE RECEIVED:EXPECTED, BITWISE
8480     NEXT,    J/GOBUT350C
(4246) DCS(0.00.0.0.0.0) BM(0110..01.01..10.10..000..111...0.1.0..0..0...0.0000...0..0000.0...11.000...010.100.111)

8482 4247:  !(FREE)
8483 GOBUT350C:
8484     SETUP,   RETURN/RESETIR350C,        !GO TO SUBR WHICH:
8485     NEXT,    CALL(DINTOIR-5)           ! D -> IR, BUT(INSTRS)
(4247) DCS(0.00.0.0.0.0) BM(0111..00.00..00.10..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)
8487

```

```

8488 7027: !(FREE)
8489 RESEtir350:
8490 SETUP, RETURN/TEST350, !GO TO SUBR WHICH PUTS SR -> IR FOR SF/DF/DM
8491 NEXT, CALL(SRINTOIR)
(7027) DCS(0.00.0.0.0.0) BM(0100..00.11..11.00..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.110.110)

```

```

8492
8493
8494
8495 ! - - - - -

```

```

8497 !TEST 3500 CHECKS ASP-ADDRS/SF, ASPHI ADDR FOR ERRORS
8498 4743:
8499 TEST3500:
8500 PO, LOAD-ENJA(ZTARGET434), !INSTRS FOR IR=(000000)
8501 LOAD-ERROR(TEST3500), !ERROR DIRECTORY KEY
8502 DCS-CTR(C7.), !COMPARE AT TARGET
8503 NEXT, J/COMP3500
(4743) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...010.101.000)

```

```

8504 4250: !(FREE)
8505 COMP3500:
8506 PO, BUMP-VERIFY, !COUNT
8507 P2-T, D+ASPHI(SF)-XOR-BSPLO(DF), SAVE-D(C), !COMPARE RECEIVED:EXPECTED, BITWISE
8508 NEXT, J/GOBUT3500
8509 (4250) DCS(0.00.0.0.0.1) BM(0110..00.00..11.11..000..111...0.1.0..0..0...0.0000...0..0000.0...11.000...010.101.001)

```

```

8510 4251: !(FREE)
8511 GOBUT3500:
8512 PO, BUMP-VERIFY, !COUNT
8513 SETUP, RETURN/SCOPE350, !GO TO SUBR WHICH:
8514 NEXT, CALL(DINTOIR-5) ! D -> IR, BUT(INSTR5)
8515 (4251) DCS(0.00.0.0.0.1) BM(0100..00.01..01.01..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

```

```

8516 4252: !(FREE)
8517 SCOPE350:
8518 P2-T, D+SR, SAVE-D(C), !FOR DISPLAY OF SF/DM/DF
8519 NEXT, BUTD(SCOPE), !NO ERROR: "RESEtir350" (+1. WORDS) KEEP ON TESTING
8520 ! ERROR: "COMP350A" (-11. WORDS) HOLD UP PATTERN
8521 J/RESEtir350
8522 (4252) DCS(0.00.0.1.0.0) BM(1111..00.00..00.00..000..111...0.1.0..0..0...0.0000...0..0000.0...11.000...111.000.011)
8523

```

```

8524
8525
8526
8527 ! - - - - -

```

```

8529 4703:
8530 RESEtir350:
8531 PO, BUMP-VERIFY, !COUNT
8532 SETUP, RETURN/NEXTPATA350, !PUT OLD PAT FROM SR INTO IR FOR DMO TEST
8533 NEXT, CALL(SRINTOIR)
(4703) DCS(0.00.0.0.0.1) BM(0100..00.01..01.01..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.110.110)

```

```

8534
8535 4253: !(FREE)
8536 NEXTPATA350:
8537 P2-T, SR#0+SR-PLUS-CSPD(D15), SAVE-D(C), !GENER NEXT PATTERN INTO D, SR
8538 NEXT, BUTR(DMO) !IF TRUE: "LOAD05-351" (+1. WORDS) ALL DONE HERE
8539 J/DINTOIRA350 !IF FALSE: "DINTOIRA" (-14. WORDS) KEEP ON TESTING
(4253) DCS(0.00.0.0.0.0) BM(1001..10.00..00.00..000..111...0.1.1..0..0...0.0010...0..0000.0...01.001...101.110.011)

```

```

8540
8541
8542
8543 ! - - - - -
8544
8545 !COME HERE IF DONE LOOPING
8546 !THESE CONISTANTS ARE USED IN THE NEXT TESTS:
8547 4567:
8548 LOAD05-351:
8549 PO, BUMP-VERIFY, !COUNT
8550 P3, CSPD(05)+EMIT, EMIT/055255, !SF=2, DF=5 CONSTANT
8551 NEXT, J/LOAD06-351
(4567) DCS(0.00.0.0.0.1) BM(0101..10.10..10.10..101..101...0.0.0..0..0...0.1010...1..0000.0...11.000...010.101.100)

```

```

8552
8553 4254: !(FREE)
8554 LOAD06-351:
8555 P3, CSPD(06)+EMIT, EMIT/054344, !SF=3, DF=4 CONSTANT
8556 NEXT, J/TEST351A
(4254) DCS(0.00.0.0.0.0) BM(0101..10.10..00.11..100..100...0.0.0..0..0...0.1001...1..0000.0...11.000...111.011.010)

```

```

8557
8558
8559
8560 ! - - - - -
8561
8562 !TESTS 351 A-D VERIFY THAT THE RIF ADDRESS, WITH ASP HI/LO IMMED0/1 MODES, OPERATES CORRECTLY
8563
8564 ! - - - - -
8565
8566 !TEST 351A CHECKS ASPLO RIF ADDR "001 0"=(C2)
8567 4732:
8568 TEST351A:
8569 PO, LOAD-ENVA(ZTARGET434), !INSTRS FOR IR=(000000)
8570 LOAD-ERROR(TEST351A), !ERROR DIRECTORY KEY
8571 DCS-CTR(C7.), !COMPARE AT TARGET
8572 NEXT, J/COMP351A
8573 (4732) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...111.000.100)

```

```

8574
8575 4704:
8576 COMP351A:
8577 P2-T, D+CSPD(05)-XOR-ASPLO(R02), SAVE-D(C), !COMPARE EXPECTED:RECEIVED, BITWISE
8578 NEXT, J/GOBUT351A !CSP(05)=(055255)
(4704) DCS(0.00.0.0.0.0) BM(0110..10.00..10.00..101..111...0.1.0..0..0...0.1010...0..0000.0...11.000...010.101.110)

```

```

8579
8580 4256: !(FREE)
8581 GOBUT351A:
8582 PO, BUMP-VERIFY, !COUNT

```

```

8583      SETUP, RETURN/TEST351B,      !GO TO SUBR WHICH:
8584      NEXT,  CALL(DINTOIR-5)        !  D -> IR, BUT(INSTR5)
(4256) DCS(0.00.0.0.0.0.1) BM(0100..00.11..10.10..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

```

```

8585
8586
8587 ! - - - - -
8588

```

```

8589 !TEST 351B CHECKS ASPLO RIF ADDR "001 1"=(03)
8590 4722:
8591 TEST351B:

```

```

8592      PO,      LOAD-ENUA(ZTARGET434),      !INSTRS FOR IR=(000000)
8593      LOAD-ERROR(TEST351B),                !ERROR DIRECTORY KEY
8594      DCS-CTR(C7.),                        !COMPARE AT TARGET
8595      NEXT,   J/COMP351B
(4722) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...010.101.111)

```

```

8596
8597 4257:  !(FREE)
8598 COMP351B:

```

```

8599      P2-T,   D+CSPD(06)-XOR-ASPLO(R03), SAVE-D(C), !COMPARE EXPECTED:RECEIVED, BITWISE
8600      NEXT,   J/GOBUT351B                !CSP(06)=(054344)
(4257) DCS(0.00.0.0.0.0) BM(0110..10.00..10.01..101..111...0.1.0..0..0...0.1001...0..0000.0...11.000...010.110.000)

```

```

8601
8602 4260:  !(FREE)
8603 GOBUT351B:

```

```

8604      SETUP, RETURN/TEST351C,      !GO TO SUBR WHICH:
8605      NEXT,  CALL(DINTOIR-5)        !  D -> IR, BUT(INSTR5)
(4260) DCS(0.00.0.0.0.0) BM(0100..00.11..10.10..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

```

```

8606
8607 ! - - - - -
8608

```

```

8609 !TEST 351C CHECKS ASPHI RIF ADDR "010 0"=(04)
8610 4723:
8611 TEST351C:

```

```

8612      PO,      LOAD-ENUA(ZTARGET434),      !INSTRS FOR IR=(000000)
8613      LOAD-ERROR(TEST351C),                !ERROR DIRECTORY KEY
8614      DCS-CTR(C7.),                        !COMPARE AT TARGET
8615      BUMP-VERIFY,                          !COUNT
8616      NEXT,   J/COMP351C
(4723) DCS(1.00.1.0.0.1) BM(1000..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...010.110.001)

```

```

8618
8619 4261:  !(FREE)
8620 COMP351C:

```

```

8621      P2-T,   D+CSPD(06)-XOR-ASPHI(R04), SAVE-D(C), !COMPARE EXPECTED:RECEIVED, BITWISE
8622      NEXT,   J/GOBUT351C                !CSP(06)=(054344)
(4261) DCS(0.00.0.0.0.0) BM(0110..10.00..11.00..110..111...0.1.0..0..0...0.1001...0..0000.0...11.000...010.110.010)

```

```

8623
8624 4262:  !(FREE)
8625 GOBUT351C:

```

```

8626      SETUP, RETURN/TEST351D,      !GO TO SUBR WHICH:
8627      NEXT,  CALL(DINTOIR-5)        !  D -> IR, BUT(INSTR5)

```



(4262) DCS(0.00.0.0.0.0) BM(0100..00.11..10.01..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

8628  
8629  
8630  
8631  
8632  
8633  
8634  
8635  
8636  
8637  
8638  
8639  
8640  
8641  
8642  
8643  
8644  
8645  
8646  
8647  
8648  
8649  
8650  
8651  
8652  
8653  
8654  
8655  
8656  
8657  
8658  
8659  
8660  
8661  
8662  
8663  
8664  
8665  
8666  
8667  
8668  
8669  
8670  
8671  
8672  
8673  
8674

! - - - - -

!TEST 351D CHECKS ASPHI RIF ADDR "010 1"=(05)

4712:

TEST351D:

PO, LOAD-ENUA(ZTARGET434),  
LOAD-ERROR(TEST351D),  
DCS-CTR(C7.),  
NEXT J/COMP351D

!INSTRS FOR IR=(000000)  
!ERROR DIRECTORY KEY  
!COMPARE AT TARGET

(4712) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...010.110.011)

4263: !(FREE)

COMP351D:

PO, BUMP-VERIFY,  
P2-T, D+CSPD(05)-XOR-ASPHI(ROS), SAVE-D(C),  
NEXT J/GOBUT351D

!COUNT  
!COMPARE EXPECTED:RECEIVED, BITWISE  
!CSP(05)=(055255)

(4263) DCS(0.00.0.0.0.1) BM(0110..10.00..11.01..110..111...0.1.0..0..0...0.1010...0..0000.0...11.000...010.110.100)

4264: !(FREE)

GOBUT351D:

PO, BUMP-VERIFY,  
SETUP, RETURN/SCOPE351,  
NEXT CALL(DINTOIR-5)

!COUNT  
!GO TO SUBR WHICH:  
! D -> IR, BUT(INSTRS)

(4264) DCS(0.00.0.0.0.1) BM(0100..00.01..01.10..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

4265: !(FREE)

SCOPE351:

NEXT, BUTD(SCOPE1),  
J/TEST352A

!NO ERROR: "TEST352A" (+1. WORDS)  
!ERROR: "COMP351A" (-11. WORDS)

(4265) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...111.000.101)

! - - - - -

!TESTS 352 A-D VERIFY THAT THE RIF ADDRESS, WITH BSP HI/LO IMMEDO/1 MODES, OPERATES CORRECTLY

! - - - - -

!TEST 352A CHECKS BSPLO RIF ADDR "001 0"=(02)

4705:

TEST352A:

PO, LOAD-ENUA(ZTARGET434),  
LOAD-ERROR(TEST352A),  
DCS-CTR(C7.),  
NEXT J/COMP352A

!INSTRS FOR IR=(000000)  
!ERROR DIRECTORY KEY  
!COMPARE AT TARGET

(4705) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...111.001.100)

```

8675
8676 4714:
8677 COMP352A:
8678     P2-T, D+ASPLO(02)-XOR-BSPLO(R02), SAVE-D(C), !COMPARE EXPECTED:RECEIVED, BITWISE
8679     NEXT, J/G08UT352A !DATA=(055255)
(4714) DCS(0.00.0.0.0.0) BM(0110..00.10..10.00..101..111...0.1.0..0..0...0.0000...0..0000.0...11.000...010.110.110)

```

```

8680
8681 4266: !(FREE)
8682 G08UT352A:
8683     PO, BUMP-VERIFY, !COUNT
8684     SETUP, RETURN/TEST352B, !GO TO SUBR WHICH:
8685     NEXT, CALL(DINTOIR-5) ! D -> IR, BUT(INSTR5)
(4266) DCS(0.00.0.0.0.1) BM(0100..00.11..10.01..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

```

```

8686
8687 ! - - - - -
8688
8689

```

```

8690 !TEST 352B CHECKS BSPLO RIF ADDR "001 1"=(03)
8691 4713:
8692 TEST352B:
8693     PO, LOAD-ENUA(ZTARGET434), !INSTRS FOR IR=(000000)
8694     LOAD-ERROR(TEST352B), !ERROR DIRECTORY KEY
8695     DCS-CTR(C7.), !COMPARE AT TARGET
8696     NEXT, J/COMP352B
(4713) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...010.110.111)

```

```

8697
8698 4267: !(FREE)
8699 COMP352B:
8700     P2-T, D+ASPLO(03)-XOR-BSPLO(R03), SAVE-D(C), !COMPARE EXPECTED:RECEIVED, BITWISE
8701     NEXT, J/G08UT352B !DATA=(054344)
(4267) DCS(0.00.0.0.0.0) BM(0110..00.11..10.01..101..111...0.1.0..0..0...0.0000...0..0000.0...11.000...010.111.000)

```

```

8702
8703 4270: !(FREE)
8704 G08UT352B:
8705     SETUP, RETURN/TEST352C, !GO TO SUBR WHICH:
8706     NEXT, CALL(DINTOIR-5) ! D -> IR, BUT(INSTR5)
(4270) DCS(0.00.0.0.0.0) BM(0100..00.11..10.10..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

```

```

8707
8708 ! - - - - -
8709
8710

```

```

8711 !TEST 352C CHECKS BSPLO RIF ADDR "010 0"=(04)
8712 4724:
8713 TEST352C:
8714     PO, LOAD-ENUA(ZTARGET434), !INSTRS FOR IR=(000000)
8715     LOAD-ERROR(TEST352C), !ERROR DIRECTORY KEY
8716     DCS-CTR(C7.), !COMPARE AT TARGET
8717     BUMP-VERIFY, !COUNT
8718     NEXT, J/COMP352C
(4724) DCS(1.00.1.0.0.1) BM(1000..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...010.111.001)

```

```

8719
8720 4271: !(FREE)

```

```

8721 COMP352C:
8722 P2-T, D+ASPLO(04)-XOR-BSPLO(RO4), SAVE-D(C), !COMPARE EXPECTED:RECEIVED, BITWISE
8723 NEXT, J/GOBUT352C !DATA=(053433)
(4271) DCS(0.00.0.0.0.0) BM(0110..00.10..10.00..110..111...0.1.0..0..0...0.0000...0..0000.0...11.000...010.111.010)

8724 4272: !(FREE)
8725 GOBUT352C:
8726 SETUP, RETURN/TEST3520, !GO TO SUBR WHICH:
8727 NEXT, CALL(DINTOIR-5) ! D -> IR, BUT(INSTRS)
8728 (4272) DCS(0.00.0.0.0.0) BM(0100..00.11..10.10..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

8729
8730 ! - - - - -
8731 !
8732 !
8733 !TEST 3520 CHECKS BSPLO RIF ADDR "010 1"=(05)
8734 4275:
8735 TEST3520:
8736 PO, LOAD-ENUA(ZTARGET434), !INSTRS FOR IR=(000000)
8737 LOAD-ERROR(TEST3520), !ERROR DIRECTORY KEY
8738 DCS-CTR(C7.), !COMPARE AT TARGET
8739 NEXT, J/COMP3520
(4275) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...010.111.011)

8740 4273: !(FREE)
8741 COMP3520:
8742 PO, BUMP-VERIFY, !COUNT
8743 P2-T, D+ASPLO(05)-XOR-BSPLO(RO5), SAVE-D(C), !COMPARE EXPECTED:RECEIVED, BITWISE
8744 NEXT, J/GOBUT3520 !DATA=(052522)
8745 (4273) DCS(0.00.0.0.0.1) BM(0110..00.11..10.01..110..111...0.1.0..0..0...0.0000...0..0000.0...11.000...010.111.100)

8746 4274: !(FREE)
8747 GOBUT3520:
8748 PO, BUMP-VERIFY, !COUNT
8749 SETUP, RETURN/SCOPE352, !GO TO SUBR WHICH:
8750 NEXT, CALL(DINTOIR-5) ! D -> IR, BUT(INSTRS)
8751 (4274) DCS(0.00.0.0.0.1) BM(0100..00.01..01.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

8752
8753
8754 4275: !(FREE)
8755 SCOPE352:
8756 P2-T, D+CSPD(D13), D(C)+0, !RESTORE CONSTANT (000000)
8757 P3, A#BSPHI(01)+0, !
8758 NEXT, BUT(SCOPE), !NO ERROR: "RESTORED1" (+1. WORDS)
8759 J/RESTORED1 ! ERROR: "COMP352A" (-11. WORDS)
8760 (4275) DCS(0.00.0.1.0.0) BM(1010..10.11..00.01..100..000...0.1.0..0..0...0.0100...0..1011.0...11.000...111.001.101)

8761
8762 ! - - - - -
8763 !
8764 !
8765 ! THIS SECTION OF CODE FINISHES RESTORING THE 4 CONSTANTS IN ASPHI/BSPHI
8766 ! THAT WERE WIPED OUT IN THE PREVIOUS GROUP OF TESTS.

```

```

8767
8768 4715:
8769 RESTORE01:
8770 P2-T, D+CSPD(D10), !(052525)
8771 P3, A#BSPHI[07]+D,
8772 NEXT, GOTO-PAGE(7),
8773 J/RESTORE02
(4715) DCS[0.00.0.0.0.0] BM[1010..10.11..00.01..111..111...0.1.0..0..0...0.0111...0..1011.0...11.100...000.011.000]

8774
8775 7030: !(FREE)
8776 RESTORE02:
8777 P2-T, D+CSPD(D11), D[C]+0, !(125252)
8778 P3, A#BSPHI[05]+D,
8779 NEXT, J/RESTORE03
(7030) DCS[0.00.0.0.0.0] BM[1010..10.11..00.01..110..000...0.1.0..0..0...0.0110...0..1011.0...11.000...000.011.010]

8780
8781 7032: !(FREE)
8782 RESTORE03:
8783 P2-T, D+CSPD(D12), D[C]+0, !(177777)
8784 P3, A#BSPHI[03]+D,
8785 NEXT, J/RESTORE04
(7032) DCS[0.00.0.0.0.0] BM[1010..10.11..00.01..101..000...0.1.0..0..0...0.0101...0..1011.0...11.000...000.011.011]

8786
8787 7033: !(FREE)
8788 RESTORE04:
8789 SETUP, RETURN/TEST361A, !GO TO SUBR THAT RESTORES CURRENT
8790 NEXT, GOTO-PAGE(7), !DCS MICROCODE VERSION NUMBER
8791 J/INSERTREVN0 !B<15>=(0), INTO B.M. GPR "RS"
(7033) DCS[0.00.0.0.0.0] BM[0110..00.10..11.11..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.001.110]

```

```

8792
8793
8794
8795
8796
8797 !.PAGE=====
8798
8799

```

```

8800 .TOC * TEST361-372: TESTING SR, GUARD, RES, AND XMUX
8801
8802 !*****
8803 !*
8804 !* TESTS: 361 - 372 UWORDS: 125 + 117
8805 !*
8806 !* FUNCTIONS: TESTS THAT SR CAN BE LOADED FROM ALU OUT, READ THRU
8807 !* SR AND FLOAT PORTS OF XMUX, SR CAN BE SHIFTED LEFT, RIGHT,
8808 !* AND NOP, GUARD-ENABLED, DISABLED, SHIFTED, AND THAT
8809 !* ANY "BUTTABLE" BITS CAN BE TESTED.
8810 !* PREVIOUSLY TESTED IN TEST 105 A-E WAS THE ABILITY TO
8811 !* LOAD/READ THE SR THRU XMUX/SR, AND BUT(SR3-0).
8812 !* THESE FUNCTIONS ARE NOT TESTED HERE.
8813 !*
8814 !*****
8815

```

```

8816
8817
8818
8819
8820
8821 ! -----
8822
8823 !*** TEST 361A ***
8824 !TEST-361A TESTS THAT SR CAN BE LOADED, GUARD CLEARED, WITH PATTERN (052525)(0);
8825 !READ THRU XMUX-FLTPT PORT = (100125)
8826 6573:
8827 TEST361A:
8828     PO,          LOAD-ENUA(ZTARGET434),          !SETUP FOR IR=(000000)/INSTR5 TEST
8829     P2-T,        LOAD-ERROR(TEST361A),          !ERROR DIRECTORY KEY
8830     P3,          DCS-CTR(C11.),                !COMPARE ENUA:TNUA AT TARGET
8831     NEXT,        J/SETRES361A
(6573) DCS(1.00.1.0.0.0) BM(0100..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...101.111.000)

8832
8833 6570:
8834 SETRES361A:
8835     P3          CSPD(16)+EMIT                  !CSP GETS
8836     EMITC,      SENDMUX-4567-SEL             !RES REG VALUES
8837     SR-LOAD,    GUARD-EN,
8838     NEXT,        J/LOADRES361A
(6570) DCS(0.00.0.0.0.0) BM(0000..10.10..00.00..000..000...0.0.0..0..0...0.0001...1..0000.0...11.000...000.000.001)

8839
8840 6001: !(FREE)
8841 LOADRES361A:
8842     PO,          BUMP-VERIFY,                  !COUNT
8843     P2-T,        RES+CSPB(816)                !STORE RES
8844     D+ASPHI(C000000), D(C)+1,                !SET D=ZER0ES, D(C)=1
8845     NEXT,        J/LOADSR361A
(6001) DCS(0.00.0.0.0.1) BM(1111..11.01..11.01..100..000...0.1.0..0..0...0.0000...0..1000.1...11.000...000.000.010)

8846
8847 6002: !(FREE)
8848 LOADSR361A:
8849     P2-T,        SR+BSPHI(C052525),          !LOAD SR WITH DATA
8850     NEXT,        J/EXPEC361B
(6002) DCS(0.00.0.0.0.0) BM(1010..01.11..00.00..111..000...0.0.1..0..0...0.0000...0..0000.0...11.000...000.000.011)

8851
8852 6003: !(FREE)
8853 EXPEC361B:
8854     PO,          BUMP-VERIFY,                  !COUNT
8855     P3,          CSPD(16)+EMIT,              !EXPECTED VALUE OUT OF XMUX-FLOAT
8856     EMIT/100125,                ! (100125)
8857     NEXT,        J/COMP361B
(6003) DCS(0.00.0.0.0.1) BM(1000..10.00..00.01..010..101...0.0.0..0..0...0.0001...1..0000.0...11.000...000.000.100)

8858
8859 6004: !(FREE)
8860 COMP361B:
8861     SETUP,      RETURN/TEST361D,             !RETURN TO START OF NEXT SUBTEST
8862     NEXT,        CALL(CSP16XORFLTTOIR-5)     !SUBR: CSP(16).XOR.FLOAT -> IR, BUT(INSTR5)
(6004) DCS(0.00.0.0.0.0) BM(0110..00.11..11.11..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.011.101)

```

```

8863
8864
8865
8866
8867
8868 !-----
8869
8870 !*** TEST 361D ***
8871 !TEST 361D CHECKS THAT THE BUT ON SR<1:0> READS THE (052525) IN THE SR CORRECTLY
8872 6771:
8873 TEST361D:
8874     PO,          LOAD-ENUA(ZTARGET403),          !EXPECTED VALUE "01"#"1" IN SR<1:0>
8875                 LOAD-ERROR(TEST361D),          !ERROR DIRECTORY KEY
8876                 DCS-CTR(C3.),                  !COMPARE ENUA:TNUA AT TARGET
8877     NEXT,        J/GOBUT361D
(6771) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...000.000.101)

8878
8879 6005: !(FREE)
8880 GOBUT361D:
8881     SETUP,       RETURN/TEST361E,              !RETURN TO START OF NEXT SUBTEST
8882     NEXT,        GOTO-PAGE(7),                 !BUT'S ARE ON PAGE 7
8883     J/BUTSR1-0   !GO BUT ON SR<1:0>
(6005) DCS(0.00.0.0.0.0) BM(0110..00.11..11.10..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.101)

8884
8885
8886
8887
8888
8889 !-----
8890
8891 !*** TEST 361E ***
8892 !TEST-361E CHECKS THAT GD<3:2> IS CLEARED UNDER SR-LOAD, GUARD-ENABLED
8893 6767:
8894 TEST361E:
8895     PO,          LOAD-ENUA(ZTARGET400),          !EXPECTED VALUE "00" IN GUARD<3:2>
8896                 LOAD-ERROR(TEST361E),          !ERROR DIRECTORY KEY
8897                 DCS-CTR(C3.),                  !COMPARE ENUA:TNUA AT TARGET
8898                 BUMP-VERIFY,                   !COUNT
8899     NEXT,        J/GOBUT361E
(6767) DCS(1.00.1.0.0.1) BM(1100..00.11..11.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...000.000.110)

8900
8901 6006: !(FREE)
8902 GOBUT361E:
8903     SETUP,       RETURN/TEST362A,              !RETURN TO START OF NEXT SUBTEST
8904     NEXT,        GOTO-PAGE(7),                 !BUT'S ARE ON PAGE 7
8905     J/BUTGD3-2   !GO BUT ON GD<3:2>
(6006) DCS(0.00.0.0.0.0) BM(0110..00.11..11.10..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.100)

8906
8907
8908
8909
8910

```

```

8911 ! -----
8912
8913 !*** TEST 362A ***
8914 !TEST-362A CHECKS THAT THE SR#GO CAN BE SHIFTED RIGHT 1 POSITION, INSERTING
8915 !A "1" FROM D<00> INTO SR<15>, SHIFTING THE "1" IN SR<00> INTO GO<3>
8916 6765:
8917 TEST362A:
8918     PO,          LOAD-ENUA(ZTARGET434),          !SETUP FOR IR=(000000)/INSTR5 TEST
8919     LOAD-ERROR(TEST362A),          !ERROR DIRECTORY KEY
8920     DCS-CTR(C10.),          !COMPARE ENUA:TNUA AT TARGET
8921     NEXT          J/SETRES362A
(6765) DCS(1.00.1.0.0.0) BM(0101..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...000.000.111)

8922 6007: !(FREE)
8923 SETRES362A:
8924     PO,          BUMP-VERIFY,          !COUNT
8925     P3,          CSPD(16)+EMIT,          !CSP GETS
8926     EMITC,       SENDMUX-4567-SEL,          !RES REG VALUES
8927     SR-RIGHT,   GUARD-EN,
8928     NEXT          J/SETDDC362A
(6007) DCS(0.00.0.0.0.1) BM(0010..10.10..00.00..000..000...0.0.0..0..0...0.0001...1..0000.0...11.000...000.001.000)

8930 6010: !(FREE)
8931 SETDDC362A:
8932     P2-T,       D+BSPHI(C000001), D(C)+0,          !SETUP D<00>=1 FOR SHIFT
8933     NEXT          J/LOADRES362A
(6010) DCS(0.00.0.0.0.0) BM(1010..01.11..00.00..000..000...0.1.0..0..0...0.0000...0..0000.0...11.000...000.001.001)

8935 6011: !(FREE)
8936 LOADRES362A:
8937     P2,          RES+CSP8(B16),          !STORE RES
8938     P3-T,       SR+SR-RIGHT-1,          !SHIFT SR RIGHT, GUARD-ENABLED
8939     NEXT          J/COMP362A
(6011) DCS(0.00.0.0.0.0) BM(0000  :1.01..00.00..000..000...1.0.1..0..0...0.0000...0..1000.1...11.000...000.001.010)

8941 6012: !(FREE)
8942 COMP362A:
8943     PO,          BUMP-VERIFY,          !COUNT
8944     P2-T,       D+SR-XOR-BSPHI(C125252), SAVE-D(C),          !D + (125252)=SR, BITWISE
8945     NEXT          J/GOBUT362A
(6012) DCS(0.00.0.0.0.1) BM(0110..01.11..00.00..110..111...0.1.0..0..0...0.0000...0..0000.0...11.000...000.001.011)

8947 6013: !(FREE)
8948 GOBUT362A:
8949     SETUP,       RETURN/TEST362B,          !RETURN TO START OF NEXT SUBTEST
8950     NEXT          CALL(DINTOIR-5)          !SUBR: D -> IR, BUT(INSTR5)
8951 (6013) DCS(0.00.0.0.0.0) BM(0110..00.11..11.10..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

8952
8953
8954
8955
8956
8957 ! -----

```

```

8958
8959 !*** TEST 3628 ***
8960 !TEST-3628 CHECKS THAT SR-XMUX-FLOAT PORT CAN BE READ, WITH SR=(125252)
8961 6763:
8962 TEST3628:
8963     PO,          LOAD-ENUA(ZTARGET434),      !SETUP FOR IB=(000000)/INSTR5 TEST
8964     LOAD-ERROR(TEST3628),                    !ERROR DIRECTORY KEY
8965     DCS-CTR(C8.),                             !COMPARE ENUA:TNUA AT TARGET
8966     NEXT,      J/EXPEC3628
(6763) DCS(1.00.1.0.0.0) BM(0111..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...000.001.100)

8967
8968 6014: !(FREE)
8969 EXPEC3628:
8970     PO,          BUMP-VERIFY,                !COUNT
8971     P3,          CSPD(16)+EMIT,              !EXPECTED VALUE OF XMUX-FLOAT
8972     EMIT/000052,                               ! (000052)
8973     NEXT,      J/COMP3628
(6014) DCS(0.00.0.0.0.0.1) BM(0000..10.00 .00.00..101..010...0.0.0..0..0...0.0001...1..0000.0...11.000...000.001.101)

8974
8975 6015: !(FREE)
8976 COMP3628:
8977     SETUP,      RETURN/TEST3620,              !RETURN TO START OF NEXT SUBTEST
8978     NEXT,      CALL(CSP16XORFLTTOIR-5)        !SUBR: CSP(16).XOR.FLOAT -> IR, BUT(INSTR5)
(6015) DCS(0.00.0.0.0.0) BM(0110..00.11..11.01..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.011.101)

8979
8980
8981
8982
8983
8984 !-----
8985
8986 !*** TEST 3620 ***
8987 !TEST-3620 CHECKS THAT THE BUT ON SR<1:0> READS THE (125252) IN THE SR CORRECTLY
8988 6757:
8989 TEST3620:
8990     PO,          LOAD-ENUA(ZTARGET405),      !EXPECTED VALUE "10"#"1" IN SR<1:0>
8991     LOAD-ERROR(TEST3620),                    !ERROR DIRECTORY KEY
8992     DCS-CTR(C3.),                             !COMPARE ENUA:TNUA AT TARGET
8993     NEXT,      J/GOBUT3620
(6757) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...000.001.110)

8994
8995 6016: !(FREE)
8996 GOBUT3620:
8997     SETUP,      RETURN/TEST362E,              !RETURN TO START OF NEXT SUBTEST
8998     NEXT,      GOTO-PAGE(7),                  !BUT'S ARE ON PAGE 7
8999     J/BUTSR1-0,                                !GO BUT ON SR<1:0>
(6016) DCS(0.00.0.0.0.0) BM(0110..00.11..11.01..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.101)

9000
9001
9002
9003
9004

```



```

9005 ! -----
9006
9007 !*** TEST 362E ***
9008 !TEST-362E CHECKS THAT GD<3:2> RECEIVED THE "1" IN SR<00> AFTER THE SHIFT RIGHT
9009 6755:
9010 TEST362E:
9011     PD,      LOAD-ENUA(ZTARGET402),      !EXPECTED VALUE "10" IN GUARD<3:2>
9012     LOAD-ERROR(TEST362E),      !ERROR DIRECTORY KEY
9013     DCS-CTR(C3.),      !COMPARE ENUA:TNUA AT TARGET
9014     NEXT,     J/GOBUT362E
(6755) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...000.001.111)

9015 6017: !(FREE)
9016 GOBUT362E:
9017     SETUP,   RETURN/TEST363A,      !RETURN TO START OF NEXT SUBTEST
9018     PD,      BUMP-VERIFY,      !COUNT
9019     NEXT,     GOTO-PAGE(7),      !BUT'S ARE ON PAGE 7
9020     J/BUTGD3-2,      !GO BUT ON GD<3:2>
9021 (6017) DCS(0.00.0.0.0.1) BM(0110..00.11..11.01..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.100)

9022
9023
9024
9025
9026
9027 ! -----
9028
9029 !*** TEST 363A ***
9030 !TEST-363A CHECKS THAT THE SR#GD CAN AGAIN BE SHIFTED RIGHT 1 POSITION, INSERTING
9031 !A "0" FROM D<00> INTO SR<15>, SHIFTING THE "0" IN SR<00> INTO GD<3>
9032 6753:
9033 TEST363A:
9034     PD,      LOAD-ENUA(ZTARGET434),      !SETUP FOR IR=(000000)/INSTRS TEST
9035     LOAD-ERROR(TEST363A),      !ERROR DIRECTORY KEY
9036     DCS-CTR(C9.),      !COMPARE ENUA:TNUA AT TARGET
9037     NEXT,     J/SETDOC363A
(6753) DCS(1.00.1.0.0.0) BM(0110..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...000.010.000)

9038 6020: !(FREE)
9039 SETDOC363A:
9040     P3-T,     D+NOT-BSPHI(C000001), D(C)+1,      !SETUP D=(177776), D(C)=0 FOR SHIFT RIGHT
9041     NEXT,     J/SHIFT363A
(6020) DCS(0.00.0.0.0.0) BM(0000..00.00..11.01..000..000...1.1.0..0..0...0.0000...0..0000.0...11.000...000.010.001)

9043 6021: !(FREE)
9044 SHIFT363A:
9045     PD,      BUMP-VERIFY,      !COUNT
9046     P2-T,     SR+SR-RIGHT-1,      !SHIFT SR RIGHT, GUARD-ENABLED
9047     NEXT,     J/COMP363A
(6021) DCS(0.00.0.0.0.1) BM(0000..00.00..00.00..000..000...0.0.1..0..0...0.0000...0..0000.0...11.000...000.010.010)

9049 6022: !(FREE)
9050 COMP363A:
9051     P2-T,     D+SR-XOR-BSPHI(C052525), SAVE-D(C),      !D + (052525)=SR, BITWISE
9052

```

9053 (6022) DCS(0.00.0.0.0.0) BM(0110..01.11..00.00..111..111!...0.1.0..0..0...0.0000...0..0000.0...11.000...000.010.011)

9054 6023: !(FREE)  
9055 GOBUT363A:

9057 SETUP, RETURN/TEST3638, !RETURN TO START OF NEXT SUBTEST  
9058 NEXT, CALL(DINTOIR-5) !SUBR: 0 -> IR, BUT(INSTR5)  
(6023) DCS(0.00.0.0.0.0) BM(0110..00.11..11.01..001..111!...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

9059  
9060  
9061  
9062  
9063  
9064 !-----  
9065

9066 !\*\*\* TEST 3638 \*\*\*  
9067 !TEST-3638 CHECKS THE GUARD REGISTER WAS SHIFTED FIGHT ALSO

9068 6751:  
9069 TEST3638:  
9070 PO, LOAD-ENUA(ZTARGET401), !EXPECTED VALUE "01" IN GUARD<3:2>  
9071 LOAD-ERROR(TEST3638), !ERROR DIRECTORY KEY  
9072 DCS-CTR(C3.), !COMPARE ENUA:TNUA AT TARGET  
9073 NEXT, J/GOBUT3638

(6751) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..001!...0.0.0..0..0...0.0000...0..0000.0...11.000...000.010.100)

9074 6024: !(FREE)  
9075 GOBUT3638:  
9076 SETUP, RETURN/TEST364A, !RETURN TO START OF NEXT SUBTEST  
9077 NEXT, GOTO-PAGE(7), !BUT'S ARE ON PAGE 7  
9078 J/BUTGD3-2 !GO BUT ON GD<3:2>  
9079

(6024) DCS(0.00.0.0.0.0) BM(0110..00.11..11.00..111..111!...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.100)

9080  
9081  
9082  
9083  
9084  
9085 !-----  
9086

9087 !\*\*\* TEST 364A \*\*\*  
9088 !TEST-364A/364B CHECK THAT WE ARE ABLE TO FILTER A "1" DOWN TO GD<0>, AND  
9089 !SUBSEQUENTLY ABLE TO RECOVER IT, SO GO<3:0>="0101"

9090 6747:  
9091 TEST364A:  
9092 PO, LOAD-ENUA(ZTARGET402), !EXPECTED VALUE "10" IN GUARD <3:2>  
9093 LOAD-ERROR(TEST364A), !ERROR DIRECTORY KEY  
9094 DCS-CTR(C4.), !COMPARE ENUA:TNUA AT TARGET  
9095 BUMP-VERIFY, !COUNT  
9096 NEXT, J/SETDCC364A

(6747) DCS(1.00.1.0.0.1) BM(1011..00.11..11.00..000..010!...0.0.0..0..0...0.0000...0..0000.0...11.000...000.010.101)

9097 6025: !(FREE)  
9098 SETDCC364A:  
9099

9100 P2-T, D+NOT-ASPHI(C000001), D(C)+1, !SETUP D=(177776), D(C)=0 FOR SHIFT RIGHT  
 9101 NEXT, J/GOBUT364A ! WHERE SR<15> <- (0)  
 (6025) DCS(0.00.0.0.0.0) BM(0000..00.00..11.01..000..000...0.1.0..0..0...0.0000...0..0000.0...11.000...000.C10.110)

9102 6026: !(FREE)  
 9103 GOBUT364A:  
 9104 SETUP, RETURN/TEST364B, !RETURN TO START OF NEXT SUBTEST  
 9105 P3-T, SR+SR-RIGHT-1, !SHIFT SR#GD RIGHT  
 9106 NEXT, GOTO-PAGE(7), !BUT'S ARE ON PAGE 7  
 9107 J/BUTGD3-2 !GO BUT ON GD<3:2>  
 9108 (6026) DCS(0.00.0.0.0.0) BM(0110..00.11..11.00..101..111...1.0.1..0..0...0.0000...0..0000.0...11.100...011.001.100)

9109  
 9110  
 9111  
 9112  
 9113

9114 !-----  
 9115  
 9116 !\*\*\* TEST 364B \*\*\*  
 9117 !TEST-364B CHECKS THAT GD<3:2> READS AS "01"

9118 6745:  
 9119 TEST364B:  
 9120 PO, LOAD-ENUA(ZTARGET401), !EXPECTED VALUE "01" IN GUARD<3:2>  
 9121 LOAD-ERROR(TEST364B), !ERROR DIRECTORY KEY  
 9122 DCS-CTR(C4.), !COMPARE ENUA:TNUA AT TARGET  
 9123 NEXT, J/SETDCC364B  
 (6745) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...000.010.111)

9124 6027: !(FREE)  
 9125 SETDCC364B:  
 9126 P3-T, D+BSPHI(C000001), D(C)+0, !SETUP D, D(C) FOR SR<15>+"1"  
 9127 NEXT, J/GOBUT364B  
 9128 (6027) DCS(0.00.0.0.0.0) BM(1010..01.11..00.00..000..000...1.1.0..0..0...0.0000...0..0000.0...11.000...000.011.000)

9129 6030: !(FREE)  
 9130 GOBUT364B:  
 9131 SETUP, RETURN/TEST365A, !RETURN TO START OF NEXT SUBTEST  
 9132 P2-T, SR+SR-RIGHT-1, !SHIFT SR#GD RIGHT AGAIN  
 9133 NEXT, GOTO-PAGE(7), !BUT'S ARE ON PAGE 7  
 9134 J/BUTGD3-2 !GO BUT ON GD<3:2>  
 9135 (6030) DCS(0.00.0.0.0.0) BM(0110..00.11..11.00..011..111...0.0.1..0..0...0.0000...0..0000.0...11.100...011.001.100)

9136  
 9137  
 9138  
 9139  
 9140

9141 !-----  
 9142  
 9143 !\*\*\* TEST 365A \*\*\*  
 9144 !TEST-365A CHECKS THAT CAN SHIFT SR-RIGHT, W/GUARD-DISABLED  
 9145 6743:  
 9146 TEST365A:

```

9147      PO,      LOAD-ENUA(ZTARGET434),      !SETUP FOR IR=(000000)/INSTRS TEST
9148      LOAD-ERROR(TEST365A),      !ERROR DIRECTORY KEY
9149      DCS-CTR(C10.),      !COMPARE ENUA:TNUA AT TARGET
9150      NEXT,      J/SETRES365A
(6743) DCS(1.00.1.0.0.0) BM(0101..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...000.011.001)

```

```

9151      6031:  !(FREE)
9152      SETRES365A:
9154      PO,      BUMP-VERIFY      !COUNT
9155      P3,      CSPD(16)+EMIT      !CSP GETS
9156      EMITC,    SENDMUX-4567-SEL      !RES REG VALUES
9157      SR-RIGHT GUARD-DIS,
9158      NEXT,      J/LOADRES365A
(6031) DCS(0.00.0.0.0.0) BM(0010..10.00..00.00..000..000...0.0.0..0..0...0.0001...1..0000.0...11.000...000.011.010)

```

```

9159      6032:  !(FREE)
9160      LOADRES365A:
9162      P2,      RES+CSPB(816),      !STORE RES
9163      P3-T,    SR+SR-RIGHT-1,      !SHIFT SR RIGHT, GUARD-DISABLED
9164      NEXT,      J/EXPEC365A
(6032) DCS(0.00.0.0.0.0) BM(0000..11.01..00.00..000..000...1.0.1..0..0...0.0000...0..1000.1...11.000...000.011.011)

```

```

9165      6033:  !(FREE)
9166      EXPEC365A:
9168      PO,      BUMP-VERIFY      !COUNT
9169      P3,      CSPD(16)+EMIT,      !EXPECTED VALUE OUT OF XMUX-SR
9170      EMIT/145252,      !(145252)
9171      NEXT,      J/COMP365A
(6033) DCS(0.00.0.0.0.0) BM(1100..10.10..10.10..101..010...0.0.0..0..0...0.0001...1..0000.0...11.000...000.011.100)

```

```

9172      6034:  !(FREE)
9173      COMP365A:
9175      SETUP,    RETURN/TEST3658      !RETURN TO START OF NEXT SUBTEST
9176      NEXT,      CALL(CSP16XORSRTOIR-5)      !SUBR: CSP(16).XOR.SR -> IR, BUT(INSTRS)
(6034) DCS(0.00.0.0.0.0) BM(0110..00.11..11.00..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.011.001)

```

```

9177
9178
9179
9180
9181
9182      ! -----
9183

```

```

9184      !*** TEST 3658 ***
9185      !TEST-3658 CHECKS THAT THE GUARD WASN'T ALTERED ON SHIFT RIGHT/DISABLED

```

```

9186      6741:
9187      TEST3658:
9188      PO,      LOAD-ENUA(ZTARGET401),      !EXPECTED VALUE "01" IN GUARD<3:2>
9189      LOAD-ERROR(TEST3658),      !ERROR DIRECTORY KEY
9190      DCS-CTR(C3.),      !COMPARE ENUA:TNUA AT TARGET
9191      NEXT,      J/GOBUT3658
(6741) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...000.011.101)

```

```

9192      6035:  !(FREE)
9193

```

```

9194 GOBUT365B:
9195     SETUP, RETURN/TEST366A,      !RETURN TO START OF NEXT SUBTEST
9196     NEXT,  GOTO-PAGE(7),        !BUT'S ARE ON PAGE 7
9197           J/BUTG03-2           !GO BUT ON GD<3:2>
(6035) DCS(0.00.0.0.0.0) BM(0110..00.11..10.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.100)

9198
9199
9200
9201
9202
9203 !-----
9204
9205 !*** TEST 366A ***
9206 !TEST-366A CHECKS THAT SR CAN SHIFT LEFT, GUARD-DISABLED & NOT ALTERED, "0" IN
9207 !D(C) SHIFTED INTO SR<00>
9208 6737:
9209 TEST366A:
9210     PO,    LOAD-ENUA(ZTARGET434), !SETUP FOR IR=(000000)/INSTRS TEST
9211           LOAD-ERROR(TEST366A), !ERROR DIRECTORY KEY
9212           DCS-CTR(C11.),        !COMPARE ENUA:TNUA AT TARGET
9213     NEXT  J/SETRES366A
(6737) DCS(1.00.1.0.0.0) BM(0100..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...000.011.110)

9214
9215 6036: !(FREE)
9216 SETRES366A:
9217     PO,    BUMP-VERIFY           !COUNT
9218     P3,    CSPD(16)+EMIT        !CSP GETS
9219     EMITC, SENDMUX-4567-SEL    !RES REG VALUES
9220           SR-LEFT, GUARD-DIS,
9221     NEXT  J/LOADRES366A
(6036) DCS(0.00.0.0.0.1) BM(0001..10.00..00.00..000..000...0.0.0..0..0...0.0001...1..0000.0...11.000...000.011.111)

9222
9223 6037: !(FREE)
9224 LOADRES366A:
9225     P2-T,  RES+CSPB(B16)        !STORE RES
9226           D+ZERO, D(C)+ALU07,  !AND SET D, D(C)
9227     NEXT  J/SHIFT366A
(6037) DCS(0.00.0.0.0.0) BM(0011..11.01..00.00..000..011...0.1.0..0..0...0.0000...0..1000.1...11.000...000.100.000)

9228
9229 6040: !(FREE)
9230 SHIFT366A:
9231     PO,    BUMP-VERIFY           !COUNT
9232     P3-T,  SR+SR-LEFT-1,       !SHIFT SR LEFT 1, SR<00>+D(C)="0"
9233     NEXT  J/EXPEC366A
(6040) DCS(0.00.0.0.0.1) BM(0000..00.00..00.00..000..000...1.0.1..0..0...0.0000...0..0000.0...11.000...000.100.001)

9234
9235 6041: !(FREE)
9236 EXPEC366A:
9237     P3,    CSPD(16)+EMIT,       !EXPECTED RESULT AFTER SHIFT
9238           EMIT/112524,        !(112524)
9239     NEXT  J/COMP366A
(6041) DCS(0.00.0.0.0.0) BM(1001..10.01..01.01..010..100...0.0.0..0..0...0.0001...1..0000.0...11.000...000.100.010)

9240

```

```

9241 6042: !(FREE)
9242 COMP366A:
9243     SETUP, RETURN/TEST366B, !RETURN TO START OF NEXT SUBTEST
9244     NEXT, CALL(CSP16XORSR->IR-5) !SUBR: CSP(16).XOR.SR -> IR, BUT(INSTRS)
(6042) DCS(0.00.0.0.0.0) BM(0110..00.11..10.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.011.001)

```

```

9245
9246
9247
9248
9249
9250
9251
9252

```

! - - - - -

```

9253 !*** TEST 366B ***
9254 !TEST-366B CHECKS THAT GUARD WASN'T ALTERED ON SHIFT LEFT/GUARD-DISABLED
9255 6735:
9256 TEST366B:
9257     PO, LOAD-ENUA(ZTARGET401), !GUARD SHOULD STILL BE "01"#"01"
9258     LOAD-ERROR(TEST366B), !ERROR DIRECTORY KEY
9259     DCS-CTR(C3.), !COMPARE ENUA:TNUA AT TARGET
9260     NEXT, J/GOBUT366B
(6735) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...000.100.011)

```

```

9261 6043: !(FREE)
9262 GOBUT366B:
9263     SETUP, RETURN/TEST366C, !RETURN TO START OF NEXT SUBTEST
9264     NEXT, GOTO-PAGE(7), !BUT'S ARE ON PAGE 7
9265     J/BUTGD3-2 !GO BUT ON GUARD <3:2>
9266 (6043) DCS(0.00.0.0.0.0) BM(0110..00.11..10.11..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.100)

```

```

9267
9268
9269
9270
9271
9272
9273
9274

```

! - - - - -

```

9275 !*** TEST 366C ***
9276 !TEST-366C CHECKS THAT SR CAN AGAIN SHIFT LEFT, GUARD-DISABLED, SR<00>+D[C]="1"
9277 6733:
9278 TEST366C:
9279     PO, LOAD-ENUA(ZTARGET434), !SETUP FOR IR=(000000)/INSTRS TEST
9280     LOAD-ERROR(TEST366C), !ERROR DIRECTORY KEY
9281     DCS-CTR(C10.), !COMPARE ENUA:TNUA AT TARGET
9282     NEXT, J/SETDOC366C
(6733) DCS(1.00.1.0.0.0) BM(0101..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...000.100.100)

```

```

9283 6044: !(FREE)
9284 SETDOC366C:
9285     P2-T, D+ASPHI(C177777), D[C]+1, !SETUP D, D[C]
9286     NEXT, J/SHIFT366C
9287 (6044) DCS(0.00.0.0.0.0) BM(1111..00.00..11.01..101..000...0.1.0..0..0...0.0000...0..0000.0...11.000...000.100.101)

```

```

9288
9289 6045: !(FREE)
9290 SHIFT366C:
9291 P2-T, SR+SR-LEFT-1, !SHIFT SR LEFT 1, SR<00>*D(C)="1"
9292 NEXT, J/EXPEC366C
(6045) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.1..0..0...0.0000...0..0000.0...11.000...000.100.110)

9293
9294 6046: !(FREE)
9295 EXPEC366C:
9296 P0, BUMP-VERIFY !COUNT
9297 P3, CSPD(16)+EMIT, !EXPECTED VALUE AFTER SHIFT
9298 EMIT/025251, !(025251)
9299 NEXT, J/COMP366C
(6046) DCS(0.00.0.0.0.1) BM(0010..10.10..10.10..101..001...0.0.0..0..0...0.0001...1..0000.0...11.000...000.100.111)

9300
9301 6047: !(FREE)
9302 COMP366C:
9303 SETUP, RETURN/TEST367A, !RETURN TO START OF NEXT SUBTEST
9304 NEXT, CALL(CSP16XORSRTOIR-5) !SUBR: CSP(16).XOR.SR -> IR, BUT(INSTRS)
(6047) DCS(0.00.0.0.0.0) BM(0110..00.11..10.11..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.011.001)

9305
9306
9307
9308
9309
9310
9311 !-----
9312
9313 !*** TEST 367A ***
9314 !TEST-367A CHECKS THAT LOAD/GUARD-DISABLED LOADS SR, GUARD NOT ALTERED
9315 !CHECK THAT BUTA(CLEAR-FLAGS) CLEARS RES TO SR-LOAD/GUARD-DIS
9316 6731:
9317 TEST367A:
9318 P0, LOAD-ENUA(ZTARGET401), !GUARD SHOULD STILL BE "01"*"01" AFTER LOAD
9319 LOAD-ERROR(TEST367A), !ERROR DIRECTORY KEY
9320 DCS-CTR(C6.), !COMPARE ENUA:TNUA AT TARGET
9321 NEXT, J/SETRES367A
(6731) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...000.101.000)

9322
9323 6050: !(FREE)
9324 SETRES367A:
9325 P0, BUMP-VERIFY !COUNT
9326 P3, CSPD(16)+EMIT, !CSP GETS
9327 EMITC, SENDMUX-4567-SEL, !RES VALUES
9328 SR-NOP GUARD-EN, !FIRST LOAD COMPLEMENT OF THOSE AFTER BUTA(CLEAR-FLAGS)
9329 NEXT, J/LOADRES367A
(6050) DCS(0.00.0.0.0.1) BM(0011..10.10..00.00..000..000...0.0.0..0..0...0.0001...1..0000.0...11.000...000.101.001)

9330
9331 6051: !(FREE)
9332 LOADRES367A:
9333 P0, BUMP-VERIFY !COUNT
9334 P2-T, RES+CSPB(816), !STORE RES
9335 D+ZERO, D(C)+ALU07, !SET D, D(C)

```

```

9336          P3,      BUTA(CLR-FLAG-RES-UCON),      !RESET RES TO SR-LOAD, GUARD-DIS
9337          NEXT,    J/LOADSR367A
(6051) DCS(0.00.0.0.0.1) BM(0011..11.01..00.00..000..011...0.1.0..0..0...0.0000...0..1000.1...11.010...000.101.010]
9338
9339          6052:  !(FREE)
9340          LOADSR367A:
9341          P2-T,    SR+BSPhi(C052525),      !SR+(052525)
9342          NEXT,    J/GOBUT367B
(6052) DCS(0.00.0.0.0.0) BM(1010..01.11..00.00..111..000...0.0.1..0..0...0.0000...0..0000.0...11.000...000.101.011]
9343
9344          6053:  !(FREE)
9345          GOBUT367B:
9346          SETUP,   RETURN/TEST370A,      !RETURN TO START OF NEXT SUBTEST
9347          NEXT,    GOTO-PAGE(7),          !BUT'S ARE ON PAGE 7
9348          J/BUTGD3-2,      !GO BUT ON GUARD <3:2>
(6053) DCS(0.00.0.0.0.0) BM(0110..00.11..10.10..101..111...0.0.0..0..0...0.0000 ..0..0000.0...11.100...011.001.100]
9349
9350
9351
9352
9353
9354
9355          ! -----
9356
9357          !*** TEST 370A ***
9358          !TEST-370A CHECKS THAT SHIFT LEFT/GUARD-ENABLED SHIFTS GD<3>="0" INTO SR<00>
9359          6725:
9360          TEST370A:
9361          PD,      LOAD-ENUA(ZTARGET434),      !SETUP FOR IR=(000000)/INSTRS TEST
9362          LOAD-ERROR(TEST370A),      !ERROR DIRECTORY KEY
9363          DCS-CTR(C10.),      !COMPARE ENUA:TNUA AT TARGET
9364          NEXT,    J/SETRES370A
(6725) DCS(1.00.1.0.0.0) BM(0101..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...000.101.100]
9365
9366          6054:  !(FREE)
9367          SETRES370A:
9368          PD,      BUMP-VERIFY,      !COUNT
9369          P3,      CSPD(16)+EMIT,      !CSP GETS
9370          EMITC,   SENDMUX-4567-4 IL,      !RES VALUES
9371          SR-LEFT, GUARD-EN,
9372          NEXT,    J/LOADRES370A
(6054) DCS(0.00.0.0.0.1) BM(0001..10.10..00.00..000..000...0.0.0..0..0...0.0001...1..0000.0...11.000...000.101.101]
9373
9374          6055:  !(FREE)
9375          LOADRES370A:
9376          P2-T,    RES+CSP8(B16),      !STORE RES
9377          D+ASPhi(C000000), D(C)+1,      !SETUP D, D(C)
9378          NEXT,    J/SHIFT370A
(6055) DCS(0.00.0.0.0.0) BM(1111..11.01..11.01..100..000...0.1.0..0..0...0.0000...0..1000.1...11.000...000.101.110]
9379
9380          6056:  !(FREE)
9381          SHIFT370A:
9382          P2-T,    SR+SR-LEFT-1,      !SHIFT SR LEFT 1, SR<00>+GD<3>="0"

```



9383 NEXT J/COMP370A  
 (6056) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000..0.0.1..0..0...0.0000...0..0000.0...11.000...000.101.111)

9384 6057: !(FREE)  
 9385 COMP370A:  
 9386 P2-T, D+SR-XOR-BSPHI(C125252), SAVE-D(C), !D + (125252)=SR, BITWISE  
 9387 NEXT J/GOBUT370A  
 9388

(6057) DCS(0.00.0.0.0.0) BM(0110..01.11..00.00..110..111...0.1.0..0..0...0.0000...0..0000.0...11.000...000.110.000)

9389 6060: !(FREE)  
 9390 GOBUT370A:  
 9391 SETUP, RETURN/TEST370B, !RETURN TO START OF NEXT SUBTEST  
 9392 NEXT, CALL(DINTOIR-5) !SUBR: D -> IR, BUT(INSTR5)  
 9393

(6060) DCS(0.00.0.0.0.0) BM(0110..00.11..10.10..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

9394  
 9395  
 9396  
 9397  
 9398  
 9399

! - - - - -

9401  
 9402 !\*\*\* TEST 370B \*\*\*  
 9403 !TEST-370B CHECKS THAT THE GUARD, NOW ENABLED, WAS ALSO SHIFTED LEFT  
 9404 6723:  
 9405 TEST370B:

9406 PO, LOAD-ENUA(ZTARGET402), !GUARD IS NOW "10"#"10"  
 9407 LOAD-ERROR(TEST370B), !ERROR DIRECTORY KEY  
 9408 DCS-CTR(C3.), !COMPARE ENUA:TNUA AT TARGET  
 9409 NEXT J/GOBUT370B

(6723) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...000.110.001)

9410 6061: !(FREE)  
 9411 GOBUT370B:  
 9412 SETUP, RETURN/TEST370C, !RETURN TO START OF NEXT SUBTEST  
 9413 PO, BUMP-VERIFY !COUNT  
 9414 NEXT, GOTO-PAGE(7), !BUT'S ARE ON PAGE 7  
 9415 J/BUTGD3-2 !GO BUT ON GUARD (3:2)  
 9416

(6061) DCS(0.00.0.0.0.1) BM(0110..00.11..10.10..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.100)

9417  
 9418  
 9419  
 9420  
 9421  
 9422  
 9423

! - - - - -

9424  
 9425 !\*\*\* TEST 370C \*\*\*  
 9426 !TEST-370C CHECKS THAT SHIFT LEFT/GUARD-ENABLED SHIFTS GD<3>="1" INTO SR<00>  
 9427 6721:  
 9428 TEST370C:  
 9429

```

9430          PO,          LOAD-ENUA(ZTARGET434),          !SETUP FOR IR=(000000)/INSTR5 TEST
9431          LOAD-ERROR(TEST370C),          !ERROR DIRECTORY KEY
9432          DCS-CTR(C9.),          !COMPARE ENUA:TNUA AT TARGET
9433          NEXT,          J/SETDOC370C
(6721) DCS[1.00.1.0.0.0] BM[0110..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...000.110.010]

9434          6062: !(FREE)
9435          SETDOC370C:
9436          PO,          BUMP-VERIFY          !COUNT
9437          P2-T,          D+ZERO, D(C)+ALU07,          !SET D, D(C)
9438          NEXT,          J/SHIFT370C
(6062) DCS[0.00.0.0.0.1] BM[0011..00.00..00.00..000..011...0.1.0..0..0...0.0000...0..0000.0...11.000...000.110.011]

9439          6063: !(FREE)
9440          SHIFT370C:
9441          P3-T,          SR+SR-LEFT-1,          !SHIFT SR LEFT, SR<00>+GD<3>="1"
9442          NEXT,          J/COMP370C
(6063) DCS[0.00.0.0.0.0] BM[0000..00.00..00.00..000..000...1.0.1..0..0...0.0000...0..0000.0...11.000...000.110.100]

9443          6064: !(FREE)
9444          COMP370C:
9445          P2-T,          D+SR-XOR-BSPHI(C052525), SAVE-D(C),          !D + (052525)=SR, BITWISE
9446          NEXT,          J/GOBUT370C
(6064) DCS[0.00.0.0.0.0] BM[0110..01.11..00.00..111..111...0.1.0..0..0...0.0000...0..0000.0...11.000...000.110.101]

9447          6065: !(FREE)
9448          GOBUT370C:
9449          SETUP,          RETURN/TEST3700,          !RETURN TO START OF NEXT SUBTEST
9450          NEXT,          CALL(DINTOIR-5)          !SUBR: D -> IR, BUT(INSTR5)
(6065) DCS[0.00.0.0.0.0] BM[0110..00.11..10.01..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011]

9451          ! -----
9452          !
9453          !
9454          !
9455          !
9456          !
9457          !
9458          !
9459          !
9460          !
9461          !
9462          !
9463          !
9464          !*** TEST 3700 ***
9465          !TEST-3700 CHECKS THAT THE "1" THAT GOT PUT IN GD<0> CAN BE SHIFTED BACK
9466          6717:
9467          TEST3700:
9468          PO,          LOAD-ENUA(ZTARGET401),          !GUARD IS NOW "01"#"00"
9469          LOAD-ERROR(TEST3700),          !ERROR DIRECTORY KEY
9470          DCS-CTR(C3.),          !COMPARE ENUA:TNUA AT TARGET
9471          NEXT,          J/GOBUT3700
(6717) DCS[1.00.1.0.0.0] BM[1100..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...000.110.110]

9472          6066: !(FREE)
9473          GOBUT3700:
9474          SETUP,          RETURN/TEST371A,          !RETURN TO START OF NEXT SUBTEST
9475          NEXT,          GOTO-PAGE(7),          !BUT'S ARE ON PAGE 7
9476

```

```

9477          J/BUTGD3-2          !GO BUT ON GUARD (3:2)
(6066) DCS(0.00.0.0.0.0) BM(0110..00.11..10.01..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.100)
9478
9479
9480
9481
9482
9483
9484
9485 !-----
9486
9487 !*** TEST 371A ***
9488 !TEST-371A CHECKS THAT SR-NOP FUNCTION DOES NOTHING
9489 6715:
9490 TEST371A:
9491     PO,          LOAD-ENUA(ZTARGET434),          !SETUP FOR IR=(000000)/INSTRS TEST
9492     LOAD-ERROR(TEST371A),          !ERROR DIRECTORY KEY
9493     DCS-CTR(C10.),          !COMPARE ENUA:TNUA AT TARGET
9494     NEXT          J/SETRES371A
(6715) DCS(1.00.1.0.0.0) BM(0101..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...000.110.111)
9495
9496 6067: !(FREE)
9497 SETRES371A:
9498     PO,          BUMP-VERIFY,          !COUNT
9499     P3,          CSPD(16)+EMIT,          !CSP GETS
9500     EMITC,       SENDMUX-4567-SEL,          !RES VALUES
9501     SR-NOP,      GUARD-EN,
9502     NEXT          J/LOADRES371A
(6067) DCS(0.00.0.0.0.1) BM(0011..10.10..00.00..000..000...0.0.0..0..0...0.0001...1..0000.0...11.000...000.111.000)
9503
9504 6070: !(FREE)
9505 LOADRES371A:
9506     P2-T,        RES+CSPB(816),          !STORE RES
9507     D+ASPHI(C000000), D(C)+1,          !SETUP D, D(C)
9508     NEXT          J/SHIFT371A
(6070) DCS(0.00.0.0.0.0) BM(1111..11.01..11.01..100..000...0.1.0..0..0...0.0000...0..1000.1...11.000...000.111.001)
9509
9510 6071: !(FREE)
9511 SHIFT371A:
9512     PO,          BUMP-VERIFY,          !COUNT
9513     P2-T,        CLK-SR,          !DO AN SR-NOP
9514     NEXT          J/COMP371A
(6071) DCS(0.00.0.0.0.1) BM(0000..00.00..00.00..000..000...0.0.1..0..0...0.0000...0..0000.0...11.000...000.111.010)
9515
9516 6072: !(FREE)
9517 COMP371A:
9518     P2-T,        D+SR-XOR-BSPHI(C052525), SAVE-D(C),          !D + (052525)=SR, BITWISE
9519     NEXT          J/GOBUT371A
(6072) DCS(0.00.0.0.0.0) BM(0110..01.11..00.00..111..111...0.1.0..0..0...0.0000...0..0000.0...11.000...000.111.011)
9520
9521 6073: !(FREE)
9522 GOBUT371A:
9523     SETUP,       RETURN/TEST371B,          !RETURN TO START OF NEXT SUBTEST

```

```

9524          PO,      BUMP-VERIFY,      !COUNT
9525          NEXT,    CALL(DINTOIR-5)    !SUBR: D -> IR, BUT(INSTRS)
(6073) DCS(0.00.0.0.0.1) BM(0110..00.11..10.01..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

```

```

9526
9527
9528
9529
9530
9531
9532

```

```

9533 ! -----
9534
9535 !*** TEST 371B ***
9536 !TEST-371B CHECKS THAT THE GUARD WASN'T ALTERED EITHER
9537 6713:
9538 TEST371B:
9539         PO,      LOAD-ENUA(ZTARGET401), !GUARD IS STILL "01"*"00"
9540         NEXT,    LOAD-ERROR(TEST371B), !ERROR DIRECTORY KEY
9541         DCS-CTR(C3.), !COMPARE ENUA:TNUA AT TARGET
9542         J/GOBUT371B
(6713) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...000.111.100)

```

```

9543 6074: !(FREE)
9544 GOBUT371B:
9545         SETUP,  RETURN/SCOPE371,      !RETURN TO SCOPE LOOP TEST WORD
9546         NEXT,  GOTO-PAGE(7),         !BUT'S ARE ON PAGE 7
9547         J/BUTG03-2                   !GO BUT ON GUARD <3:2>
9548 (6074) DCS(0.00.0.0.0.0) BM(0110..00.00..01.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.100)

```

```

9549
9550 !SCOPE LOOP TEST FOR SR, GUARD, XMUX, RES AREA CODE
9551 6075: !(FREE)
9552 SCOPE371:
9553         PO,      BUMP-VERIFY,      !COUNT
9554         P2,      RES+CSPD(C000000), !RESET RES TO SR-LOAD/GUARD-DIS
9555         NEXT,    BUTD(SCOPE),       !NO ERROR: "TEST372A" [+3. WORDS]
9556         J/TEST372A                   ! ERROR: "SETRES361A" [-85. WORDS]
9557 (6075) DCS(0.00.0.1.0.1) BM(0000..10.00..00.00..000..000...0.0.0..0..0...0.0100...0..1000.1...11.000...101.111.001)

```

```

9558
9559
9560
9561 ! -----
9562
9563 !
9564 ! THE FOLLOWING TWO SUBROUTINES ARE ALSO USED IN THE ABOVE TESTS:
9565
9566

```

```

9567 7031: !(FREE)
9568 CSP16XORSRTOIRS:
9569         P2-T,    D+SR-XOR-CSPB(B16), SAVE-D(C), !COMPARE SR-XMUX:EXPECTED VALUE, BITWISE
9570         NEXT,    J/DINTOIRS          ! AND PUT IN IR TO DO INSTRS TEST
(7031) DCS(0.00.0.0.0.0) BM(0110..11.01..00.00..000..111...0.1.0..0..0...0.0000...0..0000.0...11.000...010.111.011)

```

```

9571
9572
9573 7035: !(FREE)
9574 CSP16XORFLTTOIRS:
9575     P2-T, D=FLTPT-XOR-CSPB(B16), SAVE-D(C), !COMPARE FLTPT-XMUX:EXPECTED VALUE, BITWISE
9576     NEXT, J/DINTOIRS ! AND PUT IN IR TO DO INSTRS TEST
(7035) DCS(0.00.0.0.0.0) BM(0110..11.01..00.01..000..111...0.1.0..0..0...0.0000...0..0000.0...11.000...010.111.011)

```

9577  
9578  
9579  
9580  
9581  
9582  
9583  
9584  
9585  
9586  
9587  
9588  
9589  
9590  
9591  
9592  
9593  
9594  
9595  
9596  
9597  
9598  
9599  
9600  
9601  
9602  
9603  
9604  
9605  
9606  
9607  
9608  
9609  
9610  
9611  
9612  
9613  
9614  
9615  
9616  
9617  
9618  
9619  
9620  
9621

!.PAGE=====

.TOC \* TEST372A-372B: TESTING CUA, PROCESSOR MUX, AND BUTA(SUBR A)

```

*****
*
* TESTS: 372A - 372B UMWORDS: 015 + 002
*
* FUNCTIONS: TESTS THAT CUA IS LOADED, AND CAN BE READ THRU PROCESSOR MUX.
* ALSO TESTS THAT CAN BE PUT INTO D, AND BUT(SUBR A) LOADS
* D<14:03> INTO RETURN.
*****

```

!-----

!\*\*\* TEST 372A \*\*\*  
!TEST-372A CHECKS THE CUA -> D -> SUBR A -> RETURN PATH WITH PATTERN [6222]

```

6571:
TEST372A:
    PO, LOAD-ENLUA(CUA372A), !WHERE WE BUT (RETURN) TO
        LOAD-ERROR(TEST372A), !ERROR DIRECTORY KEY
        DCS-CTR(C7.), !IN 7. MICROWORDS
    NEXT, J/LOOP372A
(6571) DCS(1.00.1.0.0.0) BM(1000..00.11..00.10..010..010...0.0.0..0..0...0.0000...0..0000.0...11.000...101.100.000)

```

```

6540:
LOOP372A:
    SELECT, UCON-PROC, !SELECT PROCESSOR UCON CONTROL
    ENABLE, BUSDIN+CUA(14-03), !PUT 0#CUA(11-00)#EXFLAG<2:1>#FOVP ON BUSDIN
    PO, SET-UCON-CONTROL, !LOAD UCON REGISTER AT PO
        BUMP-VERIFY, !COUNT
    P3, BUTA(CUA-TRACK), !RESET TRACKING OF CUA
    NEXT, J/SETDC372A
(6540) DCS(0.00.0.0.0.1) BM(0000..00.00..01.01..000..000...0.0.0..0..0...1.1001...0..0000.0...11.001...000.111.110)

```

6076: !(FREE)



```

(6100) DCS(0.00.0.0.0.0) BM(0101..00.01..11.11..110..101...0.0.0..0..0...0.0000...0..0000.0...11.100...000.000.000)
9669
9670 5000: !(FREE)
9671 SETDC3728:
9672 PO BUMP-VERIFY !COUNT
9673 P2-T, D+ASPHI(C000000), D(C)+1, !SET D(C) FLAG = (1) FOR FIRST LOOP
9674 P3, BUTA(CUA-TRACK), !RESET TRACKING OF CUA
9675 NEXT J/CUA3728
(5000) DCS(0.00.0.0.0.1) BM(1111..00.00..11.01..100..000...0.1.0..0..0...0.0000...0..0000.0...11.001...101.101.101)
9676
9677 5555:
9678 CUA3728:
9679 P3, CSPD(16)+BUSDIN, RETURN/BUTERRORS, !COPY CUA (WHICH IS ADDRESS OF THIS WORD) INTO CSP
9680 NEXT, BUTR(D(C)-8), !IF D(C) SET (PASS 1), GO TO "LOAD3728"
9681 J/SCOPE3728 !IF D(C) RESET (PASS 2), GO TO "SCOPE3728"
(5555) DCS(0.00.0.0.0.0) BM(0101..10.01..11.11..110..000...0.0.0..0..0...0.0001...1..0000.0...10.011...101.000.001)
9682
9683 5503:
9684 LOAD3728:
9685 P2-T, D+CSPB(B16), D(C)+0, !PUT CUA FROM CSP INTO D, RESET D(C)
9686 NEXT, J/SUBRA3728
(5503) DCS(0.00.0.0.0.0) BM(1010..11.01..00.00..000..000...0.1.0..0..0...0.0000...0..0000.0...11.000...010.010.011)
9687
9688 5223: !(FREE)
9689 SUBRA3728:
9690 PO BUMP-VERIFY !COUNT
9691 SETUP, RETURN/BUTERRORS, !IF BUTA(SUBR-B) USED INSTEAD
9692 NEXT, PAGE(7) !SUBROUTINE IS ON PAGE 7
9693 BUTA(SUBR-A), !LOAD PAGE, LOAD RETURN FROM D
9694 J/RESETUONP !THE SUBR: RESETS PROC UCON: BUSDIN+EMIT, EN-CLK-IR
(5223) DCS(0.00.0.0.0.1) BM(0101..00.01..11.11..110..111...0.0.0..0..0...0.0000...0..0000.0...11.101...010.111.001)
9695
9696 !NEXT WORD COMES FROM 'RESETUONP' WHICH DOES ONLY A BUTA(RETURN),
9697 !WHICH SHOULD RETURN TO 'CUA3728' [-3.WORDS]
9698
9699 5501:
9700 SCOPE3728:
9701 NEXT, GOTO-PAGE(6), !XFER
9702 BUTD(SCOPE), !NO ERROR: "TEST373A" [+1.WORDS]
9703 J/TEST373A ! ERROR: "LOOP372A" [-12.WORDS]
(5501) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..110...0.0.0..0..0...0.0000...0..0000.0...11.100...101.100.001)
9704
9705
9706
9707
9708
9709 !.PAGE=====
9710
9711
9712 .TOC * TEST373: CHECK JAMUPP W/ BUTA(DIAGNOSE), BM EXT BIT FLPADR
9713
9714

```

```

9715 *****
9716 !*
9717 !* TESTS: 373 A - B UWORDS: 007 + 014
9718 !*
9719 !* FUNCTIONS:
9720 !*
9721 !* THE FOLLOWING SET OF TWO TESTS PERFORMS SEVERAL FUNCTIONS:
9722 !*
9723 !* TEST-373-A CHECKS THAT CONTROL CAN BE PASSED TO THE BASE MACHINE, VIA
9724 !* BUTA(DIAGNOSE), AND SEVERAL B.M. WORDS EXECUTED. A BUTA(DIAGNOSE) IN
9725 !* THE B.M. SHOULD THEN BE ENCOUNTERED, RETURNING CONTROL TO THE DCS VIA
9726 !* A JAMUPP FORCE.
9727 !*
9728 !* TEST-373-B THEN CHECKS THAT THE B.M. MACHINE CODE CORRECTLY
9729 !* ASSERTED THE "FLPADR-L" EXTENSION BIT, FORCING A READ, VIA THE ASP/DF
9730 !* FIELD MODE, OF A SCRATCHPAD.
9731 !*
9732 *****
9733
9734
9735 !THIS TEST GOES TO THE B.M. VIA BUTA(DIAGNOSE)
9736 6541:
9737 TEST373A:
9738 PO, LOAD-ENUA(4777), !JAMUPPS GO HERE IN DCS
9739 LOAD-ERROR(TEST373A), !ERROR DIRECTORY KEY
9740 DCS-CTR(C7.), !COMPARE AT JAMUPP
9741 NEXT, J/RETURN373A
9742 (6541) DCS(1.00.1.0.0.0) BM(1000..00.10..01.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.000...101.110.010)
9743
9744 6562:
9745 RETURN373A:
9746 P3, CSPD(00)+EMIT, RETURN/TEST373B, !RETURN AFTER PROCESSING, TO NEXT TEST
9747 NEXT, GOTO-PAGE(7), !JAMUPP ROUTINE EXPECTS RETURN ADDRESS IN CSP(00)
9748 J/SETIR373A
9749 (6562) DCS(0.00.0.0.0.0) BM(0110..10.10..11.11..010..111...0.0.0..0..0...0.1111...1..0000.0...11.100...000.011.100)
9750
9751 7034: !(FREE)
9752 SETIR373A:
9753 P2-U, IR+EMIT, EMIT/000002, !SET DF=(2) IN IR
9754 P3, BUTA(DIAGNOSE), !BEGIN THE TRANSFER SEQUENCE
9755 NEXT, J/SETSR373A
9756 (7034) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..010...0.0.0..0..0...1.1010...0..0000.0...11.011...000.011.111)
9757
9758 7037: !(FREE)
9759 SETSR373A:
9760 P2-T, SR+CSPD(C052525), !SET SR(00)=(1) FOR JAMUPP EXPECTED (BY JAMUPP SERVICER)
9761 NEXT, GOTO-PAGE(3), !SET PAGE TO BE ACTUAL (3)
9762 J/MED25 !POINT UPF TO B.M. DISP. ON PAGE
9763 (7037) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..011...0.0.1..0..0...0.0111...0..0000.0...11.100...000.010.000)
9764 !THE SEQUENCE OF CONTROL SHOULD NOW BE COMING FROM THE B.M.:

```



```

9763
9764 :3020:
9765 :MED25:
9766 :NEXT, GOTO-PAGE(2), :XFER
9767 : J/MED25A :
9768
9769 :2071:
9770 :MED25A:
9771 :P2-T, D+ASPHI(DF)-TOP, SAVE-D(C), !READ FROM ASPHI/DF, WITH FLPADR ASSERTED
9772 : !SINCE DF=(2), THE SPADRS=(12) WILL BE FORCED
9773 :P3, BUTA(DIAGNOSE), !ALSO START TO RETURN XFER TO DCS, VIA JAMUPP
9774 :NEXT, J/MED19A :
9775
9776 :2066:
9777 :MED19A:
9778 :NEXT, GOTO-PAGE(3), !THIS WORD ALSO GETS EXECUTED
9779 : J/MED19 !BUT NOW JAMUPP GETS FORCED, AND WE GO NO FURTHER
9780
9781 :*** CONTROL NOW COMES BACK TO DCS JAMUPP POINT ***
9782
9783 :4777:
9784 :JAMUPP001:
9785 :P3-T, SR+D !SAVE OLD D IN SR
9786 :NEXT, BUTR(SR00), !TEST SR<00>:
9787 : J/JAMUPP003 !IF=(0), NOT-EXPECTED JAM, GOTO(JAMUPP003), (SEE JAMUPP ROUTINE)
9788 : !IF=(1), EXPECTED JAM, GOTO(JAMUPP002B)
9789
9790 :* COME HERE FOR EXPECTED JAM *
9791 :4757:
9792 :JAMUPP002B:
9793 :P2-T, D+CSPD(000), SAVE-D(C), !GET RETURN ADDRESS, STORED IN CSP(00)
9794 :NEXT, J/JAMUPP002C :
9795
9796 :=[4000:4777]
9797 :JAMUPP002C:
9798 :P0, RETURN+D(14-03), PAGE(7), !PUT RETURN ADDRESS INTO RETURN REGISTER
9799 :P2-T, D+SR, SAVE-D(C), !RESTORE OLD D
9800 :NEXT, J/JAMUPP002D :
9801
9802 :=[7000:7377]
9803 :JAMUPP002D:
9804 :P2-T, SR+ZERO !ZERO OUT SR<00>, FURTHER JAMUPPS NOW ILLEGAL
9805 :NEXT, BUTA(RETURN), !AND RETURN TO FLOWS
9806 : J/BUTERROR7 !ONLY IF ERROR
9807
9808
9809 :!AT THIS POINT, CONTROL SHOULD NOW RETURN TO THIS POINT; THE NEXT TEST
9810 :!THIS TEST NOW CHECKS TO SEE THAT THE B.M. FUNCTION WAS EXECUTED CORRECTLY
9811 :!NOTE THAT IN TEST-350 ( A WHILE BACK ), A#8SPHI(12) WERE LOADED WITH (000152) DATA
9812
9813 :6572:
9814 :TEST3738:
9815 :P0, LOAD-ENUA(ZTARGET425), !FOR INSTRS-E78-(425) DECODE
9816 : LOAD-ERROR(TEST3738), !ERROR DIRECTORY KEY

```

```

9817          DCS-CTR(C6.),          !COMPARE AT TARGET
9818      NEXT      J/G0BUT3738      !
(6572) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..010..101...0.0.0..0..0...0.0000...0..0000.0...11.000...001.000.001)
9819
9820      6101:  !(FREE)
9821      G0BUT3738:
9822          SETUP, RETURN/SCOPE373, !RETURN TO SCOPE LOOP TEST WORD
9823          NEXT,  CALL(DINTOIR-5)  !GO DO D -> DBUF -> IR, BUT(INSTRS)
(6101) DCS(0.00.0.0.0.0) BM(0110..00.00..10.00..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)
9824
9825
9826
9827      6102:  !(FREE)
9828      SCOPE373:
9829          PO,      BUMP-VERIFY,    !COUNT
9830          NEXT,    BUTD(SCOPE),    !NO ERROR: "TEST374" (+1. WORDS)
9831          J/TEST374                !      ERROR: "RETURN373A" (-12. WORDS)
(6102) DCS(0.00.0.1.0.1) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...101.110.011)

```

!.PAGE=====

.TOC \* TEST374: A/B SP REWRITE MODES VERIFICATION

\*\*\*\*\*

TESTS: 374 A - F UWORDS: 076 + 336

FUNCTIONS:

THE FOLLOWING GROUP OF TWELVE TESTS PERFORMS A VERIFICATION OF THE A/B SP REWRITE FUNCTIONS:

WR(A,L,A)	WR(A,L,B)	WR(A,H,A)	WR(A,H,B)
WR(B,L,A)	WR(B,L,B)	WR(B,H,A)	WR(B,H,B)
WR(AB,L,A)	WR(AB,L,B)	WR(AB,H,A)	WR(AB,H,B)

EACH FUNCTION IS INVOKED, AND THE RESULTANT SCRATCHPADS ARE CHECKED TO INSURE THAT (1) ONLY THE RIGHT SCRATCHPADS WERE WRITTEN, AND (2) THE CORRECT ADDRESS WAS USED FOR THE REWRITE.

\*\*\*\*\*

SUMMARY OF ASP/BSP HI/LO REWRITE FUNCTIONALITY TESTS:

			--DF---	--SF---	
TEST REWRITE--FUNCTION	ASP	BSP	ASP	BSP	SR IN

NUMB	A/B	HI/LO	A/B-ADDR	ADDR	ADDR	H-L	H-L	H-L	H-L	OCTAL
9867	A1	LO	A-ADDR	SF/4	DF/2	0	0	0	0	032100
9868	A2	LO	B-ADDR			0	1	0	0	
9870	B1	HI	A-ADDR	DF/2	SF/4	1	0	0	0	100010
9871	B2	HI	B-ADDR			0	0	0	0	
9872							1	0	0	
9873	C1	LO	A-ADDR	DF/4	SF/2	0	0	0	1	010001
9874	C2	LO	B-ADDR			0	0	0	0	
9875									1	
9876	D1	HI	A-ADDR	SF/2	DF/4	0	0	0	0	001040
9877	D2	HI	B-ADDR			0	0	1	0	
9878									0	
9879	E1	LO	A-ADDR	SF/4	DF/2	0	0	0	0	002520
9880	E2	LO	B-ADDR			0	1	0	1	
9881									0	
9882	F1	HI	A-ADDR	DF/4	SF/2	1	0	1	0	120012
9883	F2	HI	B-ADDR			0	0	0	0	
9884							1	0	1	
9885									0	
9886									1	
9887									0	
9888									1	
9889									0	
9890									1	
9891									0	
9892									1	
9893									0	
9894									1	

!\*\*\* TEST 374A \*\*\*

6563:

TEST374:

```

P0,      BUMP-VERIFY,          !COUNT
          LOAD-ERROR(TEST374), !ERROR DIRECTORY KEY
P3,      CSPD(01)+EMIT,        !CONSTANT FOR:
EMITC,   SR-LEFT, GUARD-DIS,  !SETUP RES FOR SR FUNCTION
          EMIT07/1,           !FLAG BIT FOR TESTING
NEXT,    J/LOADRES374

```

(6563) DCS(1.00.0.0.0.1) BM(0001..10.00..00.10..000..000...0.0.0..0..0...0.1110...1..0000.0...11.000...001.000.011)

6103: !(FREE)

LOADRES374:

```

P0,      DCS-CTR(C15.),       !HOLD UP ERROR COUNTER
P2,      RES+CSPD(D01),       !SR NOW SETUP
NEXT,    GOTO-PAGE(5),        !XFR
          J/EXPEC374A1

```

(6103) DCS(0.00.1.0.0.0) BM(0000..10.00..00.00..000..101...0.0.0..0..0...0.1110...0..1000.1...11.100...010.010.010)

5222: !(FREE)

EXPEC374A1:

```

P3,      CSPD(02)+EMIT, EMIT/002100, !EXPECTED "SERIAL" REPRESENTATION OF RESULT
NEXT,    J/ZER0374A1

```

(5222) DCS(0.00.0.0.0.0) BM(0000..10.01..00.01..000..000...0.0.0..0..0...0.1101...1..0000.0...11.000...101.011.110)

```

9917
9918 5536:
9919 ZERO374A1:
9920     PO,      BUMP-VERIFY,          !COUNT
9921     SETUP,   RETURN/DOWRITE374A1, !EXEC SUBR WHICH:
9922     NEXT,    CALL(ZEROSFO4DF02)    !(1) (000402) -> IR,
9923     (5536) DCS(0.00.0.0.0.1) BM(0101..00.01..00.10..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.100.001)
          !(2) WRITES ZEROES TO A/B SP HI/LO SF/DF

9924
9925 5225: !(FREE)
9926 DOWRITE374A1:
9927     P2-T,    D+CSPD(001) D(C)+0,    !DATA WITH BIT<07> SET
9928     ASP-ADDRS-R(SF), !ADDRESS ASP WITH SF MODE
9929     BSP-ADDRS-R(DF), !ADDRESS BSP WITH DF MODE
9930     P3,      WR(A,LO,A-ADDR),      !SELECT THE PARTICULAR FUNCTION TO TEST,
9931     NEXT,    J/GETTEM374A1         ! USING A-ADDR FOR REWRITE
9932     (5225) DCS(0.00.0.0.0.0) BM(1010..10.00..00.11..000..000...0.1.0..0..0...0.1110...0..0001.0...11.000...010.010.110)

9933
9934 5226: !(FREE)
9935 GETTEM374A1:
9936     P3,      CSPD(00)+EMIT, RETURN/ZERO374A2, ! (SEE DESCRIPT OF SUBR FOR FUNCTION)
9937     NEXT,    CALL(SFDFTOSR)
9938     (5226) DCS(0.00.0.0.0.0) BM(0101..10.01..00.10..111..101...0.0.0..0..0...0.1111...1..0000.0...11.100...111.100.110)

9939
9940 5227: !(FREE)
9941 ZERO374A2:
9942     SETUP,   RETURN/DOWRITE374A2, !AGAIN GO WRITE ZEROES TO A/B-SP-HI/LO
9943     NEXT,    CALL(ZEROSFO4DF02)
9944     (5227) DCS(0.00.0.0.0.0) BM(0101..00.01..00.11..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.100.001)

9945
9946 5230: !(FREE)
9947 DOWRITE374A2:
9948     P2-T,    D+CSPD(001) D(C)+0,    !DATA WITH BIT<07> SET
9949     ASP-ADDRS-R(SF), !ADDRESS ASP WITH SF MODE
9950     BSP-ADDRS-R(DF), !ADDRESS BSP WITH DF MODE
9951     P3,      WR(A,LO,B-ADDR),      !USE SAME FUNCTION AS ABOVE,
9952     NEXT,    J/GETTEM374A2         ! ONLY USE B-ADDR FOR REWRITE THIS TIME
9953     (5230) DCS(0.00.0.0.0.0) BM(1010..10.00..00.11..000..000...0.1.0..0..0...0.1110...0..0101.0...11.000...010.011.001)

9954
9955 5231: !(FREE)
9956 GETTEM374A2:
9957     P3,      CSPD(00)+EMIT, RETURN/TEST374A2, ! (SEE DESCRIPT OF SUBR FOR FUNCTION)
9958     NEXT,    CALL(SFDFTOSR)
9959     (5231) DCS(0.00.0.0.0.0) BM(0101..10.11..00.00..100..101...0.0.0..0..0...0.1111...1..0000.0...11.100...111.100.110)

9960
9961 5604:
9962 TEST374A2:
          PO,      LOAD-ENVA(ZTARGET434), !NOW SETUP FOR IR=ZERO COMPARE
          LOAD-ERROR(TEST374A2), !ERROR DIRECTORY KEY
          DCS-CTR(C6.), !COMPARE AT TARGET

```

```

9963          BUMP-VERIFY          !COUNT
9964          NEXT, J/GOTEST374A2  !
(5604) DCS(1.00.1.0.0.1) BM(1001..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...010.011.010)
9965          5232: !(FREE)
9966          GOTEST374A2:
9967          SETUP, RETURN/SCOPE374A, !GO EXEC SUBR THAT:
9968          NEXT, CALL(DINTOIR-5) ! PUTS D -> IR, BUT(INSTR5) TO TEST FOR ZERO
9969          (5232) DCS(0.00.0.0.0.0) BM(0101..00.01..00.11..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)
9970
9971
9972
9973          5233: !(FREE)
9974          SCOPE374A:
9975          NEXT, BUTD(SCOPE) !NO ERROR: "EXPEC374B1" (+1. WORDS)
9976          J/EXPEC374B1 ! ERROR: "ZER0374A1" (-8. WORDS)
9977          (5233) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...101.011.111)
9978
9979
9980
9981          ! - - - - -
9982
9983          !*** TEST 374B ***
9984
9985          5537:
9986          EXPEC374B1:
9987          P3, CSPD(02)+EMIT, EMIT/1000!0, !EXPECTED "SERIAL" REPRESENTATION OF RESULT
9988          NEXT, J/ZER0374B1 !
9989          (5537) DCS(0.00.0.0.0.0) BM(1000..10.00..00.00..001..000...0.0.0..0..0...0.1101...1..0000.0...11.000...101.101.110)
9990
9991          5556:
9992          ZER0374B1:
9993          PD, BUMP-VERIFY, !COUNT
9994          SETUP, RETURN/DOWRITE374B1, !EXEC SUBR WHICH:
9995          NEXT, CALL(ZEROSF04DF02) ! (1) (000402) -> IR,
9996          (5556) DCS(0.00.0.0.0.1) BM(0101..00.01..00.11..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.100.001)
9997
9998          5234: !(FREE)
9999          DOWRITE374B1:
10000          P2-T, D+CSPD(001), D(C)+0, !DATA WITH BIT<07> SET
10001          ASP-ADDRS-R(DF), !ADDRESS ASP WITH DF MODE
10002          BSP-ADDRS-R(SF), !ADDRESS BSP WITH SF MODE
10003          P3, WR(A,HI,A-ADDR), !SELECT THE PARTICULAR FUNCTION TO TEST,
10004          NEXT, J/GETTEM374B1 ! USING A-ADDR FOR REWRITE
(5234) DCS(0.00.0.0.0.0) BM(1010..10.01..00.10..000..000...0.1.0..0..0...0.1110...0..1001.0...11.000...010.011.101)
10005
10006          5235: !(FREE)
10007          GETTEM374B1:
10008          P3, CSPD(00)+EMIT, RETURN/ZER0374B2, !(SEE DESCIP OF SUBR FOR FUNCTION)

```

```

10009      NEXT, CALL(SFDFTOSR)
(5235) DCS(0.00.0.0.0.0) BM(0101..10.01..00.11..110..101...0.0.0..0..0...0.1111...1..0000.0...11.100...111.100.110)
10010
10011
10012
10013      5236: !(FREE)
10014      ZERO37482:
10015      SETUP, RETURN/DOWRITE37482, !AGAIN GO WRITE ZEROES TO A/B-SP-HI/LO
10016      NEXT, CALL(ZEROSFO4DF02)
(5236) DCS(0.00.0.0.0.0) BM(0101..00.01..00.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.100.001)
10017
10018      5237: !(FREE)
10019      DOWRITE37482:
10020      P2-T, D+CSPD(001), D(C)+0, !DATA WITH BIT<07> SET
10021      ASP-ADDRS-R[DF], !ADDRESS ASP WITH DF MODE
10022      BSP-ADDRS-R[SF], !ADDRESS BSP WITH SF MODE
10023      P3, WR(A HI, B-ADDR), !USE SAME FUNCTION AS ABOVE
10024      NEXT, J/GETTEM37482 ! ONLY USE B-ADDR FOR REWRITE THIS TIME
(5237) DCS(0.00.0.0.0.0) BM(1010..10.01..00.10..000..000...0.1.0..0..0...0.1110...0..1101.0...11.000...010.100.000)
10025
10026      5240: !(FREE)
10027      GETTEM37482:
10028      P3, CSPD(00)+EMIT, RETURN/TEST37482, !(SEE DESCRIP OF SUBR FOR FUNCTION)
10029      NEXT, CALL(SFDFTOSR)
(5240) DCS(0.00.0.0.0.0) BM(0101..10.11..00.00..110..101...0.0.0..0..0...0.1111...1..0000.0...11.100...111.100.110)
10030
10031      5606:
10032      TESTJ37482:
10033      PO, LOAD-ENLUA(ZTARGET434), !NOW SETUP FOR IR=ZERO COMPARE
10034      LOAD-ERROR(TEST37482), !ERROR DIRECTORY KEY
10035      DCS-CTR(C6.), !COMPARE AT TARGET
10036      BUMP-VERIFY, !COUNT
10037      NEXT, J/GOTEST37482
(5606) DCS(1.00.1.0.0.1) BM(1001..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...010.100.001)
10038
10039      5241: !(FREE)
10040      GOTEST37482:
10041      SETUP, RETURN/SCOPE3748, !GO EXEC SUBR THAT:
10042      NEXT, CALL(DINTOIR-5) ! PUTS D -> IR, BUT(INSTR5) TO TEST FOR ZERO
(5241) DCS(0.00.0.0.0.0) BM(0101..00.01..01.00..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)
10043
10044
10045
10046
10047      5242: !(FREE)
10048      SCOPE3748:
10049      NEXT, BUTD(SCOPE1, !NO ERROR: "EXPEC374C1" (+1. WORDS)
10050      J/EXPEC374C1 ! ERROR: "ZERO374B1" (-8. WORDS)
(5242) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...101.101.111)
10051
10052
10053

```

```

10054
10055 ! - - - - -
10056
10057 !*** TEST 374C ***
10058
10059 5557:
10060 EXPECT374C1:
10061     P3,      CSPD(02)+EMIT, EMIT/010001,      ! EXPECTED "SERIAL" REPRESENTATION OF RESULT
10062     NEXT,    J/ZER0374C1                      !
(5557) DCS(0.00.0.0.0.0) BM(0001..10.00..00.00..000..001...0.0.0..0..0...0.1101...1..0000.0...11.000...101.110.110)

10063
10064 5566:
10065 ZER0374C1:
10066     P0,      BUMP-VERIFY                      ! COUNT
10067     SETUP,   RETURN/DOWRITE374C1,           ! EXEC SUBR WHICH:
10068     NEXT,    CALL(ZER0SF02DF04)             ! (1) (000204) -> IR,
10069     ! (2) WRITES ZEROES TO A/B SP HI/LO SF/DF
(5566) DCS(0.00.0.0.0.1) BM(0101..00.01..01.00..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.011.110)

10070
10071 5243: !(FREE)
10072 DOWRITE374C1:
10073     P2-T,    D+CSPD(001), D(C)+0,           ! DATA WITH BIT<07> SET
10074     ASP-ADDS-R(DF),                          ! ADDRESS ASP WITH DF MODE
10075     BSP-ADDS-R(SF),                          ! ADDRESS BSP WITH SF MODE
10076     P3,      WR(B,LO,A-ADDR),              ! SELECT THE PARTICULAR FUNCTION TO TEST,
10077     NEXT,    J/GETTEM374C1                  ! USING A-ADDR FOR REWRITE
(5243) DCS(0.00.0.0.0.0) BM(1010..10.01..00.10..000..000...0.1.0..0..0...0.1110...0..0010.0...11.000...010.100.100)

10078
10079 5244: !(FREE)
10080 GETTEM374C1:
10081     P3,      CSPD(00)+EMIT, RETURN/ZER0374C2, ! (SEE DESCRIPTOR OF SUBR FOR FUNCTION)
10082     NEXT,    CALL(SFDFTSR)
(5244) DCS(0.00.0.0.0.0) BM(0101..10.01..01.00..110..101...0.0.0..0..0...0.1111...1..0000.0...11.100...111.100.110)

10083
10084
10085 5246: !(FREE)
10086 ZER0374C2:
10087     SETUP,   RETURN/DOWRITE374C2,           ! AGAIN GO WRITE ZEROES TO A/B-SP-HI/LO
10088     NEXT,    CALL(ZER0SF02DF04)
(5246) DCS(0.00.0.0.0.0) BM(0101..00.01..01.00..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.011.110)

10090
10091 5247: !(FREE)
10092 DOWRITE374C2:
10093     P2-T,    D+CSPD(001), D(C)+0,           ! DATA WITH BIT<07> SET
10094     ASP-ADDS-R(DF),                          ! ADDRESS ASP WITH DF MODE
10095     BSP-ADDS-R(SF),                          ! ADDRESS BSP WITH SF MODE
10096     P3,      WR(B,LO,B-ADDR),              ! USE SAME FUNCTION AS ABOVE
10097     NEXT,    J/GETTEM374C2                  ! ONLY USE B-ADDR FOR REWRITE THIS TIME
(5247) DCS(0.00.0.0.0.0) BM(1010..10.01..00.10..000..000...0.1.0..0..0...0.1110...0..0110.0...11.000...010.101.000)

10098
10099 5250: !(FREE)

```

```

10100 GETTEM374C2:
10101     P3,      CSPD[00]←EMIT, RETURN/TEST374C2,      ;(SEE DESCRIP OF SUBR FOR FUNCTION)
10102     NEXT,   CALL[SFDFTOSR]                          ;
(5250) DCS[0.00.0.0.0.0] BM[0101..10.10..10.10..100..101...0.0.0..0..0...0.1111...1..0000.0...11.100...111.100.110]

10103
10104     5524:
10105     TEST374C2:
10106     PO,      LOAD-ENUA(ZTARGET434),                ;NOW SETUP FOR IR=ZERO COMPARE
10107     LOAD-ERROR(TEST374C2),                          ;ERROR DIRECTORY KEY
10108     DCS-CTR(C6.),                                    ;COMPARE AT TARGET
10109     BUMP-VERIFY,                                     ;COUNT
10110     NEXT,   J/GOTEST374C2
(5524) DCS[1.00.1.0.0.1] BM[1001..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...010.101.001]

10111
10112     5251: !(FREE)
10113     GOTEST374C2:
10114     SETUP,  RETURN/SCOPE374C,                        ;GO EXEC SUBR THAT:
10115     NEXT,   CALL[DINTOIR-5]                          ; PUTS 0 -> IR, BUT(INSTR5) TO TEST FOR ZERO
(5251) DCS[0.00.0.0.0.0] BM[0101..00.01..01.01..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011]

10116
10117
10118
10119
10120     5252: !(FREE)
10121     SCOPE374C:
10122     NEXT,   BITD[SCOPE],                             ;NO ERROR: "EXPEC374D1" (+1. WORDS)
10123     J/EXPEC374D1                                     ; ERROR: "ZER0374C1" (-8. WORDS)
(5252) DCS[0.00.0.1.0.0] BM[0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...101.110.111]

10124
10125
10126
10127
10128     ! - - - - -
10129
10130     !*** TEST 374D ***
10131
10132     5567:
10133     EXPEC374D1:
10134     P3,      CSPD[02]←EMIT, EMIT/001040,             ;EXPECTED "SERIAL" REPRESENTATION OF RESULT
10135     NEXT,   J/ZER0374D1
(5567) DCS[0.00.0.0.0.0] BM[0000..10.00..10.00..100..000...0.0.0..0..0...0.1101...1..0000.0...11.000...101.001.100]

10136
10137     5514:
10138     ZER0374D1:
10139     PO,      BUMP-VERIFY,                             ;COUNT
10140     SETUP,  RETURN/DOWRITE374D1,                    ;EXEC SUBR WHICH:
10141     NEXT,   CALL[ZER0SF02DF04]                       ;(1) (000204) -> IR,
10142     ;(2) WRITES ZEROES TO A/B SP HI/LO SF/DF
(5514) DCS[0.00.0.0.0.1] BM[0101..00.01..01.01..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.011.110]

10143
10144     5253: !(FREE)
10145     DOWRITE374D1:

```



```

10146      P2-T,      D=CSPD(D01), D(C)+0,      !DATA WITH BIT<07> SET
10147      ASP-ADDRS-R(SF),      !ADDRESS ASP WITH SF MODE
10148      BSP-ADDRS-R(DF),      !ADDRESS BSP WITH DF MODE
10149      P3,      WR(B HI, A-ADDR),      !SELECT THE PARTICULAR FUNCTION TO TEST,
10150      NEXT,      J/GETTEM374D1      ! USING A-ADDR FOR REWRITE
(5253) DCS(0.00.0.0.0.0) BM(1010..10.00..00.11..000..000...0.1.0..0..0...0.1110...0..1010.0...11.000...010.101.100)

10151      5254:      !(FREE)
10152      GETTEM374D1:
10153      P3,      CSPD(00)+EMIT, RETURN/ZER0374D2,      !(SEE DESCRIIP OF SUBR FOR FUNCTION)
10154      NEXT,      CALL(SFDFTSR)
10155      (5254) DCS(0.00.0.0.0.0) BM(0101..10.01..01.01..101..101...0.0.0..0..0...0.1111...1..0000.0...11.100...111.100.110)

10156
10157
10158      5255:      !(FREE)
10159      ZER0374D2:
10160      SETUP,      RETURN/DOWRITE374D2,      !AGAIN GO WRITE ZEROES TO A/B-SP-HI/LO
10161      NEXT,      CALL(ZER0SF02DF04)
10162      (5255) DCS(0.00.0.0.0.0) BM(0101..00.01..01.01..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.0.1.110)

10163      5256:      !(FREE)
10164      DOWRITE374D2:
10165      P2-T,      D=CSPD(D01), D(C)+0,      !DATA WITH BIT<07> SET
10166      ASP-ADDRS-R(SF),      !ADDRESS ASP WITH SF MODE
10167      BSP-ADDRS-R(DF),      !ADDRESS BSP WITH DF MODE
10168      P3,      WR(B HI, B-ADDR),      !USE SAME FUNCTION AS ABOVE,
10169      NEXT,      J/GETTEM374D2      ! ONLY USE B-ADDR FOR REWRITE THIS TIME
(5256) DCS(0.00.0.0.0.0) BM(1010..10.00..00.11..000..000...0.1.0..0..0...0.1110...0..1110.0...11.000...010.101.111)

10171      5257:      !(FREE)
10172      GETTEM374D2:
10173      P3,      CSPD(00)+EMIT, RETURN/TEST374D2,      !(SEE DESCRIIP OF SUBR FOR FUNCTION)
10174      NEXT,      CALL(SFDFTSR)
10175      (5257) DCS(0.00.0.0.0.0) BM(0101..10.10..10.00..000..101...0.0.0..0..0...0.1111...1..0000.0...11.100...111.100.110)

10176      5500:
10177      TEST374D2:
10178      PO,      LOAD-ENUA(ZTARGET434),      !NOW SETUP FOR IR=ZERO COMPARE
10179      LOAD-ERROR(TEST374D2),      !ERROR DIRECTORY KEY
10180      DCS-CTR(C6.),      !COMPARE AT TARGET
10181      BUMP-VERIFY,      !COUNT
10182      NEXT,      J/GOTEST374D2
(5500) DCS(1.00.1.0.0.1) BM(1001..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...010.110.000)

10184      5260:      !(FREE)
10185      GOTEST374D2:
10186      SETUP,      RETURN/SCOPE374D,      !GO EXEC SUBR THAT:
10187      NEXT,      CALL(DINTOIR-5)      ! PUTS D -> IR, BUT(INSTR5) TO TEST FOR ZERO
(5260) DCS(0.00.0.0.0.0) BM(0101..00.01..01.10..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

10189
10190

```

```

10191
10192
10193 5261: !(FREE)
10194 SCOPE3740:
10195     NEXT, BUTD(SCOPE) !NO ERROR: "EXPEC374E1" (+1. WORDS)
10196     J/EXPEC374E1 ! ERROR: "ZER037401" (-8. WORDS)
(5261) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...101.001.101)

10197
10198
10199
10200
10201 ! - - - - -
10202
10203 !*** TEST 374E ***
10204
10205 5515:
10206 EXPEC374E1:
10207     P3, CSPD(02)+EMIT, EMIT/002520, !EXPECTED "SERIAL" REPRESENTATION OF RESULT
10208     NEXT, J/ZER0374E1
(5515) DCS(0.00.0.0.0.0) BM(0000..10.01..01.01..010..000...0.0.0..0..0...0.1101...1..0000.0...11.000...101.010.110)

10209
10210 5526:
10211 ZER0374E1:
10212     PD, BUMP-VERIFY, !COUNT
10213     SETUP, RETURN/DOWRITE374E1, !EXEC SUBR WHICH:
10214     ! (1) (000402) -> IR,
10215     NEXT, CALL(ZEROSF04DF02) ! (2) WRITES ZEROES TO A/B SP HI/LO SF/DF
(5526) DCS(0.00.0.0.0.1) BM(0101..00.01..01.10..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.100.001)

10216
10217 5262: !(FREE)
10218 DOWRITE374E1:
10219     P2-T, D+CSPD(001), D(C)+0, !DATA WITH BIT<07> SET
10220     ASP-ADDRS-R(SF), !ADDRESS ASP WITH SF MODE
10221     BSP-ADDRS-R(DF), !ADDRESS BSP WITH DF MODE
10222     P3, WR(AB,LO,A-ADDR), !SELECT THE PARTICULAR FUNCTION TO TEST,
10223     NEXT, J/GETTEM374E1 ! USING A-ADDR FOR REWRITE
(5262) DCS(0.00.0.0.0.0) BM(1010..10.00..00.11..000..000...0.1.0..0..0...0.1110...0..0011.0...11.000...010.110.011)

10224
10225 5263: !(FREE)
10226 GETTEM374E1:
10227     P3, CSPD(00)+EMIT, RETURN/ZER0374E2, !(SEE DESCRIP OF SUBR FOR FUNCTION)
10228     NEXT, CALL(SFDF10SR)
(5263) DCS(0.00.0.0.0.0) BM(0101..10.01..01.10..100..101...0.0.0..0..0...0.1111...1..0000.0...11.100...111.100.110)

10229
10230
10231
10232 5264: !(FREE)
10233 ZER0374E2:
10234     SETUP, RETURN/DOWRITE374E2, !AGAIN GO WRITE ZEROES TO A/B-SP-HI/LO
10235     NEXT, CALL(ZEROSF04DF02)
(5264) DCS(0.00.0.0.0.0) BM(0101..00.01..01.10..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.100.001)

10236

```

```

10237 5265: !(FREE)
10238 DOWRITE374E2:
10239 P2-T, D=CSPD(001), D(C)+0, !DATA WITH BIT<07> SET
10240 ASP-ADDRS-R[SF], !ADDRESS ASP WITH SF MODE
10241 BSP-ADDRS-R[DF], !ADDRESS BSP WITH DF MODE
10242 P3, WR(AB,LO,B-ADDR), !USE SAME FUNCTION AS ABOVE
10243 NEXT, J/GETTEM374E2 ! ONLY USE B-ADDR FOR REWRITE THIS TIME
(5265) DCS[0.00.0.0.0.0] BM[1010..10.00..00.11..000..000...0.1.0..0..0...0.1110...0..0111.0...11.000...010.110.110]

```

```

10244 5266: !(FREE)
10245 GETTEM374E2:
10246 P3, CSPD[00]+EMIT, RETURN/TEST374E2, !(SEE DESCRIPT OF SUBR FOR FUNCTION)
10247 NEXT, CALL[SFDFIOSR]
10248 (5266) DCS[0.00.0.0.0.0] BM[0101..10.10..10.00..010..101...0.0.0..0..0...0.1111...1..0000.0...11.100...111.100.110]

```

```

10249 5502:
10250 TEST374E2:
10251 PO, LOAD-ENUA(ZTARGET434), !NOW SETUP FOR IR=ZERO COMPARE
10252 LOAD-ERROR(TEST374E2), !ERROR DIRECTORY KEY
10253 DCS-CTR(C6.), !COMPARE AT TARGET
10254 BUMP-VERIFY, !COUNT
10255 NEXT, J/GOTEST374E2
10256 (5502) DCS[1.00.1.0.0.1] BM[1001..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...010.110.111]

```

```

10257 5267: !(FREE)
10258 GOTEST374E2:
10259 SETUP, RETURN/SCOPE374E, !GO EXEC SUBR THAT:
10260 NEXT, CALL[DINTOIR-5] ! PUTS D -> IR, BUT(INSTR5) TO TEST FOR ZERO
10261 (5267) DCS[0.00.0.0.0.0] BM[0101..00.01..01.11..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011]

```

```

10262
10263
10264
10265
10266 5270: !(FREE)
10267 SCOPE374E:
10268 NEXT, BUTD[SCOPE], !NO ERROR: "EXPEC374F1" (+1. WORDS)
10269 J/EXPEC374F1 ! ERROR: "ZER0374E1" (-8. WORDS)
(5270) DCS[0.00.0.1.0.0] BM[0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...101.010.111]

```

```

10270
10271
10272
10273
10274 ! - - - - -
10275
10276 !*** TEST 374F ***
10277

```

```

10278 5527:
10279 EXPEC374F1:
10280 P3, CSPD[02]+EMIT, EMIT/120012, !EXPECTED "SERIAL" REPRESENTATION OF RESULT
10281 NEXT, J/ZER0374F1
(5527) DCS[0.00.0.0.0.0] BM[1010..10.00..00.00..001..010...0.0.0..0..0...0.1101...1..0000.0...11.000...101.001.110]
10282 5516:
10283

```

```

10284 ZERO374F1:
10285     PO,      BUMP-VERIFY,          !COUNT
10286     SETUP,  RETURN/DOWRITE374F1,  !EXEC SUBR WHICH:
10287     NEXT,    CALL(ZEROSF02DF04)     ! (1) (000204) -> IR,
10288 (5516) DCS(0.00.0.0.0.1) BM(0101..00.01..01.11..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.011.110)
! (2) WRITES ZEROES TO A/B SP HI/LO SF/DF

10289
10290     5271:    !(FREE)
10291     DOWRITE374F1:
10292     P2-T,    D+CSPD(D01), D(C)+0,    !DATA WITH BIT<07> SET
10293     ASP-ADDRS-R(DF),    !ADDRESS ASP WITH DF MODE
10294     BSP-ADDRS-R(SF),    !ADDRESS BSP WITH SF MODE
10295     P3,      WR(AB HI, A-ADDR),      !SELECT THE PARTICULAR FUNCTION TO TEST,
10296     NEXT,    J/GETTEM374F1          ! USING A-ADDR FOR REWRITE
(5271) DCS(0.00.0.0.0.0) BM(1010..10.01..00.10..000..000...0.1.0..0..0...0.1110...0..1011.0...11.000...010.111.010)

10297
10298     5272:    !(FREE)
10299     GETTEM374F1:
10300     P3,      CSPD(00)+EMIT, RETURN/ZER0374F2,    !(SEE DESCIP OF SUBR FOR FUNCTION)
10301     NEXT,    CALL(SFDF10SR)
(5272) DCS(0.00.0.0.0.0) BM(0101..10.01..01.11..011..101...0.0.0..0..0...0.1111...1..0000.0...11.100...111.100.110)

10302
10303
10304
10305     5273:    !(FREE)
10306     ZERO374F2:
10307     SETUP,  RETURN/DOWRITE374F2,    !AGAIN GO WRITE ZEROES TO A/B-SP-HI/LO
10308     NEXT,    CALL(ZEROSF02DF04)
(5273) DCS(0.00.0.0.0.0) BM(0101..00.01..01.11..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.011.110)

10309
10310     5274:    !(FREE)
10311     DOWRITE374F2:
10312     P2-T,    D+CSPD(D01), D(C)+0,    !DATA WITH BIT<07> SET
10313     ASP-ADDRS-R(DF),    !ADDRESS ASP WITH DF MODE
10314     BSP-ADDRS-R(SF),    !ADDRESS BSP WITH SF MODE
10315     P3,      WR(AB HI, B-ADDR),      !USE SAME FUNCTION AS ABOVE,
10316     NEXT,    J/GETTEM374F2          ! ONLY USE B-ADDR FOR REWRITE THIS TIME
(5274) DCS(0.00.0.0.0.0) BM(1010..10.01..00.10..000..000...0.1.0..0..0...0.1110...0..1111.0...11.000...010.111.101)

10317
10318     5275:    !(FREE)
10319     GETTEM374F2:
10320     P3,      CSPD(00)+EMIT, RETURN/TEST374F2,    !(SEE DESCIP OF SUBR FOR FUNCTION)
10321     NEXT,    CALL(SFDF10SR)
(5275) DCS(0.00.0.0.0.0) BM(0101..10.10..10.00..100..101...0.0.0..0..0...0.1111...1..0000.0...11.100...111.100.110)

10322
10323     5504:
10324     TEST374F2:
10325     PO,      LOAD-ENUA(ZTARGET434),    !NOW SETUP FOR IR=ZERO COMPARE
10326     LOAD-ERROR(TEST374F2),    !ERROR DIRECTORY KEY
10327     DCS-CTR(C6.),    !COMPARE AT TARGET
10328     BUMP-VERIFY,    !COUNT
10329     NEXT,    J/GOTEST374F2

```

```

10330 (5504) DCS(1.00.1.0.0.1) BM(1001..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...010.111.110)
10331 5276: !(FREE)
10332 GOTEST374F2:
10333     SETUP, RETURN(SCOPE374F,           !GO EXEC SUBR THAT:
10334     NEXT,  CALL(DINTOIR-5)           !POTS 0 -> IR, BUT(INSTR5) TO TEST FOR ZERO
(5276) DCS(0.00.0.0.0.0) BM(0101..00.01..01.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

```

```

10335
10336
10337
10338
10339 5277: !(FREE)
10340 SCOPE374F:
10341     P2, RES+CSPO(C000000),           !RESET RES TO SR-L 700/GUARD-DISABLED
10342     NEXT, BOTO(SCOPE),               !NO ERROR: "TEST3.5A" (+17. WORDS)
10343     J/TEST375A                       !ERROR: "ZER0374F1" (-8. WORDS)
(5277) DCS(0.00.0.1.0.0) BM(0000..10.00..00.00..000..000...0.0.0..0..0...0.0100...0..1000.1...11.000...101.001.111)

```

```

10344
10345
10346
10347
10348
10349

```

THE FOLLOWING SUBROUTINES ARE USED IN THE ABOVE TESTS (374 A1-F2):

THIS FIRST SUBROUTINE LOADS THE IR WITH THE APPROPRIATELY SELECTED SF AND DF FIELD VALUES, THEN PROCEEDS TO ZERO OUT THE ASP AND BSP LOCATIONS CORRESPONDING TO THESE VALUES, IN BOTH ASP/BSP HI/LO SP'S.

NOTE: THIS SUBR MUST BE ENTERED WITH BUSDIN+EMIT AND EN CLK IR PROC UCONS ENABLED

ALSO: THE WRITE FUNCTION USED IS: WR(AB,H,A) OR WR(AB,L,A) WITH A-ADDR = SF OR DF, RESPECTIVELY

```

10363
10364 7036: !(FREE)
10365 ZEROSF02DF04:
10366     P2-U, IR+EMIT, EMIT/000204,       !SF=2, DF=4
10367     NEXT, J/ZEROSFDF                 !GO ZERO
(7036) DCS(0.00.0.0.0.0) BM(0000..00.00..00.10..000..100...0.0.0..0..0...1.1010...0..0000.0...11.000...000.100.010)

```

```

10368
10369 7041: !(FREE)
10370 ZEROSF04DF02:
10371     P2-U, IR+EMIT, EMIT/000402,       !SF=4, DF=2
10372     NEXT, J/ZEROSFDF                 !GO ZERO
(7041) DCS(0.00.0.0.0.0) BM(0000..00.00..01.00..000..010...0.0.0..0..0...1.1010...0..0000.0...11.000...000.100.010)

```

```

10373
10374 7042: !(FREE)
10375 ZEROSFDF:
10376     P2-T, D+ZERO, SAVE-D(C),         !ZER0ES
10377     P3,  A#BSPHI(DF)+D,             !

```

10378 NEXT, J/ZEROSFDFA  
 (7042) DCS(0.00.0.0.0.0) BM(0011..00.00..00.10..000..111...0.1.0..0..0...0.0000...0..1011.0...11.000...000.100.011)

10379 7043: !(FREE)  
 10380 ZEROSFDFA:

10381 P3, A#BSPLO(DF)+D,  
 10382 NEXT, J/ZEROSF  
 10383 (7043) DCS(0.00.0.0.0.0) BM(0000..00.00..00.10..000..000...0.0.0..0..0...0.0000...0..0011.0...11.000...000.100.100)

10384 7044: !(FREE)  
 10385 ZEROSF:

10386 P2-T, D+ZERO, SAVE-D(C),  
 10387 P3, A#BSPHI(SF)+D,  
 10388 NEXT, J/ZEROSFA  
 10389 (7044) DCS(0.00.0.0.0.0) BM(0011..00.01..00.11..000..111...0.1.0..0..0...0.0000...0..1011.0...11.000...000.100.101)

10390 7045: !(FREE)  
 10391 ZEROSFA:

10392 P3, A#BSPLO(SF)+D,  
 10393 NEXT, BUTA(RETURN),  
 10394 J/BUTERROR?  
 10395 (7045) DCS(0.00.0.0.0.0) BM(0000..00.01..00.11..000..000...0.0.0..0..0...0.0000...0..0011.0...11.111...011.111.110)

10396  
 10397  
 10398  
 10399

! - - - - -

10400  
 10401  
 10402  
 10403  
 10404  
 10405  
 10406  
 10407  
 10408  
 10409  
 10410  
 10411  
 10412  
 10413

THIS SECOND SUBROUTINE NOW READS BACK ALL THE REGISTERS IN ASP/BSP,  
 HI/LO, SF/DF THAT THE WRITE COULD HAVE REFERENCED.  
 IF LOCATION WAS WRITTEN, BIT<07> SHOULD BE SET,  
 ELSE IT SHOULD REMAIN CLEAR (FROM ZEROSFX(DF)X ROUTINE)  
 THE BITS ARE SHIFTED SERIALLY INTO THE SR, AND AFTER  
 TWO WRITES (WHICH GIVES 16. BITS, THE SR JUST FILLED),  
 THE COMPARE IS MADE BETWEEN EXPECTED:RECEIVED ANSWERS.  
  
 NOTE: THIS SUBR REQUIRES SR-LEFT/GUARD-DISABLED SETUP,  
 AND CSP(02) CONTAINS EXPECTED ANSWER.  
 CSP(00) CONTAINS RETURN ADDRESS, IN BITS<14:03>.

10414  
 10415  
 10416  
 10417  
 10418  
 10419  
 10420

5746:  
 SFDFTOSR:  
 PO, LOAD-ENUA(ZTARGET777),  
 LOAD-ERROR(SFDFTOSR),  
 DCS-CTR(C11.),  
 NEXT, GOTO-PAGE(7),  
 J/SFDFTOSRAA  
 (5746) DCS(1.00.1.0.0.0) BM(0100..00.11..11.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.100.000)

!COMPARE POINT IS AT EXIT  
 !THEORETICALLY THIS CODE SHOULD NEVER BE USED  
 !COMPARE AT TARGET  
 !SUBR IS HERE, NOTE OVERLAP W/ BIT<2:0> OF ENUA

10421  
 10422  
 10423  
 10424  
 10425

7040: !(FREE)  
 SFDFTOSRAA:  
 P2-T, D+ASPHI(DF), D(C)+ALU07,  
 NEXT, J/SFDFTOSRA  
 !SR<15,07> = ASPHI/DF

```

(7040) DCS(0.00.0.0.0.0) BM(1111..00.00..11.10..000..011...0.1.0..0..0...0.0000...0..0000.0...11.000...000.100.111)
10426
10427 7047: !(FREE)
10428 SFDFTOSRA:
10429 P2-T, SR+SR-LEFT-1, !STORE PAST
10430 D+ASPLO(DF), D(C)+ALU07, !SR<14,06> = ASPLO/DF
10431 NEXT J/SFDFTOSRB
(7047) DCS(0.00.0.0.0.0) BM(1111..00.00..10.10..000..011...0.1.1..0..0...0.0000...0..0000.0...11.000...000.101.000)
10432
10433 7050: !(FREE)
10434 SFDFTOSRB:
10435 P2-T, SR+SR-LEFT-1, !STORE PAST
10436 D+BSPHI(DF), D(C)+ALU07, !SR<13,05> = BSPHI/DF
10437 NEXT J/SFDFTOSRC
(7050) DCS(0.00.0.0.0.0) BM(1010..01.00..00.00..000..011...0.1.1..0..0...0.0000...0..0000.0...11.000...000.101.001)
10438
10439 7051: !(FREE)
10440 SFDFTOSRC:
10441 P2-T, SR+SR-LEFT-1, !STORE PAST
10442 D+BSPLO(DF), D(C)+ALU07, !SR<12,04> = BSPLO/DF
10443 NEXT J/SFDFTOSRD
(7051) DCS(0.00.0.0.0.0) BM(1010..00.00..00.00..000..011...0.1.1..0..0...0.0000...0..0000.0...11.000...000.101.010)
10444
10445 7052: !(FREE)
10446 SFDFTOSRD:
10447 P2-T, SR+SR-LEFT-1, !STORE PAST
10448 D+ASPFI(SF), D(C)+ALU07, !SR<11,03> = ASPFI/SF
10449 NEXT J/SFDFTOSRE
(7052) DCS(0.00.0.0.0.0) BM(1111..00.00..11.11..000..011...0.1.1..0..0...0.0000...0..0000.0...11.000...000.101.011)
10450
10451 7053: !(FREE)
10452 SFDFTOSRE:
10453 P2-T, SR+SR-LEFT-1, !STORE PAST
10454 D+ASPLO(SF), D(C)+ALU07, !SR<10,02> = ASPLO/SF
10455 NEXT J/SFDFTOSRF
(7053) DCS(0.00.0.0.0.0) BM(1111..00.00..10.11..000..011...0.1.1..0..0...0.0000...0..0000.0...11.000...000.101.100)
10456
10457 7054: !(FREE)
10458 SFDFTOSRF:
10459 P2-T, SR+SR-LEFT-1, !STORE PAST
10460 D+BSPHI(SF), D(C)+ALU07, !SR<09,01> = BSPHI/SF
10461 NEXT J/SFDFTOSRG
(7054) DCS(0.00.0.0.0.0) BM(1010..01.01..00.00..000..011...0.1.1..0..0...0.0000...0..0000.0...11.000...000.101.101)
10462
10463 7055: !(FREE)
10464 SFDFTOSRG:
10465 P2-T, SR+SR-LEFT-1, !STORE PAST
10466 D+BSPLO(SF), D(C)+ALU07, !SR<08,00> = BSPLO/SF
10467 NEXT J/SFDFTOSRH
(7055) DCS(0.00.0.0.0.0) BM(1010..00.01..00.00..000..011...0.1.1..0..0...0.0000...0..0000.0...11.000...000.101.110)
10468
10469 7056: !(FREE)
10470 SFDFTOSRH:
10471 P2-T, SR+SR-LEFT-1, !STORE PAST

```

```

10472          D+CSPD(D00), D(C)+0,          !RETRIEVE RETURN ADDRESS
10473          NEXT J/SFDFTOSRI
(7056) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...0.1.1..0..0...0.1111...0..0000.0...11.000...000.101.111)

```

```

10474
10475          7057: !(FREE)
10476          SFDFTOSRI:
10477          PO, RETURN=D(14-03), PAGE(7),          !PUT RETURN ADDRESS INTO RETURN REGISTER,
10478          P2-T, D+SR-XOR-CSPD(D02), SAVE-D(C),          !COMPARE RECEIVED:EXPECTED
10479          NEXT J/ZTARGET777          !GO DO A BUTA(RETURN)
(7057) DCS(0.00.0.0.0.0) BM(0110..10.00..00.00..000..111...0.1.0..0..0...0.1101...0..0000.0...11.101...111.111.111)

```

10480  
10481  
10482  
10483  
10484

!.PAGE=====

10485  
10486  
10487

.TOC \* TEST375-376: BYTE WRITE TO ASP/BSP LO, SP ADDRS R-OR-1/FLTPT-INHIBIT

10488  
10489  
10490

\*\*\*\*\*

10491  
10492  
10493  
10494

TESTS: 375A - 376 UWORDS: 020 + 017

10495  
10496  
10497

FUNCTIONS:

10498  
10499  
10500

THE FOLLOWING SET OF THREE CHECKS TESTS THE FOLLOWING FEATURES OF THE ASP/BSP:  
TEST 375 A/B TEST THAT BYTE WRITES CAN BE PERFORMED TO ASP/BSP LO, USING THE "DAD" EXTENSION BIT COMBINATION.

10501  
10502  
10503  
10504

TEST 376 VERIFIES THAT A FLOATING POINT INSTRUCTION IN THE IR FORCES SF-ADDRS BIT<02> TO A (0), AND THAT BUTA(R-OR-1) IN THE REFERENCING MICROWORD FORCES BIT<00> TO A (1).

10505  
10506  
10507

\*\*\*\*\*

10508  
10509  
10510

!TEST 375 A CHECKS THAT THE DAD BIT COMBINATION /11 DOES A BYTE WRITE (IE, LO BYTE ONLY)

10511  
10512  
10513  
10514

5517:  
TEST375A:  
PO, LOAD-ENUA(ZTARGET432), !SETUP FOR INSTRS IR=(000125)/E78/432  
LOAD-ERROR(TEST375A), !ERROR DIRECTORY KEY  
DCS-CTR(C11.), !COMPARE AT TARGET  
NEXT J/SETSP375A

10515  
10516

(5517) DCS(1.00.1.0.0.0) BM(0100..00.11..11.00..011..010...0.0.0..0..0...0.0000...0..0000.0...11.000...101.000.110)

10517  
10518  
10519

5506:  
SETSP375A:  
P2-T, D+CSPD(C125252), !FIRST SETUP SP'S WITH JUNK  
P3, A#BSPL0(06)+D-[A],  
NEXT, GOTO-PAGE(7), !XFER FOR DCS-DAD BITS

10520  
10521  
10522



```

10523 J/FIRST375A
(5506) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..111..111...0.1.0..0..0...0.0110...0..0011.0...11.100...000.100.110)
10524
10525 7046: !(FREE)
10526 FIRST375A:
10527 SETUP, FIRST-1-OR-2, !DAD/01, SHOULDN'T CAUSE BYTE WRITE
10528 P2-T, D+ZERO !SHOULD WRITE ALL ZEROES
10529 P3, ASPL0(06)+D
10530 NEXT, J/SECOND375A
(7046) DCS(0.01.0.0.0.0) BM(0011..00.00..00.00..111..000...0.1.0..0..0...0.0000...0..0001.0...11.000...000.110.001)
10531
10532 7061: !(FREE)
10533 SECOND375A:
10534 SETUP, SECOND-1-OR-2, !DAD/10, SHOULDN'T CAUSE BYTE WRITE
10535 P2-T, D+ZERO !SHOULD WRITE ALL ZEROES
10536 P3, BSPL0(06)+D,
10537 NEXT, J/BYTE375A
(7061) DCS(0.10.0.0.0.0) BM(0011..00.10..00.00..111..000...0.1.0..0..0...0.0000...0..0110.0...11.000...000.110.010)
10538
10539 7062: !(FREE)
10540 BYTE375A:
10541 SETUP, BYTE-WRITE, !DAD/11, SHOULD CAUSE A BYTE WRITE
10542 P2-T, D+ASPHI(C052525), !WRITE THE (125) IN THE LOW BYTE
10543 P3, A#BSPL0(06)+D-(B), !UPPER BYTE SHOULD BE (000) FROM ABOVE
10544 NEXT, J/CHECK375A
(7062) DCS(0.11.0.0.0.0) BM(1111..00.10..11.01..111..000...0.1.0..0..0...0.0000...0..0111.0...11.000...000.110.011)
10545
10546 7063: !(FREE)
10547 CHECK375A:
10548 P2-T, D+ASPL0(R06), !GET THE A SIDE SP
10549 NEXT, J/GOBUT375A
(7063) DCS(0.00.0.0.0.0) BM(1111..00.00..10.00..111..000...0.1.0..0..0...0.0000...0..0000.0...11.000...000.110.100)
10550
10551 7064: !(FREE)
10552 GOBUT375A:
10553 SETUP, RETURN/TEST375B, !RETURN TO START OF NEXT SUBTEST
10554 NEXT, CALL(DINTOIR-5) !GO CHECK (000125) OBTAINED
(7064) DCS(0.00.0.0.0.0) BM(0101..00.11..00.10..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)
10555
10556
10557
10558
10559
10560 !TEST 375 B NOW CHECKS THE SAME THING (000125) IS ON THE B SIDE
10561 5625:
10562 TEST375B:
10563
10564 PO, LOAD-ENVA(ZTARGET432), !SETUP FOR INSTRS IR=(000125)/E78/432
10565 LOAD-ERROR(TEST375B), !ERROR DIRECTORY KEY
10566 DCS-CTR(C7.), !COMPARE AT TARGET
10567 NEXT, J/CHECK375B
(5625) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..011..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.000.000)
10568

```

10569 5300: !(FREE)  
 10570 CHECK3758:  
 10571 P2-T, D+BSPL0(R06), !GET THE B SIDE SP  
 10572 NEXT, J/G08UT3758  
 (5300) DCS(0.00.0.0.0.0) BM(1010..00.10..00.00..111..000...0.1.0..0..0...0.0000...0..0000.0...11.000...011.000.001)

10573 5301: !(FREE)  
 10574 G08UT3758:  
 10575 SETUP, RETURN/TEST376A, !RETURN TO START OF NEXT SUBTEST  
 10576 NEXT, CALL(DINTOIR-5) !GO CHECK (000125) OBTAINED  
 10577 (5301) DCS(0.00.0.0.0.0) BM(0101..00.10..10.01..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

10578  
 10579  
 10580  
 10581  
 10582  
 10583  
 10584  
 10585

!TEST 376 A NOW DOES THE SF ADDRESS MODE, W/ FLTPT-INHIBIT AND BUTA(R-OR-1) ACTIVE  
 !NOTE:  
 THE ASPHI CONTAINS THE FOLLOWING VALUES IN THESE LOCATIONS:

ASPHI(02)	ASPHI(03)	ASPHI(06)	ASPHI(07)
(052522)	(177777)	(056166)	(052525)

10593 5513:  
 10594 TEST376A:  
 10595 PO, LOAD-EMUA(DOIT376A), !MAKE SURE BUTA(R-OR-1) DOESN'T CAUSE A BRANCH  
 10596 LOAD-ERROR(TEST376A), !ERROR DIRECTORY KEY  
 10597 DCS-CTR(C2.), !COMPARE AT TARGET  
 10598 NEXT, J/SETIR376A  
 10599 (5513) DCS(1.00.1.0.0.0) BM(1101..00.10..11.00..100..000...0.0.0..0..0...0.0000...0..0000.0...11.000...011.000.010)

10600 5302: !(FREE)  
 10601 SETIR376A:  
 10602 SETUP, BUTA(R-OR-1), !SETUP ACTIVE BUT MODIFICATION OF SF BIT(00) ADDRESS  
 10603 P2-U, IR+EMIT, EMIT/170600, !IR=FLTPT INSTR, SF=(6)  
 10604 NEXT, J/DOIT376A  
 10605 (5302) DCS(0.00.0.0.0.0) BM(1111..00.00..01.10..000..000...0.0.0..0..0...1.1010...0..0000.0...10.010...100.100.000)

10606 5440:  
 10607 DOIT376A:  
 10608 P2-T, D+A-XOR-B, !COMPARE SF/OBTAINED:REGISTER EXPECTED  
 10609 BUS-A+ASPHI(SF), !SF ON A GOES FROM (6) -> (2) FROM FLTPT,  
 10610 AND FROM (2) -> (3) FOR (R-OR-1)  
 10611 BUS-B+BSPHI(R03), !THIS WE EXPECT  
 10612 NEXT, J/TEST376A  
 10613 (5440) DCS(0.00.0.0.0.0) BM(0110..01.11..11.11..101..000...0.1.0..0..0...0.0000...0..0000.0...11.000...110.001.100)

10614  
 10615  
 10616

```

10617 !NOW CHECK THE RIGHT RESULT WAS OBTAINED
10618 5614:
10619 TEST376A1:
10620 PO, LOAD-ENVA(ZTARGET434), !SETUP FOR IR=(000000)/INSTRS COMPARE
10621 LOAD-ERROR(TEST376A1), !ERROR DIRECTORY KEY
10622 DCS-CTR(C6.), !COMPARE AT TARGET
10623 NEXT, J/GOBUT376A
(5614) DCS[1.00.1.0.0.0] BM[1001..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...011.000.011]

10624 5303: !(FREE)
10625 GOBUT376A:
10626 SETUP, RETURN/SCOPE376, !RETURN TO SCOPE LOOP TEST WORD
10627 NEXT, CALL(DINTOIR-5) !CHECK FOR (000000)
10628 (5303) DCS[0.00.0.0.0.0] BM[0101..00.01..10.00..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011]

10629
10630
10631 5304: !(FREE)
10632 SCOPE376:
10633 PO, BUMP-VERIFY, !COUNT
10634 NEXT, BUTD(SCOPE), !NO ERROR: "TEST410" (+1. WORDS)
10635 J/TEST410 !ERROR: "SETSP375A" (-12. WORDS)
10636 (5304) DCS[0.00.0.1.0.1] BM[0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...101.000.111]

```

```

10637
10638
10639
10640
10641

```

!.PAGE=====

.TOC \* TEST410: BYTE/BYTE CONSTANT/D=ZERO

```

*****
*
* TESTS: 410 A - E UWORDS: 044 + 020
*
* FUNCTIONS:
*
* THE FOLLOWING TESTS RUN A COUNT PATTERN THRU THE IR, MAINTAINING TOTALS OF
* THE NUMBER OF TIMES:
*
* BYTE-H=LOW, BYTE/1-OR-2-FIRST=HIGH, BYTE/1-OR-2-SECOND=HIGH,
* (D=ZERO)-H=HIGH, AND (D=ZERO)-H=LOW.
*
* AT THE END, THE TESTS COMPARE THE EXPECTED COUNTS TO THE RECEIVED COUNTS.
*
*****

```

```

10664
10665 5507:
10666 TEST410:

```

```

10667      PO,      LOAD-ERROR(TEST410),      !ERROR DIRECTORY KEY
10668      DCS-CTR(C6.),      !COMPARE BELOW
10669      NEXT,     GOTO-PAGE(6),      !XFER
10670      J/SETBYTEB410
(5507) DCS(1.00.1.0.0.0) BM(1001..00.00..00.00..000..110...0.0.0..0..0...0.0000...0..0000.0...11.100...101.110.000)

10671
10672      6560:
10673      SETBYTEB410:
10674      P2-T,      D+ZERO      !ZER0ES INTO:
10675      P3,      A#BSPHI(10)+D-(A),      !ASPHI(10) = BYTE-FIRST
10676      NEXT,     GOTO-PAGE(7),      !
10677      J/SETBYTEC410      !BSPHI(10) = WORD (=BYTE)
(6560) DCS(0.00.0.0.0.0) BM(0011..00.00..00.00..000..111...0.1.0..0..0...0.0000...0..1011.0...11.100...000.110.000)

10678
10679      7060: !(FREE)
10680      SETBYTEC410:
10681      P3,      A#BSPLO(10)+D-(B),      !ASPLO(10) = BYTE-SECOND
10682      NEXT,     J/SETBYTED410      !BSPLO(10) = IR-DATA
(7060) DCS(0.00.0.0.0.0) BM(0000..00.10..00.00..000..000...0.0.0..0..0...0.0000...0..0111.0...11.000...000.110.110)

10683
10684      7066: !(FREE)
10685      SETBYTED410:
10686      P3,      A#BSPLO(11)+D-(B),      !ASPLO(11) = ?-NONZERO
10687      NEXT,     GOTO-PAGE(6),      !BSPLO(11) = D-ZERO
10688      J/SETBYTEE410
(7066) DCS(0.00.0.0.0.0) BM(0000..00.11..00.00..000..110...0.0.0..0..0...0.0000...0..0111.0...11.100...000.000.000)

10689
10690      6000: !(FREE)
10691      SETBYTEE410:
10692      PO,      BUMP-VERIFY,      !COUNT
10693      P3,      CSPD(17)+EMIT, EMIT/000C01,      !A (1) IN BYTE-CONSTANT
10694      NEXT,     J/SETBYTEG410      !LOCATION IN CSP
(6000) DCS(0.00.0.0.0.1) BM(0000..10.00..00.00..000..001...0.0.0..0..0...0.0000...1..0000.0...11.000...001.000.101)

10695
10696      6105: !(FREE)
10697      SETBYTEG410:
10698      P2-U,      IR+DBUF-(I),      !SETUP UCONS FOR D --> IR
10699      P3,      DBUF+D-(I),      !
10700      NEXT,     J/TESTD410
(6105) DCS(0.00.0.0.0.0) BM(0100..00.00..00.01..000..100...0.0.0..0..0...1.1011...0..0000.0...11.000...111.111.011)

10701
10702
10703
10704      !*** LOOP BACK ENTRY POINT ***
10705
10706      6773:
10707      TESTD410:
10708      PO,      LOAD-ENUA(GOFOR410),      !COMPARE AT END OF LOOP
10709      LOAD-ERROR(TESTD410),      !ERROR DIRECTORY KEY
10710      DCS-CTR(C15.),      !RELOAD EACH TIME THROUGH
10711      P2-U,      IR+DBUF,      !(DON'T CARE HERE)

```

```

10712          P3,      DBUF+D,          !COPY IR-DATA FROM D --> DBUF
10713          NEXT,    BUTR(D14-00-EQ-0), !TEST D<14:00>H:
10714          J/DNONZERO410          !ZERO --> DZER0410, NONZERO --> DNONZERO410
(6773) DCS[1.00.1.0.0.0] BM[0000..00.11..10.11..110..001...0.0.0..0..0...1.1010...0..0000.0...01.101...101.011.001]

10715          !ENTER HERE IF D<14:00> WAS DETECTED AS NON-ZERO
10716          6531:
10717          DNONZERO410:
10718          P2-T,      D+ASPL0(DNONZERO)-PLUS-1,          !BUMP NON ZERO COUNTER
10719          P2-U,      IR+DBUF,          !COPY IR-DATA FROM DBUF --> IR
10720          P3,      ASPLO[11]+0,          !SAVE NON ZERO COUNTER
10721          DBUF+D,          (DON'T CARE HERE)
10722          NEXT,      GOTO-PAGE(7),          !XFR TO 7 FOR DAD
10723          J/NEXTPAT410
10724          (6531) DCS[0.00.0.0.0.0] BM[1001..01.11..10.01..000..111...0.1.0..0..0...1.1010...0..0001.0...11.100...000.110.101]

10725          !ENTER HERE IF D<14:00> WAS DETECTED AS ZERO
10726          6533:
10727          DZER0410:
10728          P2-T,      D+BSPLO(DZERO)-PLUS-1,          !BUMP ZERO COUNTER
10729          P2-U,      IR+DBUF,          !COPY IR-DATA FROM DBUF --> IR
10730          P3,      BSPLO[11]+0,          !SAVE ZERO COUNTER
10731          DBUF+D,          (DON'T CARE NOW)
10732          NEXT,      GOTO-PAGE(7),          !XFR TO 7 FOR DAD
10733          J/NEXTPAT410
10734          (6533) DCS[0.00.0.0.0.0] BM[1001..00.11..11.01..000..111...0.1.0..0..0...1.1010...0..0110.0...11.100...000.110.101]

10735          7065: !(FREE)
10736          NEXTPAT410:
10737          SETUP,      FIRST-1-OR-2,          !SELECT DAD BITS FOR "BYTEFIRST410"
10738          P2-T,      D+BSPLO(IR-DATA)-PLUS-2,          ! (INCR DATA FOR -NEXT- TIME THRU)
10739          D[C]+COUT15,          !WHEN THIS SETS WE'RE DONE
10740          P3,      BSPLO[10]+0,          !SAVE NEXT
10741          NEXT,      J/BYTEFIRST410
10742          (7065) DCS[0.01.0.0.0.0] BM[1100..00.10..11.01..000..110...0.1.0..0..0...0.0000...0..0110.0...11.000...000.111.000]

10743          7070: !(FREE)
10744          BYTEFIRST410:
10745          SETUP,      SECOND-1-OR-2,          !SELECT DAD BITS FOR "BYTESECOND410"
10746          P2-T,      D+ASPHI(BYTE-FIRST)-PLUS-CSP[1-0], !BYTE-FIRST SELECTS EITHER
10747          SAVE-D[C],          !CSP(17)=1 OR CSP(13)=0
10748          P3,      ASPHI[10]+0,          !WRITE BACK
10749          NEXT,      BUTR(BYTE),          !BYTE --> "BYTESECOND410"
10750          J/WORD410,          !-BYTE --> "WORD410"
10751          (7070) DCS[0.10.0.0.0.0] BM[1001..10.00..11.00..000..111...0.1.0..0..0...0.0100...0..1001.0...01.001...011.110.110]

10752          !ENTER HERE IF BYTE-H NOT ASSERTED, IE IR=(WORD)
10753          7366:
10754          WORD410:
10755          SETUP,      SECOND-1-OR-2,          !SELECT DAD BITS FOR "BYTESECOND410"
10756          P2-T,      D+BSPHI(WORD)-PLUS-1,          !BUMP WORD=-BYTE COUNTER
10757          SAVE-D[C],          !SAVE PAST CARRYOUT STATUS
10758

```

```

10759          P3,      BSPHI(10)+D,      !WRITE BACK
10760          NEXT,    J/BYTESECOND410    !NOW GO TRY BYTE-SECOND
(7366) DCS(0.10.0.0.0.0) BM(1001..01.10..11.01..000..111...0.1.0..0..0...0.0000...0..1110.0...11.000...011.110.111)

10761          !ENTER HERE IF BYTE-H WAS ASSERTED, IE IR=(BYTE)
10762          7367:
10763          BYTESECOND410:
10764          SETUP,   NO-DAD,              !KEEP FOR NOISE
10765          P2-T,    D+ASPL0(BYTE-SECOND)-PLUS-CSP(1-0), !BYTE-SECOND SELECTS EITHER
10766          P2-T,    SAVE-D(C),          !CSP(17)=1 OR CSP(13)=0
10767          P3,      ASPLO(10)+D,        !WRITE BACK
10768          NEXT,    BUTR(D(C)-8),       !IF SET, SKIP OUT TO TEST-410A
10769          J/GOFOR410                    !IF CLEAR, FALL THRU TO NEXT
10770 (7367) DCS(0.00.0.0.0.0) BM(1001..10.00..10.00..000..111...0.1.0..0..0...0.0100...0..0001.0...10.011...011.110.001)

10771          7361:
10772          GOFOR410:
10773          P2-T,    D+BSPLO(IR-DATA),    !GET DATA FOR IR INTO D
10774          NEXT,    GOTO-PAGE(6),
10775          J/TESTD410                    !LOOP BACK FOR NEXT
10776 (7361) DCS(0.00.0.0.0.0) BM(1010..00.10..00.00..000..110...0.1.0..0..0...0.0000...0..0000.0...11.100...111.111.011)

10777
10778
10779
10780
10781          ! - - - - -
10782
10783          !TEST 410A CHECKS THAT D<14:00>=ZERO WAS ONLY ASSERTED TWICE
10784          7363:
10785          EXPEC410A:
10786          NEXT,    GOTO-PAGE(6),
10787          J/TEST410A                    !FOR LOADING DCS-CTR
(7363) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..110...0.0.0..0..0...0.0000...0..0000.0...11.100...111.001.001)

10788
10789          6711:
10790          TEST410A:
10791          PO,      LOAD-ENVA(ZTARGET422), !FOR IR=(000002) W/INSTR5
10792          LOAD-ERROR(TEST410A),        !ERROR DIRECTORY KEY
10793          DCS-CTR(C7.),                !COMPARE AT TARGET
10794          NEXT,    J/COMP410A
(6711) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..010..010...0.0.0..0..0...0.0000...0..0000.0...11.000...001.000.110)

10795          6106: !(FREE)
10796          COMP410A:
10797          P2-T,    D+BSPLO(DZERO),      !GET DATA
10798          NEXT,    J/INTOIR410A
(6106) DCS(0.00.0.0.0.0) BM(1010..00.11..00.00..000..000...0.1.0..0..0...0.0000...0..0000.0...11.000...001.000.111)

10800          6107: !(FREE)
10801          INTOIR410A:
10802          SETUP,  RETURN/TEST410B,
10803          NEXT,   CALL(DINTOIR-5)
10804

```

(6107) DCS(0.00.0.0.0.0) BM(0110..00.11..10.00..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

10805  
10806  
10807  
10808  
10809  
10810  
10811  
10812  
10813  
10814  
10815  
10816  
10817  
10818  
10819

! - - - - -

!TEST 4108 CHECKS THAT D<14:00>=ZERO WAS NOT ASSERTED  
! 32768.-2. = 32766. (077776) TIMES

6704:

TEST4108:

PO, LOAD-ENUA(ZTARGET434), !FOR IR=(00000) W/INSTR5  
LOAD-ERROR(TEST4108), !ERROR DIRECTORY KEY  
DCS-CTR(C8.), !COMPARE AT TARGET  
BUMP-VERIFY, !COUNT  
NEXT, J/EXPEC4108

(6704) DCS(1.00.1.0.0.1) BM(0111..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...001.001.000)

10820  
10821  
10822  
10823  
10824

6110: !(FREE)

EXPEC4108:

P3, CSPD(17)+EMIT, EMIT/077776, !EXPECTED NUMBER OF TIMES  
NEXT, J/COMP4108 !D<14:00> WAS NON ZERO

(6110) DCS(0.00.0.0.0.0) BM(0111..10.11..11.11..111..110...0.0.0..0..0...0.0000...1..0000.0...11.000...001.001.001)

10825  
10826  
10827  
10828  
10829

6111: !(FREE)

COMP4108:

P3-T, D+ASPLO(DNONZERO)-MINUS-CSPD(17), !COMPARE RECEIVED:EXPECTED  
NEXT, J/GOPUT4108

(6111) DCS(0.00.0.0.0.0) BM(1101..10.00..10.01..000..000...1.1.0..0..0...0.0000...0..0000.0...11.000...001.001.010)

10830  
10831  
10832  
10833  
10834

6112: !(FREE)

GOPUT4108:

SETUP, RETURN/TEST410C, !GO PUT D --> IR  
NEXT, CALL(DINTOIR-5) ! AND CHECK IT-S ALL ZERO

(6112) DCS(0.00.0.0.0.0) BM(0110..00.11..10.00..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

10835  
10836  
10837  
10838  
10839  
10840

! - - - - -

!TEST 410C CHECKS THAT BYTE-H WAS NOT ASSERTED 21696.,  
!OR WAS ASSERTED 11072. TIMES IN 32768. ITERATIONS

6707:

TEST410C:

PO, LOAD-ENUA(ZTARGET402), !SETUP FOR D=ZERO TEST  
LOAD-ERROR(TEST410C), !ERROR DIRECTORY KEY  
DCS-CTR(C6.), !COMPARE AT TARGET  
NEXT, J/EXPEC410C

(6707) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...001.001.011)

10849  
10850

6113: !(FREE)

```

10851 EXPEC410C:
10852 P3, CSPD(17)+EMIT, EMIT/52300, !32768.-#TIMES BYTE-H ASSERTED
10853 NEXT, J/ASIDE410C ! 216%=(52300)
(6113) DCS(0.00.0.0.0.0) BM(0101..10.01..00.11..000..000...0.0.0..0..0...0.0000...1..0000.0...11.000...001.001.100)

10854 6114: !(FREE)
10855 ASIDE410C:
10856 P2-T, SR+CSPD(017), !GET ONTO A-SIDE
10857 NEXT, J/COMP410C
10858 (6114) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...0.0.1..0..0...0.0000...0..0000.0...11.000...001.001.101)

10859 6115: !(FREE)
10860 COMP410C:
10861 P3-T, D+SR-MINUS-BSPHI(WORD), !COMPARE EXPEC:RECEIVED
10862 NEXT, J/GOBUT410C
10863 (6115) DCS(0.00.0.0.0.0) BM(1101..01.10..00.00..000..000...1.1.0..0..0...0.0000...0..0000.0...11.000...001.001.110)

10864 6116: !(FREE)
10865 GOBUT410C:
10866 SETUP, RETURN/TEST4100, !RETURN TO START OF NEXT SUBTEST
10867 NEXT, GOTO-PAGE(7) !BUT TABLE
10868 J/BUTD-IS-ZERO !GO CHECK EQUALITY
10869 (6116) DCS(0.00.0.0.0.0) BM(0110..00.11..10.10..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

10870
10871
10872
10873
10874 ! - - - - -
10875
10876 !TEST 4100 CHECKS THAT BYTE-CONSTANT WAS ASSERTED (4270)=2232.
10877 !TIMES UNDER "FIRST-1-OR-2"
10878 6727:
10879 TEST4100:
10880 PO, LOAD-ENUA(ZTARGET402), !SETUP FOR D=ZERO TEST
10881 LOAD-ERROR(TEST4100), !ERROR DIRECTORY KEY
10882 DCS-CTR(CS.), !COMPARE AT TARGET
10883 NEXT, J/EXPEC4100
(6727) DCS(1.00.1.0.0.0) BM(1010..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...001.001.111)

10884 6117: !(FREE)
10885 EXPEC4100:
10886 PO, BUMP-VERIFY !COUNT
10887 P3, CSPD(17)+EMIT, ! (4270)=2232. TIMES FOR FIRST
10888 EMIT/004270,
10889 NEXT, J/COMP4100
10890 (6117) DCS(0.00.0.0.0.1) BM(0000..10.10..00.10..111..000...0.0.0..0..0...0.0000...1..0000.0...11.000...001.010.000)

10891 6120: !(FREE)
10892 COMP4100:
10893 P3-T, D+ASPHI(BYTE-FIRST)-MINUS-CSPD(17), !COMPARE RECEIVED:EXPEC
10894 NEXT, J/GOBUT4100
10895 (6120) DCS(0.00.0.0.0.0) BM(1101..10.00..11.00..000..000...1.1.0..0..0...0.0000...0..0000.0...11.000...001.010.001)

10896

```



```

10897 6121: !(FREE)
10898 GOBUT4100:
10899     SETUP, RETURN/TEST410E, !RETURN TO START OF NEXT SUBTEST
10900     NEXT,  GOTO-PAGE(7), !BUT TABLE
10901     J/BUTD-IS-ZERO !CHECK EQUALITY
(6121) DCS(0.00.0.0.0.0) BM(0110..00.11..10.00..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.0011)

10902
10903
10904
10905
10906 ! - - - - -
10907
10908 !TEST 410E CHECKS THAT BYTE-CONSTANT WAS ASSERTED (3600)=1920.
10909 !TIMES UNDER "SECOND-1-OR-2"
10910 6705:
10911 TEST410E:
10912     PD, LOAD-ENVA(ZTARGET402), !SETUP FOR D=ZERO TEST
10913     LOAD-ERROR(TEST410E), !ERROR DIRECTORY KEY
10914     DCS-CTR(CS.), !COMPARE AT TARGET
10915     NEXT, J/EXPEC410E
(6705) DCS(1.00.1.0.0.0) BM(1010..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...001.010.0101)

10916
10917 6122: !(FREE)
10918 EXPEC410E:
10919     PD, BUMP-VERIFY, !COUNT
10920     P3, CSPD[17]+EMIT, !
10921     EMIT/003600, ! (3600)=1920. TIMES FOR SECOND
10922     NEXT, J/COMP410E
(6122) DCS(0.00.0.0.0.1) BM(0000..10.01..11.10..000..000...0.0.0..0..0...0.0000...1..0000.0...11.000...001.010.0111)

10923
10924 6123: !(FREE)
10925 COMP410E:
10926     P3-T, D+ASPLO[BYTE-SECOND]-MINUS-CSPD[17], !COMPARE RECEIVED:EXPEC
10927     NEXT, J/GOBUT410E
(6123) DCS(0.00.0.0.0.0) BM(1101..10.00..10.00..000..000...1.1.0..0..0...0.0000...0..0000.0...11.000...001.010.1001)

10928
10929 6124: !(FREE)
10930 GOBUT410E:
10931     SETUP, RETURN/SCOPE410, !RETURN TO SCOPE LOOP TEST WORD
10932     NEXT, GOTO-PAGE(7), !BUT TABLE
10933     J/BUTD-IS-ZERO !CHECK EQUALITY
(6124) DCS(0.00.0.0.0.0) BM(0110..00.00..10.10..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.0011)

10934
10935
10936 6125: !(FREE)
10937 SCOPE410:
10938     PD, BUSDIN+EMIT-[1], !RESET PROC UCON
10939     EN-CLK-IR[15-00], !
10940     NEXT, BUTD[SCOPE], !NO ERROR: "TEST500" (+1. WORDS)
10941     J/TEST500 !ERROR: "SETBYTEB410" (-34. WORDS)
(6125) DCS(0.00.0.1.0.0) BM(0000..00.00..00.01..000..100...0.0.0..0..0...1.1001...0..0000.0...11.000...101.110.0011)

10942

```

10943  
10944  
10945  
10946  
10947  
10948  
10949  
10950  
10951  
10952  
10953  
10954  
10955  
10956  
10957  
10958  
10959  
10960  
10961  
10962  
10963  
10964  
10965  
10966  
10967  
10968  
10969  
10970  
10971  
10972  
10973  
10974  
10975  
10976  
10977  
10978  
10979  
10980  
10981  
10982  
10983  
10984  
10985  
10986  
10987  
10988  
10989  
10990  
10991  
10992

!.PAGE=====

!\*\*\* VERSION /V101AD/ \*\*\*

.TOC \* TEST500: PREFETCH/OVERLAP/SP DEFEAT

!\*\*\*\*\*  
!\*  
!\* TESTS: 500C - 500F UWORDS: 034 + 013  
!\*  
!\* FUNCTIONS: TESTS 500C - 500F RUN PATTERNS THRU THE IR TO TEST  
!\* THE PREFETCH/OVERLAP ROM & THE SP WRITE DEFEAT LOGIC.  
!\*  
!\*\*\*\*\*

6561:

TEST500:

PO, LOAD-ENVA(SETFETCH500), !INTERMEDIATE COMPARE AT START OF LOOP  
LOAD-ERROR(TEST500), !ERROR DIRECTORY KEY  
DCS-CTR(CS.) !COMPARE AT TARGET  
NEXT, J/SETFETCH500

(6561) DCS(1.00.1.0.0.0) BM(1010..00.11..00.01..011..001...0.0.0..0..0...0.0000...0..0000.0...11.000...110.111.110)

6676:

SETFETCH500:

P2-T, D+ZERO, D(C)+ALU15, !ZERO  
SR+ZERO !TO FLAG JAMUPPS AS ILLEGAL -- KEEP ZERO THRU "TEST505"  
P3, A#BSPL0(OVERLAP)+D, !ZERO OVERLAP -A, -B COUNTERS  
NEXT, J/SETFETCH500

(6676) DCS(0.00.0.0.0.0) BM(0011..00.00..10.00..000..100...0.1.1..0..0...0.0000...0..0011.0...11.000...001.010.110)

6126: !(FREE)

SETFETCH500:

P3, A#BSPHI(PATTERN)+D, !ZERO PATTERN (A AND B)  
NEXT, J/SETFETCH500

(6126) DCS(0.00.0.0.0.0) BM(0000..00.00..11.01..011..000...0.0.0..0..0...0.0000...0..1011.0...11.000...001.010.111)

6127: !(FREE)

SETFETCH500:

P3, A#BSPHI(PREFETCH)+D, !ZERO PREFETCH-A COUNTER, ALSO B SIDE  
NEXT, J/SETFETCH500

(6127) DCS(0.00.0.0.0.0) BM(0000..00.00..11.00..000..000...0.0.0..0..0...0.0000...0..1011.0...11.000...001.011.000)

6130: !(FREE)

SETFETCH500:

```

10993          PO,      BUMP-VERIFY,      !COUNT
10994          P3,      CSPD(17)+020010,
10995          NEXT,     J/SETFETCH%500     !MASK CONSTANT FOR DATA PATTERN
(6130) DCS(0.00.0.0.0.1) BM(0010..10.00..00.00..001..000...0.0.0..0..0...0.0000...1..0000.0...11.000...001.011.001)

10996          6131:  !(FREE)
10997          SETFETCH%500:
10998          PO,      BUMP-VERIFY,      !COUNT
10999          P2-U,     IR+DBUF-[I],      !SETUP THESE UCON'S,
11000          P3,      DBUF+0-[I],      !DON'T WORRY ABOUT EFFECT HERE
11001          NEXT,     J/TESTD500
11002          (6131) DCS(0.00.0.0.0.1) BM(0100..00.00..00.01..000..100...0.0.0..0..0...1.1011...0..0000.0...11.000...110.111.001)

11003
11004
11005          !*** LOOP-BACK ENTRY POINT ***
11006
11007          6671:
11008          TESTD500:
11009          PO,      LOAD-ENVA(NEXTPAT500), !COMPARE POINT IN LOOP
11010          LOAD-ERROR(TESTD500),      !ERROR DIRECTORY KEY
11011          DCS-CTR(C4.),                !COMPARE AT TARGET
11012          P2-U,     IR+DBUF,            !JUST HAPPENS, DON'T CARE NOW
11013          P3,      DBUF+0,            !PUT PATTERN LEFT IN D INTO DBUF
11014          NEXT,     J/LOADIR500
(6671) DCS(1.00.1.0.0.0) BM(1011..00.11..00.11..111..111...0.0.0..0..0...1.1010...0..0000.0...11.000...001.011.010)

11015
11016          6132:  !(FREE)
11017          LOADIR500:
11018          P2-U,     IR+DBUF,            !PUT PATTERN NOW IN DBUF TO IR
11019          P3,      DBUF+0,            !JUST HAPPENS, DON'T CARE NOW
11020          NEXT,     J/TESTINHSP500
(6132) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...1.1010...0..0000.0...11.000...001.011.011)

11021
11022          6133:  !(FREE)
11023          TESTINHSP500:
11024          P2-T,     D+ASPLO(OVERLAP)-PLUS-1, !GET BUMPED OVERLAP COUNT, B-SIDE
11025          D(C)+0,
11026          P3,      BSPLO(OVERLAP)+0,    !ONLY WRITTEN BACK IF
11027          BUTA(INSTR-1),                !"OVERLAP L" ASSERTED
11028          NEXT,     J/TESTINHSP500     ! (BRANCH EFFECT OF BUT MASKED)
(6133) DCS(0.00.0.0.0.0) BM(1001..00.10..11.01..000..000...0.1.0..0..0...0.0000...0..0110.0...00.110...111.111.111)

11029
11030          6777:
11031          TESTINHSP500:
11032          P2-T,     D+ASPLO(OVERLAP)-PLUS-1, !GET BUMPED OVERLAP COUNT, A-SIDE
11033          SAVE-D(C),
11034          P3,      ASPLO(OVERLAP)+0,    !ONLY WRITTEN BACK IF
11035          BUTA(INSTR-1),                !"OVERLAP L" ASSERTED
11036          NEXT,     J/NEXTPAT500     ! (BRANCH EFFECT OF BUT MASKED)
(6777) DCS(0.00.0.0.0.0) BM(1001..01.11..10.00..000..111...0.1.0..0..0...0.0000...0..0001.0...00.110...011.111.111)

11037
11038          6377:
11039          NEXTPAT500:
11040          P2-T,     D+ASPHI(PATTERN)-PLUS-020010-PLUS-1, !GET NEXT DATA PATTERN IN D

```

```

11041          D(C)+COUNT,          !GET D(C)=OVERFLOW OUT OF BIT<15>
11042          P3,          A#BSPHI(PATTERN)+D,      !WRITE BACK UNMASKED NEXT PATTERN
11043          NEXT,        BUTR(PREFETCH-L),        !IF ASSERTED, "ASSERTFOVS00"
11044          J/ASSERTFOVS00          !OTHERWISE, "FIXPAT500"
(6377) DCS(0.00.0.0.0.0) BM(1100..11.00..11.01..011..110...0.1.0..0..0...0.0000...0..1011.0...10.000...101.010.101)

11045          !ENTER HERE IF "PREFETCH-L" WAS ASSERTED, LOW
11046          6525:
11047          ASSERTFOVS00:
11048          P2-T,          D+ASPHI(PREFETCH)-PLUS-1,      !"PREFETCH(0)" ASSERTED, BUMP COUNT
11049          SAVE-D(C),          (SAVE CARRYOUT)
11050          P3,          ASPHI(PREFETCH)+D,
11051          NEXT,        J/FIXPAT500
11052          (6525) DCS(0.00.0.0.0.0) BM(1001..01.11..11.00..000..111...0.1.0..0..0...0.0000...0..1001.0...11.000...101.010.111)

11053          !ENTER HERE IF "PREFETCH-L" WAS NOT ASSERTED, HIGH
11054          6527:
11055          FIXPAT500:
11056          P2-T,          D+ASPHI(PATTERN)-AND-NOT-020010,      !CLEAR UNNEEDED BITS
11057          SAVE-D(C),          !KEEP D(C) CONSTANT FOR "BUT"
11058          P3,          A#BSPHI(PATTERN)+D,          !RESTORE PATTERN, ALSO LEAVE IN D
11059          NEXT,        BUTR(D(C)-B),          !TEST CARRYOUT:
11060          J/TESTD500          !IF CLEAR, NEXT PATTERN AT: "TESTD500" (-6.WORDS)
11061          (6527) DCS(0.00.0.0.0.0) BM(0111..11.00..11.01..011..111...0.1.0..0..0...0.0000...0..1011.0...10.011...110.111.001)
          !IF SET, EXIT LOOP TO: "TEST500A" (+1.WORD)

11062          ! - - - - -
11063
11064          !*** TEST 500A ***
11065          !CHECK THAT REGISTER OPPOSITE ASPHI(PREFETCH), IN BSPHI, STILL ZERO
11066          6673:
11067          TEST500A:
11068          PO,          LOAD-ENVA(ZTARGET434),          !SETUP FOR IR=(000000)/INSTR5 TEST
11069          LOAD-ERROR(TEST500A),          !ERROR DIRECTORY KEY
11070          DCS-CTR(C7.),          !COMPARE AT TARGET
11071          BUMP-VERIFY,          !COUNT
11072          NEXT,        J/COMP500A
11073          (6673) DCS(1.00.1.0.0.1) BM(1000..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...001.011.100)

11076          6134: !(FREE)
11077          COMP500A:
11078          P2-T,          D+BSPHI(PREFETCH),          !GET REGISTER
11079          SAVE-D(C),
11080          NEXT,        J/GOBUT500A
11081          (6134) DCS(0.00.0.0.0.0) BM(1010..01.10..00.00..000..111...0.1.0..0..0...0.0000...0..0000.0...11.000...001.011.101)

11082          6135: !(FREE)
11083          GOBUT500A:
11084          PO,          BUMP-VERIFY,          !COUNT
11085          SETUP,        RETURN/TEST500C,          !RETURN TO START OF NEXT SUBTEST
11086          NEXT,        CALL(DINTOIR-5)          !PUT D -> D#UF -> IR, DO INSTR5 BUT FOR (000000)
11087          (6135) DCS(0.00.0.0.0.1) BM(0110..00.11..10.10..110..111...0.0.0..C..0...0.0000...0..0000.0...11.100...010.111.011)

```

!\*\*\* ALSO SETS UP BUSDIN <- EMIT UPON EXIT !\*\*

11088  
11089  
11090  
11091  
11092  
11093  
11094  
11095

!\*\*\* TEST 500C \*\*\*  
!CHECK THAT ASSERTING "OVERLAP L" ONLY ALLOWED A SP REWRITE (DURING  
!A BUT(INSTR-1)) IN ASPLO 4005. TIMES / 16384. DATA PATTERNS

6726:

TEST500C:

PO,

LOAD-ENVA(ZTARGET402),

!SETUP FOR D =0 TEST

LOAD-ERROR(TEST500C),

!ERROR DIRECTORY KEY

DCS-CTR(C5.),

!COMPARE AT TARGET

BUMP-VERIFY,

!COUNT

J/EXPEC500C

NEXT

(6726) DCS(1.00.1.0.0.1) BM(1010..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...001.011.110)

11106  
11107  
11108  
11109  
11110  
11111

6136: !(FREE)

EXPEC500C:

P3,

CSPD(EXPEC)+EMIT,

!EXPECTED COUNT =

EMIT/7645,

(7645) = 4005.

J/COMP500C

NEXT

(6136) DCS(0.00.0.0.0.0) BM(0000..10.11..11.10..100..101...0.0.0..0..0...0.0000...1..0000.0...11.000...001.011.111)

11112  
11113  
11114  
11115  
11116  
11117  
11118

6137: !(FREE)

COMP500C:

PO,

BUMP-VERIFY,

!COUNT

P3-T,

D+ASPLO(OVERLAP)-MINUS-CSPB(EXPEC),

!COMPARE RECEIVED:EXPECTED

SAVE-D(C),

J/GOBUT500C

NEXT

(6137) DCS(0.00.0.0.0.1) BM(1101..11.00..10.00..000..111...1.1.0..0..0...0.0000...0..0000.0...11.000...001.100.000)

11119  
11120  
11121  
11122  
11123  
11124

6140: !(FREE)

GOBUT500C:

SETUP,

RETURN/TEST500D,

!RETURN TO START OF NEXT SUBTEST

NEXT,

GOTO-PAGE(7),

!BUT TABLE

J/BUTD-IS-ZERO

!CHECK EQUALITY

(6140) DCS(0.00.0.0.0.0) BM(0110..00.11..11.00..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

11125  
11126  
11127  
11128  
11129

! - - - - -

11130  
11131  
11132  
11133  
11134  
11135  
11136

!\*\*\* TEST 500D \*\*\*  
!CHECK THAT ASSERTING "OVERLAP L" ALLOWED THE SAME NUMBER OF SP REWRITES  
!TO ASPLO (CHECKED ABOVE) AS TO BSPLO (CHECKED HERE)

6746:

TEST500D:

```

11137      PO,      LOAD-ENUA(ZTARGET402),      !SETUP FOR D=0 TEST
11138      LOAD-ERROR(TEST5000),      !ERROR DIRECTORY KEY
11139      DCS-CTR(C4.),      !COMPARE AT TARGET
11140      NEXT      J/COMPS000
(6746) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...001.100.001)

11141      6141:  !(FREE)
11142      COMPS000:
11143      P3-T,      D+ASPLO(OVERLAP)-MINUS-BSPLO(OVERLAP),      !COMPARE A-COUNT:B-COUNT
11144      SAVE-D(C)
11145      NEXT      J/GOBUTS000
11146      (6141) DCS(10.00.0.0.0.0) BM(1101..00.10..10.00..000..111...1.1.0..0..0...0.0000...0..0000.0...11.000...001.100.010)

11147      6142:  !(FREE)
11148      GOBUTS000:
11149      PO,      BUMP-VERIFY,      !COUNT
11150      SETUP,      RETURN/TEST500E,      !RETURN TO START OF NEXT SUBTEST
11151      NEXT,      GOTO-PAGE(7),      !BUT TABLE
11152      J/BUTD-IS-ZERO      !CHECK EQUALITY
11153      (6142) DCS(10.00.0.0.0.0) BM(0110..00.11..10.11..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

11154
11155
11156
11157
11158
11159      ! - - - - -
11160
11161      !*** TEST 500E ***
11162      !CHECK THAT "PREFETCH(0)H" WAS ONLY ASSERTED 357. TIMES/16384. DATA PATTERNS
11163      6736:
11164      TEST500E:
11165      PO,      LOAD-ENUA(ZTARGET402),      !SETUP FOR D=0 TEST
11166      LOAD-ERROR(TEST500E),      !ERROR DIRECTORY KEY
11167      DCS-CTR(C5.),      !COMPARE AT TARGET
11168      NEXT      J/EXPEC500E
(6736) DCS(1.00.1.0.0.0) BM(1010..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...001.100.011)

11169      6143:  !(FREE)
11170      EXPEC500E:
11171      PO,      BUMP-VERIFY,      !
11172      P3,      CSPD(EXPEC)+EMIT,      !EXPECTED COUNT=
11173      EMIT/545,      ! (545)=357.
11174      NEXT      J/COMPS000
11175      (6143) DCS(0.00.0.0.0.0) BM(0000..10.00..01.01..100..101...0.0.0..0..0...0.0000...1..0000.0...11.000...001.100.100)

11176      6144:  !(FREE)
11177      COMPS000:
11178      P3-T,      D+ASPHI(PREFETCH)-MINUS-CSPB(EXPEC),      !COMPARE RECEIVED:EXPECTED
11179      SAVE-D(C)
11180      NEXT      J/GOBUTS000
11181      (6144) DCS(0.00.0.0.0.0) BM(1101..11.00..11.00..000..111...1.1.0..0..0...0.0000...0..0000.0...11.000...001.100.101)

11182      6145:  !(FREE)
11183

```

```

11184 GOBUT500E:
11185     SETUP, RETURN/SCOPE500F,      !RETURN TO SCOPE LOOP TEST WORD
11186     NEXT,  GOTO-PAGE(7),          !BUT TABLE
11187     J/BUTD-15-ZERO                !CHECK EQUALITY
(6145) DCS(0.00.0.0.0.0) BM(0110..00.00..11.00..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.0011

```

```

11188
11189
11190
11191 6146: !(FREE)
11192 SCOPE500F:
11193     PO,      BUMP-VERIFY,          !COUNT
11194             BUSDIN+EMIT-[I],      !RESET PROC UCON
11195             EN-CLK-IR[15-00],
11196     NEXT,    BUTD[SCOPE],          !NO ERROR: "TEST503A" (+1. WORDS)
11197     J/TEST503A                    !ERROR: "SETFETCHB500" (-27. WORDS)
(6146) DCS(0.00.0.1.0.1) BM(0000..00.00..00.01..000..100...0.0.0..0..0...1.1001...0..0000.0...11.000...110.111.1111

```

```

11198
11199
11200
11201
11202
11203 !.PAGE=====
11204

```

```

11205 .TOC * TEST503-510: PROCESSOR UCON TESTS (FLAGS, FPS, PS, BUTM) & ASSOC LOGIC

```

```

11206 !*****
11207 !*
11208 !* TESTS: 503A - 510F UWORDS: 214 + 245
11209 !*
11210 !* FUNCTIONS: TESTS 503A - 510F MANIPULATE THE VARIOUS PROCESSOR UCON
11211 !* FUNCTIONS (FLAGS, EXFLAGS, FPS, PS, CUA) AND RELATED "BUT"
11212 !* TESTS TO SEE THAT ALL ARE FUNCTIONAL.
11213 !*
11214 !*
11215 !*
11216 !*****

```

```

11217 ! - - - - -
11218

```

```

11219 !*** TEST 503 ***

```

```

11220 !TESTS 503 A-I USE DATA PATTERNS OF:
11221 !   FLAG<8:4,2:0>H = "10101010", EXFLAG<2:1>H = "01",
11222 !   FPS<7:0>H = "1010 1010"

```

```

11223 ! - - - - -

```

```

11224 !*** TEST 503A ***

```

```

11235 !LOAD FLAGS<8:4,2:0> WITH "10101010", EXFLAGS<2:1> WITH "01", FPS<7:4> WITH "1010",
11236 !FPS<3:0> WITH "1010", AND READ BACK THRU "FLAGS#FPS" PORT OF PROCESSOR MUX
11237 6677:
11238 TEST503A:
11239     PO,      LOAD-ENVA(LOADNZWS),      !COMPARE 1/2 WAY THRU BM SUBR
11240           LOAD-ERROR(TEST503A),      !ERROR DIRECTORY KEY
11241           DCS-CTR(C10.),              !COMPARE IN ...
11242           BUMP-VERIFY,                 !COUNT
11243     NEXT,    J/LOADFLAG503A
(6677) DCS(1.00.1.0.0.1) BM(0101..00.10..00.11..011..010...0.0.0..0..0...0.0000...0..0000.0...11.000...110.001.100)

11244 6614:
11245 LOADFLAG503A:
11246     P2-U,    IR+EMIT,                  ! (125122)=NOT-FLTPT-H INSTR (NOTE UCON/CSPADDR OVLP)
11247           !ALSO NOTE IR=(NOT-PREFETCH), SO PREFETCH-SAVE WILL GET "0"
11248           !AFTER SUBSEQUENT JAMUPP (IN LOADFPSCC ROUTINE EXIT)
11249           P3,    CSPD(05)+EMIT,        !INITIAL VALUE:
11250           EMIT/125122,                 !FLAG<8:4,2:0>="10101010", EXFLAG<2:1>="01"
11251     NEXT,    J/LOADFPSS503A           !"10101010 0101 0010"
(6614) DCS(0.00.0.0.0.0) BM(1010..10.10..10.01..010..010...0.0.0..0..0...1.1010...1..0000.0...11.000...001.100.111)

11253 6147: !(FREE)
11254 LOADFPSS503A:
11255     PO,      BUMP-VERIFY,              !COUNT
11256     P3,      CSPD(06)+EMIT,            !INITIAL VALUE IN FPS<7:4>:
11257           EMIT/052655,                 !"01010101 1010 1101"
11258     NEXT,    J/EXPEC503A
(6147) DCS(0.00.0.0.0.1) BM(0101..10.01..01.10..101..101...0.0.0..0..0...0.1001...1..0000.0...11.000...001.101.000)

11260 6150: !(FREE)
11261 EXPEC503A:
11262     !*** NOTE: FLAG<8>, UBREAK ENABLE, GETS SET HERE. KEEP SR=(000000) [SET IN TEST500] TO FLAG
11263     ! THAT ANY SPURIOUS UBREAKS ARE ILLEGAL (IE, CAUSING A JAMUPP SEQUENCE ***
11264     P3,      CSPD(02)+EMIT,            !EXPECTED VALUE LOADS FLAGS, FPS<7:0>
11265           EMIT/125252,                 !OF "FLAGS#FPS" PORT
11266     NEXT,    J/MASK503A                !"10101010 1010 1010"
(6150) DCS(0.00.0.0.0.0) BM(1010..10.10..10.10..101..010...0.0.0..0..0...0.1101...1..0000.0...11.000...001.101.001)

11268 6151: !(FREE)
11269 MASK503A:
11270     P3,      CSPD(04)+EMIT,            !MASK OUT (TO ZEROS)
11271           EMIT/177777,                 !LOOK AT ALL THE BITS
11272     NEXT,    J/LOADFCC503A
(6151) DCS(0.00.0.0.0.0) BM(1111..10.11..11.11..111..111...0.0.0..0..0...0.1011...1..0000.0...11.000...001.101.010)

11274 6152: !(FREE)
11275 LOADFCC503A:
11276     P3,      CSPD(15)+EMIT,            !FPS<3:0> COME FROM CSP(15)<3:0>[MD]
11277           EMIT/000012,                 !"1010"
11278     NEXT,    J/DOFCC503A
(6152) DCS(0.00.0.0.0.0) BM(0000..10.00..00.00..001..010...0.0.0..0..0...0.0010...1..0000.0...11.000...001.101.011)

11280 6153: !(FREE)
11281 DOFCC503A:
11282

```



KD11-K MICRO V00A-1 00:00:03 12-MAR-77

```

11283 P3, CSPD(00)+EMIT, !CALL BM SUBR WHICH DOES THE LOAD
11284 RETURN/TEST503A, !RETURN INLINE
11285 NEXT, CALL(LoadFPSCC)
(6153) DCS(0.00.0.0.0.0) BM(0110..10.11..10.01..100..111...0.0.0..0..0...0.1111...1..0000.0...11.100...010.110.100)

```

```

11286 ! - - - - -
11287
11288 !NOW CHECK ALL THE RITE BITS WERE SET BY READING THEM BACK
11289
11290

```

```

11291 6714:
11292 TEST503A:
11293 PO, LOAD-ENUA(ZTARGET402), !SETUP FOR D=ZERO COMPARE
11294 LOAD-ERROR(TEST503A), !ERROR DIRECTORY KEY
11295 DCS-CTR(C14.) !COMPARE AT TARGET
11296 P3, BUTA(CUA-TRACK), !RESET CUA TRACKING AFTER JAMUPP
11297 NEXT, J/GOPUT503A
(6714) DCS(1.00.1.0.0.0) BM(0001..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.001...001.101.100)

```

```

11298 6154: !(FREE)
11299 GOPUT503A:
11300 SETUP, RETURN/GOBUT503A, !GO TO SUBR WHICH:
11301 NEXT, CALL(FLAGFPSSEQL00) !1) CSP(05) -> FLAGS, EXFLAGS
11302 (6154) DCS(0.00.0.0.0.0) BM(0110..00.00..11.01..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.111.011)
!2) CSP(06) -> FPS<7:4>

```

```

11303 6155: !(FREE)
11304 GOBUT503A:
11305 SETUP, RETURN/TEST503B, !RETURN TO START OF NEXT SUBTEST
11306 NEXT, CALL(FLAGFPS100) !FLAGS#FPS.XOR.CSP(02) -> D, BUT(D=0)
11307 (6155) DCS(0.00.0.0.0.0) BM(0110..00.11..00.11..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.010)
11308

```

```

11309 ! - - - - -
11310
11311
11312
11313 !
11314

```

```

11315 !*** TEST 5038 ***
11316 !DO THE "MULTIPLE BUT" ON "FLAG7-H" TO CHECK IT'S CLEAR
11317 6630:
11318 TEST5038:
11319 PO, LOAD-ENUA(ZTARGET406), !BIT<00> CLEAR
11320 LOAD-ERROR(TEST5038), !ERROR DIRECTORY KEY
11321 DCS-CTR(C4.) !COMPARE AT TARGET
11322 NEXT, J/GOBUT5038
(6630) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..000..110...0.0.0..0..0...0.0000...0..0000.0...11.000...001.101.110)

```

```

11323 6156: !(FREE)
11324 GOBUT5038:
11325 SETUP, RETURN/TEST503C, !RETURN TO START OF NEXT SUBTEST
11326 NEXT, GOTO-PAGE(7), !BUT TABLE
11327 J/BUTMFLAG7 !FLAG 7-H IN BIT<00>
11328

```

(6156) DCS(0.00.0.0.0.0) BM(0110..00.11..00.11..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.010.1011

11329  
11330  
11331  
11332  
11333  
11334  
11335  
11336  
11337  
11338  
11339  
11340  
11341  
11342  
11343

!\*\*\* TEST 503C \*\*\*  
!DO THE "BUT" ON "FLTPT-PROC-H"="FLAGS-H\*EXFLAG1-L" TO CHECK IT'S CLEAR

6632:  
TEST503C:  
PO, LOAD-ENUA(ZTARGET402), !BIT<00> CLEAR  
LOAD-ERROR(TEST503C), !ERROR DIRECTORY KEY  
DCS-CTR(C3.), !COMPARE AT TARGET  
BUMP-VERIFY, !COUNT  
NEXT, J/GOBUT503C

(6632) DCS(1.00.1.0.0.1) BM(1100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...001.101.1111

11344  
11345  
11346  
11347  
11348  
11349

6157: !(FREE)  
GOBUT503C:  
SETUP, RETURN/TEST5030A, !RETURN TO START OF NEXT SUBTEST  
NEXT, GOTO-PAGE(7), !BUT TABLE  
J/BUTFPFPROC !FLAGS-H\*EXFLAG1-L IN BIT<00>

(6157) DCS(0.00.0.0.0.0) BM(0110..00.11..00.00..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.101.1001

11350  
11351  
11352  
11353  
11354  
11355  
11356  
11357  
11358  
11359  
11360  
11361  
11362  
11363

!\*\*\* TEST 5030A \*\*\*  
!DO THE "MULTIPLE BUT" ON "FLTPT" TO CHECK IT'S CLEAR, IR=(125122), NOT-FLTPT-H INSTR

6601:  
TEST5030A:  
PO, LOAD-ENUA(ZTARGET406), !BIT<00> CLEAR  
LOAD-ERROR(TEST5030A), !ERROR DIRECTORY KEY  
DCS-CTR(C4.), !COMPARE AT TARGET  
NEXT, J/GOBUT5030A

(6601) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..000..110...0.0.0..0..0...0.0000...0..0000.0...11.000...001.110.0001

11364  
11365  
11366  
11367  
11368  
11369

6160: !(FREE)  
GOBUT5030A:  
SETUP, RETURN/TEST5030, !RETURN TO START OF NEXT SUBTEST  
NEXT, GOTO-PAGE(7), !BUT TABLE  
J/BUTMFLTPS !FLTPT-H IN BIT<00>

(6160) DCS(0.00.0.0.0.0) BM(0110..00.11..01.11..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.011.0011

11370  
11371  
11372  
11373  
11374

```

11375
11376 !*** TEST 5030 ***
11377 !DO AN INSTR-1 FLOATING POINT DECODE, TO CHECK THAT FLAG<4:5> -> BIT<1:0>
11378 6670:
11379 TEST5030:
11380     PO,      LOAD-ENUA(ZTARGET476),      !INSTR-1 FLIPT, BIT<1:0>="10"
11381           LOAD-ERROR(TEST5030),      !ERROR DIRECTORY KEY
11382           DCS-CTR(C4.),              !COMPARE AT TARGET
11383     NEXT    J/LOADIRS030
(6670) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..111..110...0.0.0..0..0...0.0000...0..0000.0...11.000...001.110.001)

11384
11385 6161: !(FREE)
11386 LOADIRS030:
11387     PO,      BUMP-VERIFY,              !COUNT
11388     P2-U,    IR-EMIT,                  !IR <- FLTPT INSTR
11389           EMIT/175252,
11390     NEXT    J/GOBUTS030
(6161) DCS(0.00.0.0.0.1) BM(1111..00.10..10.10..101..010...0.0.0..0..0...1.1010...0..0000.0...11.000...001.110.010)

11391
11392 6162: !(FREE)
11393 GOBUTS030:
11394     SETUP,  RETURN/TEST503E,          !RETURN TO START OF NEXT SUBTEST
11395     NEXT,   GOTO-PAGE(7),             !BUT TABLE
11396           J/BUTINSTR1,               !FULL WIDTH
(6162) DCS(0.00.0.0.0.0) BM(0110..00.11..10.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110)

11397
11398
11399
11400
11401 ! - - - - -
11402
11403 !*** TEST 503E ***
11404 !READ EXFLAGS<2:1> = "01" IN CUA-PORT<2:1>,
11405 !ALSO "PREFETCH-SAVE-H" = "0" FROM PREVIOUS SETUP
11406 6700:
11407 TEST503E:
11408     PO,      LOAD-ENUA(ZTARGET402),      !SETUP FOR D=0 TEST
11409           LOAD-ERROR(TEST503E),      !ERROR DIRECTORY KEY
11410           DCS-CTR(C10.),             !COMPARE AT TARGET
11411     NEXT    J/EXPEC503E
(6700) DCS(1.00.1.0.0.0) BM(0101..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...001.110.011)

11412
11413 6163: !(FREE)
11414 EXPEC503E:
11415     PO,      BUMP-VERIFY,              !COUNT
11416     P3,      CSPD(02)+EMIT,           !CUA PORT READS AS:
11417           EMIT/073732,               !0#CUA<11:00>#EXFLAG<2:1>#PREFETCH
11418     NEXT    J/GOPUTS03E
(6163) DCS(0.00.0.0.0.1) BM(0111..10.01..11.11..011..010...0.0.0..0..0...0.1101...1..0000.0...11.000...001.110.100)

11419
11420 6164: !(FREE)
11421 GOPUTS03E:
11422     SETUP,  RETURN/TEST503F,          !GO TO SUBR WHICH:

```

11423 NEXT, CALL(CUAT00) !PUTS CUA#EXFLAGS.XOR.CSP(02) -> D, BUT(D=ZERO)  
 (6164) DCS(0.00.0.0.0.0) BM(0110..00.11..10.01..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.111)

11424  
11425  
11426  
11427  
11428  
11429  
11430  
11431  
11432  
11433  
11434  
11435  
11436  
11437

!\*\*\* TEST 503F \*\*\*  
 !DO THE "MULTIPLE BUT" ON EXFLAG<2> TO CHECK IT'S CLEAR

6710:  
 TEST503F:  
 PD, LOAD-ENVA(ZTARGET406), !BIT<00> CLEAR  
 LOAD-ERROR(TEST503F), !ERROR DIRECTORY KEY  
 DCS-CTR(C4.), !COMPARE AT TARGET  
 J/GOBUT503F

(6710) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..000..110...0.0.0..0..0...0.0000...0..0000.0...11.00C...001.110.101)

6165: !(FREE)  
 GOBUT503F:  
 SETUP, RETURN/TEST503G, !RETURN TO START OF NEXT SUBTEST  
 NEXT, GOTO-PAGE(7), !BUT TABLE  
 J/BUTMEXFLAG2 !EXFLAG<2>H IN BIT<00>

(6165) DCS(0.00.0.0.0.0) BM(0110..00.11..10.10..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.011.011)

11444  
11445  
11446  
11447  
11448  
11449  
11450  
11451  
11452  
11453  
11454  
11455  
11456  
11457  
11458

!\*\*\* TEST 503G \*\*\*  
 !DO THE "MULTIPLE BUT" ON EXFLAG<1> TO CHECK IT'S SET

6720:  
 TEST503G:  
 PD, LOAD-ENVA(ZTARGET407), !BIT<00> SET  
 LOAD-ERROR(TEST503G), !ERROR DIRECTORY KEY  
 DCS-CTR(C4.), !COMPARE AT TARGET  
 BUMP-VERIFY, !COUNT  
 J/GOBUT503G

(6720) DCS(1.00.1.0.0.1) BM(1011..00.11..11.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.000...001.110.110)

6166: !(FREE)  
 GOBUT503G:  
 SETUP, RETURN/TEST503H, !RETURN TO START OF NEXT SUBTEST  
 NEXT, GOTO-PAGE(7), !BUT TABLE  
 J/BUTMEXFLAG1 !EXFLAG<1>H IN BIT<00>

(6166) DCS(0.00.0.0.0.0) BM(0110..00.11..10.11..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.010.111)

11465  
11466  
11467  
11468

```

11469 ! - - - - -
11470
11471 !*** TEST 503H ***
11472 !CHECK FPS(5) SET, VIA BUTR
11473 6732:
11474 TEST503H:
11475     PO,      LOAD-ENUA(ZTARGET407),      !BIT<00> SET
11476           LOAD-ERROR(TEST503H),        !ERROR DIRECTORY KEY
11477           DCS-CTR(C3.),                 !COMPARE AT TARGET
11478           BUMP-VERIFY,                   !COUNT
11479           NEXT, J/GOBUT503H
(6732) DCS(1.00.1.0.0.1) BM(1100..00.11..11.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.000...001.110.1111)

```

```

11460 6167: !(FREE)
11481 GOBUT503H:
11482     SETUP,   RETURN/TEST503I,           !RETURN TO START OF NEXT SUBTEST
11483     NEXT,    GOTO-PAGE(7),             !BUT TABLE
11484           J/BUTFPS05                   !FPS <05> IN BIT<00>
11485 (6167) DCS(0.00.0.0.0.0) BM(0110..00.11..11.00..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.0101)

```

```

11486
11487
11488
11489
11490 ! - - - - -
11491

```

```

11492 !*** TEST 503I ***
11493 !CHECK FLTPT-FD-H = F(FPS<7:6>,FLAG<5,2,1,EX1>) IS CLEAR
11494 6742:
11495 TEST503I:
11496     PO,      LOAD-ENUA(ZTARGET402),      !BIT<00> CLEAR
11497           LOAD-ERROR(TEST503I),        !ERROR DIRECTORY KEY
11498           DCS-CTR(C3.),                 !COMPARE AT TARGET
11499           NEXT, J/GOBUT503I
(6742) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...001.111.0001)

```

```

11500 6170: !(FREE)
11501 GOBUT503I:
11502     PO,      BUMP-VERIFY,                !COUNT
11503     SETUP,   RETURN/TEST503J,           !RETURN TO START OF NEXT SUBTEST
11504     NEXT,    GOTO-PAGE(7),             !BUT TABLE
11505           J/BUTFPFD                     !FLTPT-FD-H IN BIT<00>
11506 (6170) DCS(0.00.0.0.0.1) BM(0110..00.11..11.01..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.111.0011)

```

```

11507 ! - - - - -
11508

```

```

11509 !*** TEST 503J ***
11510 !DO THE "MULTIPLE BUT" ON "D<00>" TO CHECK IT'S CLEAR (D WAS ZEROED IN TEST503E, ABOVE)
11511 6756:
11512 TEST503J:
11513     PO,      LOAD-ENUA(ZTARGET406),      !BIT<00> CLEAR
11514           LOAD-ERROR(TEST503J),        !ERROR DIRECTORY KEY
11515           DCS-CTR(C4.),                 !COMPARE AT TARGET
11516

```

```

11517      NEXT,      J/GOBUT503J      !
(6756) DCS(1.00.1.0.0.0) BM(1011.00.11.11.00.000.110...0.0.0.0.0.0...0.0000...0..0000.0...11.000...001.111.001)
11518
11519      6171:      !(FREE)
11520      GOBUT503J:
11521      SETUP,      RETURN/TEST503K,      !RETURN TO START OF NEXT SUBTEST
11522      NEXT,      GOTO-PAGE(7),      !BUT TABLE
11523      J/BUTM000      !D<00>-H IN BIT<00>
(6171) DCS(0.00.0.0.0.0) BM(0110.00.11.01.00.110.111...0.0.0.0.0.0...0.0000...0..0000.0...11.100...011.010.001)
11524
11525
11526      ! - - - - -
11527
11528      !*** TEST 503K ***
11529      !DO THE "MULTIPLE BUT" ON "FLTPT" TO CHECK IT'S SET, IR=(175252), FLTPT-H INSTR
11530      6646:
11531      TEST503K:
11532      PO,      LOAD-ENVA(ZTARGET407),      !BIT<00> SET
11533      LOAD-ERROR(TEST503K),      !ERROR DIRECTORY KEY
11534      DCS-CTR(C4.),      !COMPARE AT TARGET
11535      BUMP-VERIFY,      !COUNT
11536      NEXT      J/GOBUT503K
(6646) DCS(1.00.1.0.0.1) BM(1011.00.11.11.00.000.111...0.0.0.0.0.0...0.0000...0..0000.0...11.000...001.111.010)
11537
11538      6172:      !(FREE)
11539      GOBUT503K:
11540      SETUP,      RETURN/SCOPE503,      !RETURN TO SCOPE LOOP TEST WORD
11541      NEXT,      GOTO-PAGE(7),      !BUT TABLE
11542      J/BUTFLTPTS      !FLTPT-H IN BIT<00>
(6172) DCS(0.00.0.0.0.0) BM(0110.00.00.11.11.011.111...0.0.0.0.0.0...0.0000...0..0000.0...11.100...011.011.001)
11543
11544
11545
11546
11547
11548
11549      6173:      !(FREE)
11550      SCOPE503:
11551      PO,      BUSDIN+EMIT-(I),      !RESET PROC UCONS
11552      EN-CLK-IR(15-00),
11553      NEXT,      BUTD(SCOPE),      !NO ERROR: "TEST504A" (+1. WORDS)
11554      J/TEST504A      ! ERROR: "LOADFLAG503A" (-34. WORDS)
(6173) DCS(0.00.0.1.0.0) BM(0000.00.00.00.01.000.100...0.0.0.0.0.0...1.1001...0..0000.0...11.000...110.001.101)
11555
11556
11557
11558
11559
11560
11561
11562
11563

```

```

11564 ! - - - - -
11565
11566 !*** TEST 504 ***
11567
11568 !TESTS 504 A-I USE DATA PATTERNS OF:
11569 ! FLAG<8:4,2:0>H = "01010101", EXFLAG<2:1>H = "10",
11570 ! FPS<7:0>H = "0101 0101"
11571 ! - - - - -
11572
11573
11574 !*** TEST 504A ***
11575 !LOAD FLAGS<8:4, 2:0> WITH "01010101", EXFLAGS<2:1> WITH "10", FPS<7:4> WITH "0101",
11576 !FPS<3:0> WITH "0101", AND READ BACK THRU "FLAGS#FPS" PORT OF PROCESSOR MUX
11577 6615:
11578 TEST504A:
11579     P3,      LOAD-ENUA(4777),      !SETUP FOR COMPARE 1/2 WAY + 1 THRU BM SUBR
11580             LOAD-ERROR(TEST504A), !ERROR DIRECTORY KEY
11581             DCS-CTR(C11.)         !COMPARE AT ...
11582     NEXT     J/LOADFLAG504A
(6615) DCS(1.00.1.0.0.0) BM(0100..00.10..01.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.000...110.010.010)

11583
11584 6622:
11585 LOADFLAG504A:
11586     P2-U,    IR+EMIT,             !ALSO NOTE IR=(NOT-PREFETCH), SO PREFETCH-SAVE WILL GET "0"
11587             !AFTER SUBSEQUENT JAMUPP (IN LOADFPSCC ROUTINE EXIT)
11588     P3,      CSPD(05)+EMIT,       !INITIAL VALUE:
11589             EMIT/052644,         !FLAG<8:4,2:0>="01010101", EXFLAG<2:1>="10"
11590             NEXT     J/LOADFPS504A !"01010101 1010 0100"
(6622) DCS(0.00.0.0.0.0) BM(0101..10.01..01.10..100..100...0.0.0..0..0...1.1010...1..0000.0...11.000...001.111.100)

11591
11592 6174: !(FREE)
11593 LOADFPS504A:
11594     P3,      CSPD(06)+EMIT.       !INITIAL VALUE IN FPS<7:4>:
11595             EMIT/125133,         !"10101010 0101 1011"
11596     NEXT     J/EXPEC504A
(6174) DCS(0.00.0.0.0.0) BM(1010..10.10..10.01..011..011...0.0.0..0..0...0.1001...1..0000.0...11.000...001.111.101)

11597
11598 6175: !(FREE)
11599 EXPEC504A:
11600     P3,      CSPD(02)+EMIT,       !EXPECTED VALUE LOADS FLAGS, FPS<7:0>
11601             EMIT/052525,         !OF "FLAGS#FPS" PORT
11602     NEXT     J/LOADFCC504A       !"01010101 0101 0101"
(6175) DCS(0.00.0.0.0.0) BM(0101..10.01..01.01..010..101...0.0.0..0..0...0.1101...1..0000.0...11.000...001.111.110)

11603
11604 6176: !(FREE)
11605 LOADFCC504A:
11606     P3,      CSPD(15)+EMIT,       !FPS<3:0> COME FROM CSP(15)<3:0>[MD]
11607             EMIT/000005,         !"0101"
11608     NEXT     J/DOFCC504A
(6176) DCS(0.00.0.0.0.0) BM(0000..10.00..00.00..000..101...0.0.0..0..0...0.0010...1..0000.0...11.000...001.111.111)

11609
11610 6177: !(FREE)
11611 DOFCC504A:
11612     P3,      CSPD(00)+EMIT,       !CALL BM SUBR WHICH DOES THE LOAD

```

```

11613 RETURN/TEST504A, ;RETURN INLINE
11614 NEXT, CALL[LOADFPSCC]
(6177) DCS(0.00.0.0.0.0) BM(0110..10.11..11.10..110..111...0.0.0..0..0...0.1111...1..0000.0...11.100...010.110.100)
11615 ! - - - - -
11616 !
11617 !
11618 !NOW CHECK ALL THE RITE BITS WERE SET BY READING THEM BACK
11619 !
11620 6766:
11621 TEST504A:
11622 PO, LOAD-ENVA(ZTARGET402), ;SETUP FOR D=ZERO TEST
11623 LOAD-ERROR(TEST504A), ;ERROR DIRECTORY KEY
11624 DCS-CTR(C14.), ;COMPARE AT TARGET
11625 BUMP-VERIFY, ;COUNT
11626 P3, BUTA(CUA-TRACK), ;RESET CUA TRACKING AFTER JAMUPP
11627 NEXT, J/GOPUT504A
(6766) DCS(1.00.1.0.0.1) BM(0001..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.001...010.000.000)
11628 6200: !(FREE)
11629 GOPUT504A:
11631 SETUP, RETURN/GOBUT504A, ;GO TO SUBR WHICH:
11632 NEXT, CALL[FLAGFPSSEQL00] ;1) CSP(05) -> FLAGS, EXFLAGS
(6200) DCS(0.00.0.0.0.0) BM(0110..00.01..00.00..001...111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.111.011)
11633 ;2) CSP(06) -> FPS(7:4)
11634 6201: !(FREE)
11635 GOBUT504A:
11637 SETUP, RETURN/TEST504B, ;RETURN TO START OF NEXT SUBTEST
11638 NEXT, CALL[FLAGFPSST00] ; FLAGS#FPS.XOR.CSP(02) -> D, BUT(D=0)
(6201) DCS(0.00.0.0.0.0) BM(0110..00.11..10.11..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.010)
11639 ! - - - - -
11640 !
11641 !
11642 !
11643 !
11644 !*** TEST 504B ***
11645 !DO THE "MULTIPLE BUT" ON "FLAG7-H" TO CHECK IT'S SET
11646 6730:
11647 TEST504B:
11648 PO, LOAD-ENVA(ZTARGET407), ;BIT<00> SET
11649 LOAD-ERROR(TEST504B), ;ERROR DIRECTORY KEY
11650 DCS-CTR(C4.), ;COMPARE AT TARGET
11651 NEXT, J/GOBUT504B
(6730) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.000...010.000.010)
11652 6202: !(FREE)
11653 GOBUT504B:
11654 SETUP, RETURN/TEST504C, ;RETURN TO START OF NEXT SUBTEST
11655 NEXT, GOTO-PAGE(7), ;BUT TABLE
11656 ;FLAG7-H IN BIT<00>
11657

```



(6202) DCS(0.00.0.0.0.0) BM(0110..00.11..11.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.010.1011)

11658  
11659  
11660  
11661  
11662  
11663  
11664  
11665  
11666  
11667  
11668  
11669  
11670  
11671  
11672

!\*\*\* TEST 504C \*\*\*  
!DO THE "BUT" ON "FLTPT-PROC-H"="FLAG<5>-H\*EXFLAG<1>-L" TO CHECK IT'S SET

6740:  
TEST504C:  
PO, LOAD-ENUA(ZTARGET403), !BIT<00> SET  
LOAD-ERROR(TEST504C), !ERROR DIRECTORY KEY  
DCS-CTR(C3.), !COMPARE AT TARGET  
BUMP-VERIFY, !COUNT  
NEXT, J/GOBUT504C

(6740) DCS(1.00.1.0.0.1) BM(1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...010.000.0111)

11673  
11674  
11675  
11676  
11677  
11678

6203: !(FREE)  
GOBUT504C:  
SETUP, RETURN/TEST504D, !RETURN TO START OF NEXT SUBTEST  
NEXT, GOTO-PAGE(7), !BUT TABLE  
J/BUTFPFPROC !FLAGS-H\*EXFLAG1-L IN BIT<00>

(6203) DCS(0.00.0.0.0.0) BM(0110..00.11..11.01..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.101.1001)

11679  
11680  
11681  
11682  
11683  
11684  
11685  
11686  
11687  
11688  
11689  
11690  
11691  
11692

!\*\*\* TEST 504D \*\*\*  
!DO AN INSTR-1 FLOATING POINT DECODE, TO CHECK THAT FLAG<4:5> -> BIT<1:0>

6750:  
TEST504D:  
PO, LOAD-ENUA(ZTARGET475), !INSTR-1 FLTPT, BIT<1:0>="01"  
LOAD-ERROR(TEST504D), !ERROR DIRECTORY KEY  
DCS-CTR(C4.), !COMPARE AT TARGET  
NEXT, J/LOADIR504D

(6750) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..111..101...0.0.0..0..0...0.0000...0..0000.0...11.000...010.000.1001)

11693  
11694  
11695  
11696  
11697  
11698  
11699

6204: !(FREE)  
LOADIR504D:  
PO, BUMP-VERIFY, !COUNT  
P2-U, IR+EMIT, !IR <- FLTPT INSTR  
EMIT/172525,  
NEXT, J/GOBUT504D

(6204) DCS(0.00.0.0.0.1) BM(1111..00.01..01.01..010..101...0.0.0..0..0...1.1010...0..0000.0...11.000...010.000.1011)

11700  
11701  
11702  
11703  
11704  
11705

6205: !(FREE)  
GOBUT504D:  
SETUP, RETURN/TEST504E, !RETURN TO START OF NEXT SUBTEST  
NEXT, GOTO-PAGE(7), !BUT TABLE  
J/BUTINSTR1 !FULL WIDTH

```

11706 (6205) DCS(0.00.0.0.0.0) BM(0110..00.11..11.10..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.110)
11707
11708
11709
11710 ! - - - - -
11711
11712 !*** TEST 504E ***
11713 !READ EXFLAGS<2:1>="10" IN CUA-PORT<2:1>
11714 !ALSO "PREFETCH-SAVE-H"="0" FROM PREVIOUS SETUP
11715 6760:
11716 TEST504E:
11717     PO,          LOAD-ENVA(ZTARGET402),          !SETUP FOR D=0 TEST
11718                LOAD-ERROR(TEST504E),          !ERROR DIRECTORY KEY
11719                DCS-CTR(C10.),          !COMPARE AT TARGET
11720     NEXT,        J/EXPEC504E
11721 (6760) DCS(1.00.1.0.0.0) BM(0101..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...010.000.110)
11722 6206: !(FREE)
11723 EXPEC504E:
11724     PO,          BUMP-VERIFY          !COUNT
11725     P3,          CSPD(02)+EMIT,        !CUA PORT READS AS:
11726                EMIT/073734,          !D#CUA<11:00>#EXFLAG<2:1>#PREFETCH
11727     NEXT,        J/GOPUT504E
11728 (6206) DCS(0.00.0.0.0.1) BM(0111..10.01..11.11..011..100...0.0.0..0..0...0.1101...1..0000.0...11.000...010.000.111)
11729 6207: !(FREE)
11730 GOPUT504E:
11731     SETUP,       RETURN/TEST504F,      !GO TO SUBR WHICH:
11732     NEXT,        CALL(CUATOD)          !PUTS CUA#EXFLAGS.XOR.CSP(02) -> D, BUT(D=ZERO)
11733 (6207) DCS(0.00.0.0.0.0) BM(0110..00.11..01.11..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.111)
11734
11735
11736
11737 ! - - - - -
11738
11739 !*** TEST 504F ***
11740 !DO THE "MULTIPLE BUT" ON EXFLAG<2> TO CHECK IT'S SET
11741 6672:
11742 TEST504F:
11743     PO,          LOAD-ENVA(ZTARGET407),          !BIT<00> SET
11744                LOAD-ERROR(TEST504F),          !ERROR DIRECTORY KEY
11745                DCS-CTR(C4.),          !COMPARE AT TARGET
11746     NEXT,        J/GOBUT504F
11747 (6672) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.000...010.001.000)
11748 6210: !(FREE)
11749 GOBUT504F:
11750     SETUP,       RETURN/TEST504G,      !RETURN TO START OF NEXT SUBTEST
11751     NEXT,        GOTO-PAGE(7)          !BUT TABLE
11752                J/BUTMEXFLAG2          !EXFLAG<2>H IN BIT 0

```

(6210) DCS(0.00.0.0.0.0) BM(0110..00.11..10.00..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.011.011)

11753  
11754  
11755  
11756  
11757  
11758  
11759  
11760  
11761  
11762  
11763  
11764  
11765  
11766  
11767

! - - - - -

!\*\*\* TEST 504G \*\*\*  
!DO THE "MULTIPLE BUT" ON EXFLAG<1> TO CHECK IT'S CLEAR

6702:

TEST504G:

PO, LOAD-ENUA(ZTARGET406), !BIT<00> CLEAR  
LOAD-ERROR(TEST504G), !ERROR DIRECTORY KEY  
DCS-CTR(C4.), !COMPARE AT TARGET  
BUMP-VERIFY, !COUNT  
J/GOBUT504G

(6702) DCS(1.00.1.0.0.1) BM(1011..00.11..11.00..000..110...0.0.0..0..0...0.0000...0..0000.0...11.000...010.001.001)

11768  
11769  
11770  
11771  
11772  
11773

6211: !(FREE)

GOBUT504G:

SETUP, RETURN/TEST504H, !RETURN TO START OF NEXT SUBTEST  
NEXT, GOTO-PAGE(7), !BUT TABLE  
J/BUTEXFLAG1 !EXFLAG<1>H IN BIT 0

(6211) DCS(0.00.0.0.0.0) BM(0110..00.11..10.00..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.010.111)

11774  
11775  
11776  
11777  
11778

! - - - - -

!\*\*\* TEST 504H \*\*\*  
!CHECK FPS<5> CLEAR, VIA BUTR

6706:

TEST504H:

PO, LOAD-ENUA(ZTARGET406), !BIT<00> CLEAR  
LOAD-ERROR(TEST504H), !ERROR DIRECTORY KEY  
DCS-CTR(C3.), !COMPARE AT TARGET  
BUMP-VERIFY, !COUNT  
J/GOBUT504H

(6706) DCS(1.00.1.0.0.1) BM(1100..00.11..11.00..000..110...0.0.0..0..0...0.0000...0..0000.0...11.000...010.001.010)

11789  
11790  
11791  
11792  
11793  
11794

6212: !(FREE)

GOBUT504H:

SETUP, RETURN/TEST504I, !RETURN TO START OF NEXT SUBTEST  
NEXT, GOTO-PAGE(7), !BUT TABLE  
J/BUTFPS05 !FPS<05> IN BIT<00>

(6212) DCS(0.00.0.0.0.0) BM(0110..00.11..10.01..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.010)

11795  
11796  
11797  
11798

```

11799 ! - - - - -
11800
11801 !*** TEST 504I ***
11802 !CHECK FLTPT-FD-H = F(FPS<7:6>,FLAG<5,2,1,EX1>) IS SET
11803 6716:
11804 TEST504I:
11805     PO,      LOAD-ENUA(ZTARGET403),      !BIT<00> SET
11806           LOAD-ERROR(TEST504I),      !ERROR DIRECTORY KEY
11807           DCS-CTR(C3.),      !COMPARE AT TARGET
11808     NEXT,    J/GOBUT504I
(6716) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...010.001.011)

11809 5213: !(FREE)
11810 GOBUT504I:
11811     PO,      BUMP-VERIFY,      !COUNT
11812           SETUP, RETURN/SCOPE504,      !RETURN TO SCOPE LOOP TEST WORD
11813           NEXT, GOTO-PAGE(7),      !BUT TABLE
11814           J/BUTFPFD,      !FLTPT-FD-H IN BIT<00>
11815 (6213) DCS(0.00.0.0.0.0.1) BM(0110..00.01..00.01..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.111.001)

11816
11817
11818 6214: !(FREE)
11819 SCOPE504:
11820     PO,      BUSDIN+EMIT-(1),      !RESET PROC UCON
11821           EN-CLK-IR(15-00),
11822     NEXT,    BUTD(SCOPE),      !NO ERROR: "TEST505A" (+1. WORDS)
11823           J/TEST505A,      !ERROR: "LOADFLAG504A" (-27. WORDS)
11824 (6214) DCS(0.00.0.1.0.0) BM(0000..00.00..00.01..000..100...0.0.0..0..0...1.1001...0..0000.0...11.000...110.010.011)

11825
11826
11827
11828
11829
11830
11831
11832
11833
11834 ! - - - - -
11835
11836 !*** TEST 505 ***
11837
11838 !TESTS 505 A-C USE DATA PATTERNS OF:
11839 !   FLAG<8:4,2:0>H = "11111000", EXFLAG<2:1>H = "10",
11840 !   FPS<7:0>H = "0101 0101"
11841
11842 ! - - - - -
11843
11844 !*** TEST 505A ***
11845 !CHECK THAT BUTA(CLEAR-FLAG-RES-UCON) ONLY CLEARS SHORT-TERM FLAGS
11846 6623:
11847 TEST505A:
11848     PO,      LOAD-ENUA(ZTARGET402),      !SERUP FOR D=ZERO TEST

```

```

11849          LOAD-ERROR(TEST505A),          !ERROR DIRECTORY KEY
11850          DCS-CTR(C15.),                  !COMPARE AT TARGET
11851          NEXT,                            !
(6623) DCS[1.00.1.0.0.0] BM[0000..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...110.010.000]

11852
11853          6620:
11854          LOAD505A:
11855          P3,          CSPD[05]+EMIT,          !LOAD FLAGS<8:4,2:0>
11856          EMIT/177406,          !EXFLAG<2:1> WITH ALL 1-5
11857          NEXT,                            !
(6620) DCS[0.00.0.0.0.0] BM[1111..10.11..11.00..000..110...0.0.0..0..0...0.1010...1..0000.0...11.000...010.001.101]

11858
11859          6215: !(FREE)
11860          EXPECS05A:
11861          PO,          BUMP-VERIFY,          !COUNT
11862          P3,          CSPD[02]+EMIT,          !AFTER BUTA(CLR-FLAG-...), EXPECT THIS IN FLAGS#FPS PORT:
11863          EMIT/174125,          !"11111000 0101 0101"
11864          NEXT,                            !
(6215) DCS[0.00.0.0.0.1] BM[1111..10.10..00.01..010..101...0.0.0..0..0...0.1101...1..0000.0...11.000...010.001.110]

11865
11866          6216: !(FREE)
11867          SETFLAG505A:
11868          PO,          BUMP-VERIFY,          !COUNT
11869          SETUP,       RETURN/BUTCLRS05A,      !GO TO SUBR WHICH:
11870          NEXT,       CALL[FLAGL00]           !PUTS CSP(05) INTO THE FLAGS
(6216) DCS[0.00.0.0.0.1] BM[0110..00.01..00.01..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...000.111.110]

11871
11872          6217: !(FREE)
11873          BUTCLRS05A:
11874          PO,          DCS-CTR(C9.),          !POINT COUNTER AT TARGET
11875          P3,          BUTA(CLR-FLAG-RES-UCON), !CLEAR SHORT-TERM FLAGS
11876          NEXT,                            !
(6217) DCS[0.00.1.0.0.0] BM[0110..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.010...010.010.000]

11877
11878          6220: !(FREE)
11879          GOPUT505A:
11880          SETUP,       RETURN/TEST505B,        !GO TO SUBR WHICH:
11881          NEXT,       CALL[FLAGFPST00]        !FLAGS#FPS.XOR.CSP(2) -> 0, BUT(D=ZERO)
(6220) DCS[0.00.0.0.0.0] BM[0110..00.11..10.01..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.010]

11882
11883
11884
11885
11886          ! - - - - -
11887
11888          !*** TEST 505B ***
11889          !CHECK EXFLAG<2> NOT CLEARED, VIA BUTM
11890          6712:
11891          TEST505B:
11892          PO,          LOAD-ENUA(ZTARGET407),  !BIT<00> SET
11893          LOAD-ERROR(TEST505B),          !ERROR DIRECTORY KEY
11894          DCS-CTR(C4.),                  !COMPARE AT TARGET

```

```

11895          BUMP-VERIFY,          !COUNT
11896      NEXT,  J/GOBUT505B          !
(6712) DCS[1.00.1.0.0.1] BM[1011..00.11..11.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.000...010.010.001]
11897
11898      6221:  !(FREE)
11899      GOBUT505B:
11900          SETUP,  RETURN/TEST505C,  !RETURN TO START OF NEXT SUBTEST
11901          NEXT,   GOTO-PAGE(7)      !BUT TABLE
11902          J/BUTMEXFLAG2            !EXFLAG<2> IN BIT<00>
(6221) DCS[0.00.0.0.0.0] BM[0110..00.11..10.10..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.011.011]
11903
11904
11905
11906
11907      ! - - - - -
11908
11909      !*** TEST 505C ***
11910      !CHECK EXFLAG<1> WAS CLEARED, VIA BUTM
11911      6722:
11912      TEST505C:
11913          PO,      LOAD-ENUA(ZTARGET406),  !BIT<00> CLEAR
11914          LOAD-ERROR(TEST505C),          !ERROR DIRECTORY KEY
11915          DCS-CTR(C4.),                  !COMPARE AT TARGET
11916      NEXT,  J/GOBUT505C          !
(6722) DCS[1.00.1.0.0.0] BM[1011..00.11..11.00..000..110...0.0.0..0..0...0.0000...0..0000.0...11.000...010.010.011]
11917
11918      6223:  !(FREE)
11919      GOBUT505C:
11920          SETUP,  RETURN/SCOPE505,  !RETURN TO SCOPE LOOP TEST WORD
11921          NEXT,   GOTO-PAGE(7)      !BUT TABLE
11922          J/BUTMEXFLAG1            !EXFLAG<1> IN BIT<00>
(6223) DCS[0.00.0.0.0.0] BM[0110..00.01..00.10..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.010.111]
11923
11924      6224:  !(FREE)
11925      SCOPE505:
11926          PO,      BUMP-VERIFY,          !COUNT
11927          BUSDIN+EMIT-[I],            !EMIT FOR CONSTANTS
11928          P3,      FLAG[8-0]+D[15-8]-[I], !ZERO ALL FLAGS. 0 WAS LEFT
11929          FPS[7-4]+D[7-4]-[I],       !ZERO FROM TEST505A, IF ALL OK.
11930          NEXT,   BUTD[SCOPE],        !NO ERROR: "TEST506A" (+1. WORDS)
11931          J/TEST506A                  !ERROR: "LOAD505A" (-9. WORDS)
(6224) DCS[0.00.0.1.0.1] BM[0000..00.00..00.01..100..001...0.0.0..0..0...1.1011...0..0000.0...11.000...110.010.001]
11932
11933
11934
11935
11936
11937      ! - - - - -
11938
11939
11940      !*** TEST 506 ***
11941

```

```

11942 ! TESTS 506 A-E USE A "0-1-1010-1010" PATTERN IN PS<15,13,7:4,3:0>
11943 ! - - - - -
11944 !
11945 !
11946 ! * TEST 506A *
11947 ! LOAD UP PS<HI,MID,LO> IN ORDER, READ BACK THRU PS PORT OF PROC MUX
11948 ! ALSO CHECK THAT BUTA(CLR-FLAG-RES-UCON) CLEARS UCON REGISTER, SO THAT:
11949 ! (1) BUSDIN+EMIT IS SELECTED, VIA UCON-SELECT(1)H=LOW, AND
11950 ! (2) MAKE SURE THAT THE OTHER UCON BIT LATCHES ARE ALSO CLEARED
11951 6621:
11952 TEST506A:
11953     PO,      LOAD-ENUA(ZTARGET402),      ! SETUP FOR D=ZERO TEST
11954     P3,      LOAD-ERROR(TEST506A),      ! ERROR DIRECTORY KEY
11955     P3,      DCS-CTR(C15.),             ! STALL FOR NOW
11956     NEXT     J/EXPEC506A
(6621) DCS(1.00.1.0.0.0) BM(0000..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...110.001.010)

11957 6612:
11958 EXPEC506A:
11959     P3,      CSPD(02)+EMIT,             ! EXPECTED VALUE TO BE READ OUT OF
11960     P3,      EMIT/030252,             ! PS AFTER LOADING:
11961     P3,      NEXT     J/SETUCON506A     ! "0011 0000 1010 1010"
11962 (6612) DCS(0.00.0.0.0.0) BM(0011..10.00..00.10..101..010...0.0.0..0..0...0.1101...1..0000.0...11.000...010.010.101)

11963 6225: !(FREE)
11964 SETUCON506A:
11965     PO,      BUSDIN+FLAGS#FPS-[I],     ! TAKE EMIT OFF BUSDIN, FLAG#FPS=(000005)
11966     P3,      BUTA(CLR-FLAG-RES-UCON), ! THIS SHOULD NOW CLEAR THE UCON REGISTER,
11967     P3,      NEXT     J/PSHI506A       ! SETTING UCON-SELECT(1)H=L; FORCING BUSDIN+EMIT
11968 (6225) DCS(0.00.0.0.0.0) BM(0000..00.00..11.01..000..000...0.0.0..0..0...1.1001...0..0000.0...11.010...010.010.110)

11970 6226: !(FREE)
11971 PSHI506A:
11972     PO,      BUMP-VERIFY,             ! COUNT
11973     P3,      CSPD(05)+EMIT,           ! VALUE IN D WHEN LOAD PS<15,13>:
11974     P3,      EMIT/063125,           ! "0110 0110 0101 0101"
11975     P3,      NEXT     J/PSMID506A
11976 (6226) DCS(0.00.0.0.0.1) BM(0110..10.01..10.01..010..101...0.0.0..0..0...0.1010...1..0000.0...11.000...010.010.111)

11977 6227: !(FREE)
11978 PSMID506A:
11979     P3,      CSPD(06)+EMIT,           ! VALUE IN D WHEN LOAD PS<7:4>:
11980     P3,      EMIT/143245,           ! "1100 0110 1010 0101"
11981     P3,      NEXT     J/PSLO506A
11982 (6227) DCS(0.00.0.0.0.0) BM(1100..10.01..10.10..100..101...0.0.0..0..0...0.1001...1..0000.0...11.000...010.011.000)

11983 6230: !(FREE)
11984 PSLO506A:
11985     PO,      BUMP-VERIFY,             ! COUNT
11986     P3,      CSPD(07)+EMIT,           ! VALUE IN D WHEN LOAD PS<3:0>:
11987     P3,      EMIT/143132,           ! "1100 0110 0101 1010"
11988     P3,      NEXT     J/FUDGEPS506A
11989 (6230) DCS(0.00.0.0.0.1) BM(1100..10.01..10.01..011..010...0.0.0..0..0...0.1000...1..0000.0...11.000...010.011.001)

```

KD11-K MICRO V00A-1 00:00:03 12-MAR-77

```

11990
11991 6231: !(FREE)
11992 FUDGEPS506A:
11993     SETUP, RETURN/LOADUCON506A,           !GO TO SUBR WHICH:
11994                                           !1) CSP(05) -> PS<15,13>
11995                                           !2) CSP(06) -> PS<7,4>
11996                                           !3) CSP(07) -> PS<3,0>
11997     NEXT, CALL[PSSEQLOD]
(6231) DCS[0.00.0.0.0.0] BM[0111..00.00..01.10..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.000.000]
11998
11999 7067: !(FREE)
12000 LOADUCON506A:
12001     SETUP UCON-PROC,                       !SETUP FOR UCON-PROC:
12002     ENABLE, BUSDIN+FLAGS#FPS-[1],          !FLAGS#FPS=(000005)
12003           EN-CLK-PS[15-12], EN-CLK-PS[7-4], !SETUP FOR CLOCKING PS BITS
12004           EN-CLK-PS[3-0],
12005     PO, SET-UCON-CONTROL,                   !WRITE THE UCON REGISTER AT PO,
12006     P3, BUTA(CLR-FLAG-RES-UCON),          !BUT AT P3, NOW CLEAR IT OUT
12007     NEXT, J/DOIT506A
(7067) DCS[0.00.0.0.0.0] BM[1000..00.00..11.01..010..010...0.0.0..0..0...1.1001...0..0000.0...11.010...000.111.010]
12008
12009 7072: !(FREE)
12010 DOIT506A:
12011     P2-T, D+ZERO,                          !SETUP D WITH A (000000)
12012     NEXT, GOTO-PAGE(6),                    !XFER
12013           J/LOADDCS506A
(7072) DCS[0.00.0.0.0.0] BM[0011..00.00..00.00..000..110...0.1.0..0..0...0.0000...0..0000.0...11.100...001.000.100]
12014
12015 6104: !(FREE)
12016 LOADDCS506A:
12017     PO, DCS-CTR(C9.),                      !COMPARE AT TARGET
12018     P2, UCON-OPERATION,                   !IF THE UCON REGISTER WASN'T CLEARED, ABOVE,
12019     NEXT, GOTO-PAGE(7),                    !THE PS SHOULD BE OVER-WRITTEN WITH (000000),
12020           J/GOBUT506A                      !IN SOME COMBINATION OF PS<15:12,7:4,3:0>
(6104) DCS[0.00.1.0.0.0] BM[0110..00.00..00.00..000..111...0.0.0..0..0...1.1010...0..0000.0...11.100...000.111.001]
12021
12022 7071: !(FREE)
12023 GOBUT506A:
12024     SETUP, RETURN/TEST506B,                !RETURN TO START OF NEXT SUBTEST
12025     NEXT, CALL[PSTOD],                     !PS-XOR-CSP(02) -> D, BUT(D=ZERO)
(7071) DCS[0.00.0.0.0.0] BM[0110..00.11..11.01..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.011]
12026
12027
12028
12029 ! - - - - -
12030
12031 !* TEST 506B *
12032 !CHECK THAT BUTR(PS15) SHOWS PS<15>H CLEAR
12033 6754:
12034 TEST506B:
12035     PO, LOAD-ENUA(ZTARGET402),             !BIT<00> CLEAR

```



12036 LOAD-ERROR(TEST506B), !ERROR DIRECTORY KEY  
 12037 DCS-CTR(C3.), !COMPARE AT TARGET  
 12038 (6754) NEXT J/GOBUT506B !  
 (6754) DCS[1.00.1.0.0.0] BM[1100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...010.011.011]

12039 6233: !(FREE)  
 12040 GOBUT506B:  
 12041 SETUP, RETURN/TEST506C, !RETURN TO START OF NEXT SUBTEST  
 12042 NEXT, GOTO-PAGE(7), !BUT TABLE HERE  
 12043 J/BUTPS15 !PS<15>H IN BIT<00>  
 12044 (6233) DCS[0.00.0.0.0.0] BM[0110..00.11..11.00..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.011]

12045  
 12046  
 12047  
 12048  
 12049 ! - - - - -  
 12050

12051 !\* TEST 506C \*  
 12052 !CHECK THAT BUTM(PS03) SHOWS PS<03>H SET  
 12053 6744:  
 12054 TEST506C:  
 12055 PO, LOAD-ENUA(ZTARGET407), !BIT<00> SET  
 12056 LOAD-ERROR(TEST506C), !ERROR DIRECTORY KEY  
 12057 DCS-CTR(C4.), !COMPARE AT TARGET  
 12058 BUMP-VERIFY, !COUNT  
 12059 (6744) NEXT J/GOBUT506C !  
 (6744) DCS[1.00.1.0.0.1] BM[1011..00.11..11.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.000...010.011.100]

12060 6234: !(FREE)  
 12061 GOBUT506C:  
 12062 SETUP, RETURN/TEST506D, !RETURN TO START OF NEXT SUBTEST  
 12063 NEXT, GOTO-PAGE(7), !BUT TABLE HERE  
 12064 J/BUTMPS[N] !PS[N]=PS<03>H IN BIT<00>  
 12065 (6234) DCS[0.00.0.0.0.0] BM[0110..00.11..10.11..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.010.011]

12066  
 12067  
 12068  
 12069  
 12070 ! - - - - -  
 12071

12072 !\* TEST 506D \*  
 12073 !CHECK THAT WHEN FLAG<0>H=L AND PS<04>H=L,  
 12074 ! MASKED-PS[T]-H=FLAG<0>L\*PS<04>H, IS LOW  
 12075 6734:  
 12076 TEST506D:  
 12077 PO, LOAD-ENUA(ZTARGET406), !BIT<00> CLEAR  
 12078 LOAD-ERROR(TEST506D), !ERROR DIRECTORY KEY  
 12079 DCS-CTR(C4.), !COMPARE AT TARGET  
 12080 (6734) NEXT J/GOBUT506D !  
 (6734) DCS[1.00.1.0.0.0] BM[1011..00.11..11.00..000..110...0.0.0..0..0...0.0000...0..0000.0...11.000...010.011.101]

12081 6235: !(FREE)  
 12082

```

12083 GOBUT506D:
12084 PO BUMP-VERIFY !COUNT
12085 SETUP, RETURN/TEST506E, !RETURN TO START OF NEXT SUBTEST
12086 NEXT, GOTO-PAGE(7) !BUT TABLE HERE
12087 J/BUTMASKPS(T) !FLAG<0>L*PS<04>H IN BIT<00>
(6235) DCS(0.00.0.0.0.1) BM(0110..00.11..00.00..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.111)

```

```

12088
12089
12090
12091
12092 ! - - - - -
12093

```

```

12094 !* TEST 506E *
12095 !CHECK THAT INTR-HIGH-H=SERVICE(0)H*NOT(MASKED-PS(T)-H) IS HIGH,
12096 ! WHEN SERVICE(0)H=HIGH, MASKED-PS(T)-H=LOW
12097 !NOTE: THE SUBR CALLED SHOULD SET SERVICE(0)H=HIGH (NEGATED) BY CLEARING OUT ALL ITS INPUT CONDITIONS
12098 6605:
12099 TEST506E:
12100 PO, LOAD-EMUA(ZTARGET403), !BIT<01> SET
12101 LOAD-ERROR(TEST506E), !ERROR DIRECTORY KEY
12102 DCS-CTR(C9.), !COMPARE AT TARGET
12103 NEXT, J/CLEAR506E
(6605) DCS(1.00.1.0.0.0) BM(0110..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...010.011.110)

```

```

12104
12105 6236: !(FREE)
12106 CLEAR506E:
12107 PO BUMP-VERIFY !COUNT
12108 SETUP, RETURN/GOBUT506E, !GO TO SUBR WHICH CLEARS OUT
12109 NEXT, CALL(CLEAR-I-0-B) ! ALL THE UNIBUS/I-O/SERVICE LATCHES (NO UNIBUS INIT)
(6236) DCS(0.00.0.0.0.1) BM(0110..00.01..00.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.000)

```

```

12110
12111 6237: !(FREE)
12112 GOBUT506E:
12113 SETUP, RETURN/SCOPE506, !RETURN TO SCOPE LOOP TEST WORD
12114 NEXT, GOTO-PAGE(7), !BUT TABLE HERE
12115 J/BUTINTRHIGH !INTR HIGH H IN BIT<01>
(6237) DCS(0.00.0.0.0.0) BM(0110..00.01..01.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.101.101)

```

```

12116
12117
12118
12119
12120 6240: !(FREE)
12121 SCOPE506:
12122 PO, BUSDIN+EMIT-[[I], !KEEP EMIT FOR CONSTANTS
12123 P2, PS+D-[[I], !ZERO PS; D LEFT ZERO FROM
12124 !PREVIOUS TESTS IF ALL OK
12125 NEXT, BUTD(SCOPE), !NO ERROR: "TEST507A" (+1. WORDS)
12126 J/TEST507A ! ERROR: "MASK506A" (-16. WORDS)
(6240) DCS(0.00.0.1.0.0) BM(1000..00.00..00.01..010..010...0.0.0..0..0...1.1011...0..0000.0...11.000...110.001.011)

```

```

12127
12128

```

```

12129
12130
12131
12132
12133 !-----
12134
12135 !*** TEST 507 ***
12136
12137 !TESTS 507 A-F USE A "1-0-0101-0101" PATTERN IN PS<15,13,7:4,3:0>
12138 !-----
12139
12140
12141 !* TEST 507A *
12142 !LOAD UP PS<HI,MID,LO> IN ORDER, READ BACK THRU PS PORT OF PROC MUX
12143 6613:
12144 TEST507A:
12145     PO,          LOAD-ENVA(ZTARGET402),          !SETUP FOR D=ZERO TEST
12146     P3,          LOAD-ERROR(TEST507A),          !ERROR DIRECTORY KEY
12147     NEXT,        DCS-CTR(C15.),                !STALL FOR NOW
12148     J/EXPEC507A
(6613) DCS[1.00.1.0.0.0] BM[0000..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...110.000.010]

12149
12150 6602:
12151 EXPEC507A:
12152     P3,          CSPD[02]+EMIT,                !EXPECTED VALUE TO BE READ OUT OF
12153     NEXT,        EMIT/140125,                !PS AFTER LOADING:
12154     J/PSH1507A  !"1100 0000 0101 0101"
(6602) DCS[0.00.0.0.0.0] BM[1100..10.00..00.01..010..101...0.0.0..0..0...0.1101...1..0000.0...11.000...010.100.001]

12155
12156 6241: !(FREE)
12157 PSH1507A:
12158     PO,          BUMP-VERIFY
12159     P3,          CSPD[05]+EMIT,                !COUNT
12160     NEXT,        EMIT/113252,                !VALUE IN D WHEN LOAD PS<15,13>:
12161     J/PSMID507A !"1001 0110 1010 1010"
(6241) DCS[0.00.0.0.0.1] BM[1001..10.01..10.10..101..010...0.0.0..0..0...0.1010...1..0000.0...11.000...010.100.010]

12162
12163 6242: !(FREE)
12164 PSMID507A:
12165     P3,          CSPD[06]+EMIT,                !VALUE IN D WHEN LOAD PS<7:4>:
12166     NEXT,        EMIT/033122,                !"0011 0110 0101 1010"
12167     J/PSL0507A
(6242) DCS[0.00.0.0.0.0] BM[0011..10.01..10.01..010..010...0.0.0..0..0...0.1001...1..0000.0...11.000...010.100.011]

12168
12169 6243: !(FREE)
12170 PSL0507A:
12171     PO,          BUMP-VERIFY
12172     P3,          CSPD[07]+EMIT,                !COUNT
12173     NEXT,        EMIT/033245,                !VALUE IN D WHEN LOAD PS<3:0>:
12174     J/FUDGEPS507A !"0011 0110 1010 0101"
(6243) DCS[0.00.0.0.0.1] BM[0011..10.01..10.10..100..101...0.0.0..0..0...0.1000...1..0000.0...11.000...010.100.100]

12175
12176 6244: !(FREE)
12177 FUDGEPS507A:

```

```

12178          SETUP, RETURN/LOADDCS507A,          !GO TO SUBR WHICH:
12179          NEXT  CALL[PSSEQ100]                !1) CSP(05) -> PS<15,13>
(6244) DCS[0.00.0.0.0.0] BM[0110..00.01..01.00..101..111] !0.0.0.0.0.0.0.0000...0..0000.0...11.100...001.000.000]
12180          !2) CSP(06) -> PS<7:4>
12181          !3) CSP(07) -> PS<3:0>
12182
12183          6245: !(FREE)
12184          LOADDCS507A:
12185          PO, DCS-CTR(C9.),                      !COMPARE AT TARGET
12186          NEXT J/GOBUT507A
(6245) DCS[0.00.1.0.0.0] BM[0110..00.00..00.00..000..000]...0.0.0..0..0...0.0000...0..0000.0...11.000...010.100.110]
12187
12188          6246: !(FREE)
12189          GOBUT507A:
12190          SETUP, RETURN/TEST507B,                !RETURN TO START OF NEXT SUBTEST
12191          NEXT  CALL[PS100]                       ! CSP(02)-XOR-PS -> 0, BUT(D=ZERO)
(6246) DCS[0.00.0.0.0.0] BM[0110..00.11..10.10..100..111]...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.011]
12192
12193
12194
12195
12196          ! - - - - -
12197
12198          !* TEST 507B *
12199          !CHECK THAT BUTR(PS15) SHOWS PS<15>H SET
12200          6724:
12201          TEST507B:
12202          PO, LOAD-ENUA(ZTARGET403),                !BIT<00> SET
12203          LOAD-ERROR(TEST507B),                    !ERROR DIRECTORY KEY
12204          DCS-CTR(C3.),                            !COMPARE AT TARGET
12205          NEXT  J/GOBUT507B
(6724) DCS[1.00.1.0.0.0] BM[1100..00.11..11.00..000..011]...0.0.0..0..0...0.0000...0..0000.0...11.000...010.100.111]
12206
12207          6247: !(FREE)
12208          GOBUT507B:
12209          SETUP, RETURN/TEST507C,                !RETURN TO START OF NEXT SUBTEST
12210          NEXT, GOTO-PAGE(7),                    !BUT TABLE HERE
12211          J/BUTPS15                                !PS<15>H IN BIT<00>
(6247) DCS[0.00.0.0.0.0] BM[0110..00.11..00.01..001..111]...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.011]
12212
12213
12214
12215
12216          ! - - - - -
12217
12218          !* TEST 507C *
12219          !CHECK THAT BUTM(PS03) SHOWS PS<03>H CLEAR
12220          6611:
12221          TEST507C:
12222          PO, LOAD-ENUA(ZTARGET406),                !BIT<00> CLEAR
12223          LOAD-ERROR(TEST507C),                    !ERROR DIRECTORY KEY

```

```

12224          DCS-CTR(C4.),          !COMPARE AT TARGET
12225          BUMP-VERIFY,          !COUNT
12226          NEXT J/GOBUT507C
(6611) DCS(1.00.1.0.0.1) BM(1011..00.11..11.00..000..110...0.0.0..0..0...0.0000...0..0000.0...11.000...010.101.000)

```

```

12227          6250: !(FREE)
12228          GOBUT507C:
12229          SETUP, RETURN/TEST507D, !RETURN TO START OF NEXT SUBTEST
12230          NEXT, GOTO-PAGE(7), !BUT TABLE HERE
12231          J/BUTMPS(N) !PS(N)=PS<03>H IN BIT<00>
12232 (6250) DCS(0.00.0.0.0.0) BM(0110..00.10..11.01..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.010.011)

```

```

12233
12234
12235
12236
12237 ! - - - - -
12238

```

```

12239 !* TEST 507D *
12240 !CHECK THAT WHEN FLAG<0>H=L AND PS<04>H=H,
12241 ! MASKED-PS(T)-H=FLAG<0>L*PS<04>H, IS HIGH
12242 6557:
12243 TEST507D:
12244     PO,          LOAD-ENUA(ZTARGET407), !BIT<00> SET
12245     LOAD-ERROR(TEST507D), !ERROR DIRECTORY KEY
12246     DCS-CTR(C4.), !COMPARE AT TARGET
12247     NEXT J/GOBUT507C
(6557) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.000...010.101.001)

```

```

12248          6251: !(FREE)
12249          GOBUT507D:
12250          PO          BUMP-VERIFY,          !COUNT
12251          SETUP, RETURN/TEST507E, !RETURN TO START OF NEXT SUBTEST
12252          NEXT, GOTO-PAGE(7) !BUT TABLE HERE
12253          J/BUTMASKPS(T) !FLAG<0>L*PS<04>H IN BIT<00>
12254 (6251) DCS(0.00.0.0.0.1) BM(0110..00.11..01.11..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.111)

```

```

12255
12256
12257 ! - - - - -
12258

```

```

12259 !* TEST 507E *
12260 !CHECK THAT INTR-HIGH-H=SERVICE(0)H*NOT(MASKED-PS(T)-H) IS LOW,
12261 ! WHEN SERVICE(0)H=HIGH, MASKED-PS(T)-H=HIGH
12262 6674:
12263 TEST507E:
12264     PO,          LOAD-ENUA(ZTARGET401), !BIT<01> CLEAR
12265     LOAD-ERROR(TEST507E), !ERROR DIRECTORY KEY
12266     DCS-CTR(C3.), !COMPARE AT TARGET
12267     BUMP-VERIFY, !COUNT
12268     NEXT J/GOBUT507C
12269 (6674) DCS(1.00.1.0.0.1) BM(1100..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...010.101.011)
12270

```

```

12271
12272 6253: !(FREE)
12273 GOBUTS07F:
12274     SETUP, RETURN/TEST507F, !RETURN TO START OF NEXT SUBTEST
12275     NEXT,  GOTO-PAGE(7), !BUT TABLE HERE
12276           J/BUTINTRHIGH !INTR HIGH H IN BIT<01>
(6253) DCS(0.00.0.0.0.0) BM(0110..00.11..01.00..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.101.101;)

```

```

12277
12278
12279
12280 ! - - - - -
12281

```

```

12282 !* TEST 507F *
12283 !CHECK THAT PS(C)-H CAN BE READ AS A "1" THRU D(C)+CINMUX=PS(C), AND D(C)+PS(C)
12284 ! IT WAS PREVIOUSLY CHECKED, IN ALU LOGIC TESTS, THAT THESE WERE OK WITH PS(C)="0"
12285 6647:
12286 TEST507F:
12287     PO,      LOAD-ENUA(ZTARGET407), !SETUP FOR SR<3:0> = "0111"
12288           LOAD-ERROR(TEST507F), !ERROR DIRECTORY KEY
12289           DCS-CTR(C7.), !COMPARE AT TARGET
12290     NEXT,    J/CINS07F
(6647) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.000...010.101.100)

```

```

12291
12292 6254: !(FREE)
12293 CINS07F:
12294     PO,      BUMP-VERIFY !COUNT
12295     P2-T,    D+ZERO, D(C)+CINMUX, !SET D(C)=CINMUX=PS(C)=(1)
12296           SR+ZERO, !SET SR<3:0>="0000"
12297     P3,      BSPL0(17)+D, !B-SIDE COPY OF SR
12298     NEXT,    J/SETPS507F
(6254) DCS(0.00.0.0.0.1) BM(0011..00.11..00.00..011..000...0.1.1..0..0...0.0000...0..0110.0...11.000...010.101.110)

```

```

12299
12300 6256: !(FREE)
12301 SETPS507F:
12302     P2-T,    D+A-PLUS-B-PLUS-D(C), D(C)+PS(C), !SET D(C)=PS(C)=(1)
12303           SR+A-PLUS-B-PLUS-D(C), !D, SR, BSP <- SR-LEFT-1, SR<02> <- PREV D(C) = CINMUX/PS(C)
12304           BUS-A+SR
12305           BUS-B+BSPL0(R17),
12306     P3,      BSPL0(17)+D, !SAVE ON B-SIDE
12307     NEXT,    J/SETSR507F
(6256) DCS(0.00.0.0.0.0) BM(0100..00.11..00.00..011..001...0.1.1..0..0...0.0000...0..0110.0...11.000...010.101.111)

```

```

12308
12309 6257: !(FREE)
12310 SETSR507F:
12311     PO,      BUMP-VERIFY !COUNT
12312     P2-T,    D+A-PLUS-B-PLUS-D(C), D(C)+ALU15, !SET D(C)=0
12313           SR+A-PLUS-B-PLUS-D(C), !D, SR, BSP <- SR-LEFT-1, SR<01> <- PREV D(C) = PS(C)
12314           BUS-A+SR
12315           BUS-B+BSPL0(R17),
12316     P3,      BSPL0(17)+D, !SAVE ON B-SIDE
12317     NEXT,    J/SETCINS07F
(6257) DCS(0.00.0.0.0.1) BM(0100..00.11..00.00..011..100...0.1.1..0..0...0.0000...0..0110.0...11.000...010.110.000)

```

```

12318

```

KD11-K MICRO V00A-1 00:00:03 12-MAR-77

```

12319 6260: !(FREE)
12320 SETCINS07F:
12321 P2-T, D+A-PLUS-B-PLUS-PS(C), D(C)+ALU15, !SET D(C)=0
12322 SR+A-PLUS-B-PLUS-PS(C), !D, SR, BSP <- SR-LEFT-1, SR<00> <- CIN/PS(C)
12323 BUS-A+SR
12324 BUS-B+BSPLO(R17),
12325 NEXT J/GOBUT507F
(6260) DCS(0.00.0.0.0.0) BM(0001..00.11..00.00..011..100...0.1.1..0..0...0.0000...0..0000.0...11.000...010.110.001)

```

```

12326 6261: !(FREE)
12327 GOBUT507F:
12328 SETUP, RETURN/SCOPE507, !RETURN TO SCOPE LOOP TEST WORD
12329 NEXT, GOTO-PAGE(7), !BUT TABLE
12330 J/BUTSR3-0 !CHECK THAT WE GOT "0111" = (07)
12331 (6261) DCS(0.00.0.0.0.0) BM(0110..00.01..01.10..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.110)

```

```

12332
12333
12334
12335
12336 6262: !(FREE)
12337 SCOPE507:
12338 P0, BUSDIN+EMIT-[I], !KEEP EMIT FOR CONSTANTS
12339 P2, PS+D-[I], !ZERO PS; 0 LEFT ZERO FROM
12340 !PREVIOUS TESTS IF ALL OK
12341 NEXT, BUTD(SCOPE), !NO ERROR: "TEST510A" (+1. WORDS)
12342 J/TEST510A !ERROR: "EXPEC507A" (-21. WORDS)
12343 (6262) DCS(0.00.0.1.0.0) BM(1000..00.00..00.01..010..010...0.0.0..0..0...1.1011...0..0000.0...11.000...110.000.011)

```

```

12344
12345
12346
12347
12348
12349
12350 !-----
12351

```

```

12352 !*** TEST 510 ***
12353 !TESTS 510 A-F USE PS<7:5>H="111", PS<4>H="1", VARIOUS FLAG<7,0>H COMBINATIONS,
12354 ! TO TEST THE INTR-HIGH-H, SERVICE-H, AND MASKED-PS(T)-H LOGIC
12355 !-----
12356

```

```

12357 !* TEST 510A *
12358 !CHECK THAT BG-SERVICE(0)H=HIGH WHEN PS<7:5>H="111" (PSW PRIORITY 7); EG, BR>PS-H=LOW, SINCE NO
12359 ! EXTERNAL UNIBUS DEVICE CAN THEN REQUEST AN INTERRUPT (IE, IT IS MASKED OUT)
12360

```

```

12361 6603:
12362 TEST510A:
12363 PO, LOAD-ENVA(ZTARGET407), !BIT<02> SET
12364 LOAD-ERROR(TEST510A), !ERROR DIRECTORY KEY
12365 DCS-CTR(C5.) !COMPARE AT TARGET
12366 NEXT, J/SETONES510A
12367 (6603) DCS(1.00.1.0.0.0) BM(1010..00.11..11.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.000...110.000.110)

```

```

12368
12369 6606:
12370 SETONES10A:
12371 P2-T, D+ALL-ONES, SAVE-D(C), !ALL ONES FOR LOAD PRIORITY 7
12372 NEXT, J/LOADPRIOS10A
(6606) DCS(0.00.0.0.0.0) BM(1111..00.00..11.01..101..111...0.1.0..0..0...0.0000...0..0000.0...11.000...010.110.011)

12373
12374 6263: !(FREE)
12375 LOADPRIOS10A:
12376 PO, BUSDIN+EMIT-(I), !KEEP IT ON
12377 P3-T, PS(7-4)+D(7-4)-(I), !PRIO 7, T-BIT (PS04) SET
12378 NEXT, J/GOBUTS10A
(6263) DCS(0.00.0.0.0.0) BM(0000..00.00..00.01..010..000...1.0.0..0..0...1.1011...0..0000.0...11.000...010.110.100)

12379
12380 6264: !(FREE)
12381 GOBUTS10A:
12382 SETUP, RETURN/TESTS10B, !RETURN TO START OF NEXT SUBTEST
12383 NEXT, GOTO-PAGE(7), !BUT TABLE
12384 J/BUTBGSEVL !BG-SERVICE(0)H IN BIT<02>
(6264) DCS(0.00.0.0.0.0) BM(0110..00.11..01.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.110)

12385
12386
12387
12388
12389
12390 ! - - - - -
12391
12392 !* TEST 510B *
12393 !CHECK THAT BUT( NOT(BG-SERVICE(0)H) + NOT(FLPT-SERVICE-L) ) IN BIT<03>
12394 ! SEES BG-SERVICE(0)H=HIGH, FLPT-SERVICE-L=HIGH, AND THUS IS NOT ASSERTED (=LOW)
12395 6675:
12396 TESTS10B:
12397 PO, LOAD-ENVA(ZTARGET407), !BIT<03> CLEAR
12398 LOAD-ERROR(TESTS10B), !ERROR DIRECTORY KEY
12399 DCS-CTR(C3.), !COMPARE AT TARGET
12400 NEXT, J/GOBUTS10B
(6675) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.000...010.110.101)

12401
12402 6265: !(FREE)
12403 GOBUTS10B:
12404 SETUP, RETURN/TESTS10C, !RETURN TO START OF NEXT SUBTEST
12405 NEXT, GOTO-PAGE(7), !BUT TABLE
12406 J/BUTBGFPSEVL !RESULT IN BIT<03>
(6265) DCS(0.00.0.0.0.0) BM(0110..00.11..00.01..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.111)

12407
12408
12409
12410
12411
12412 ! - - - - -
12413
12414 !* TEST 510C *

```



```

12415 !CHECK THAT SERVICE-H=INTR-HIGH-H*BG-SERVICE(0)H*NOT(FLAG<7>H) IS HIGH,
12416 ! WHEN INTR-HIGH-H=LOW, BG-SERVICE(0)H=HIGH, FLAG<7>H=LOW
12417 6616:
12418 TESTS100:
12419     PO,          LOAD-ENUA(ZTARGET403),          !BIT<00> SET
12420     LOAD-ERROR(TESTS100),          !ERROR DIRECTORY KEY
12421     DCS-CTR(C3.),          !COMPARE AT TARGET
12422     NEXT,        J/GOBUTS100
(6616) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...010.110.110)

12423 6266: !(FREE)
12424 GOBUTS100:
12425     SETUP,       RETURN/TESTS100,          !RETURN TO START OF NEXT SUBTEST
12426     NEXT,        GOTO-PAGE(7),          !BUT TABLE
12427     J/BUTSERVICE
12428 (6266) DCS(0.00.0.0.0.0) BM(0110..00.11..00.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.101)

12429 ! - - - - -
12430
12431
12432
12433
12434 !* TEST 5100 *
12435 !SET ONLY FLAG<0>H, FLAG<7>H, LEAVE PS<04>H SET,
12436 !CHECK MASKED-PS(T)-H=PS<04>H*FLAG<0>L IS LOW
12437 6600:
12438 TESTS100:
12439     PO,          LOAD-ENUA(ZTARGET406),          !BIT<00> CLEAR
12440     LOAD-ERROR(TESTS100),          !ERROR DIRECTORY KEY
12441     DCS-CTR(C7.),          !COMPARE AT TARGET
12442     BUMP-VERIFY,          !COUNT
12443     NEXT,        J/FLAGS100
(6600) DCS(1.00.1.0.0.1) BM(1000..00.11..11.00..000..110...0.0.0..0..0...0.0000...0..0000.0...11.000...010.110.111)

12444 6267: !(FREE)
12445 FLAGS100:
12446     P3,          CSPD(01)+EMIT,          !BITS FOR FLAG<7,0> ONLY, BIT<00> OF D
12447     EMIT/040401
12448     NEXT,        J/ENFLAGS100
(6267) DCS(0.00.0.0.0.0) BM(0100..10.00..01.00..000..001...0.0.0..0..0...0.1110...1..0000.0...11.000...010.111.000)

12450 6270: !(FREE)
12451 ENFLAGS100:
12452     P3-T,        D+CSPD(001), D(C)+0,          !INTO D
12453     NEXT,        J/SETFLAGS100
(6270) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...1.1.0..0..0...0.1110...0..0000.0...11.000...010.111.001)

12455 6271: !(FREE)
12456 SETFLAGS100:
12457     P3-T,        FLAG(8-0)+D(15-8)-(1),          !SET FLAGS<7,0>H ONLY
12458     NEXT,        J/GOBUTS100
(6271) DCS(0.00.0.0.0.0) BM(0000..00.00..00.01..000..001...1.0.0..0..0...1.1011...0..0000.0...11.000...010.111.010)

12460 6272: !(FREE)
12461

```

```

12462 GOBUTS100:
12463     SETUP, RETURN/TESTS100A,           !RETURN TO START OF NEXT SUBTEST
12464     NEXT,  GOTO-PAGE(7),              !BUT TABLE HERE
12465     J/BUTMASKPS(T)                    !FLAG<0>L*PS<04>H IN BIT<00>
(6272) DCS(0.00.0.0.0.0) BM(0110..00.11..00.01..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.111)

12466
12467
12468
12469
12470 ! - - - - -
12471
12472 !*** TEST 5100A ***
12473 !DO THE "MULTIPLE BUT" ON "D<00>H" TO CHECK IT'S SET (DURING TESTS100, ABOVE)
12474 6617:
12475 TESTS100A:
12476     PO, LOAD-ENUA(ZTARGET407),          !BIT<00> SET
12477     LOAD-ERROR(TESTS100A),            !ERROR DIRECTORY KEY
12478     DCS-CTR(C4.),                     !COMPARE AT TARGET
12479     BUMP-VERIFY,                       !COUNT
12480     NEXT, J/GOBUTS100A
(6617) DCS(1.00.1.0.0.1) BM(1011..00.11..11.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.000...010.111.011)

12481
12482 6273: !(FREE)
12483 GOBUTS100A:
12484     SETUP, RETURN/TESTS100E,           !RETURN TO START OF NEXT SUBTEST
12485     NEXT,  GOTO-PAGE(7),              !BUT TABLE
12486     J/BUTM000                          !D<00>-H IN BIT<00>
(6273) DCS(0.00.0.0.0.0) BM(0110..00.11..00.00..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.010.001)

12487
12488
12489
12490
12491 ! - - - - -
12492
12493 !* TEST 510E *
12494 !CHECK THAT SERVICE-H=INTR-HIGH-H*BG-SERVICE(0)H*NOT(FLAG<7>H) IS HIGH,
12495 ! WHEN INTR-HIGH-H=HIGH, BG-SERVICE(0)H=HIGH, FLAG<7>H=HIGH
12496 6604:
12497 TESTS10E:
12498     PO, LOAD-ENUA(ZTARGET403),          !BIT<00> SET
12499     LOAD-ERROR(TESTS10E),            !ERROR DIRECTORY KEY
12500     DCS-CTR(C4.),                     !COMPARE AT TARGET
12501     NEXT, J/ZER00S10E
(6604) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...010.111.100)

12502
12503 6274: !(FREE)
12504 ZER00S10E:
12505     P2-T, D+ZERO, D(C)+ALU15,         !ZER0ES FOR BELOW
12506     NEXT, J/GOBUTS10E
(6274) DCS(0.00.0.0.0.0) BM(0011..00.00..00.00..000..100...0.1.0..0..0...0.0000...0..0000.0...11.000...010.111.101)

12507
12508 6275: !(FREE)

```

```

12509 GOBUT510E:
12510 PO, BUMP-VERIFY, !COUNT
12511 SETUP, RETURN/TEST510F, !RETURN TO START OF NEXT SUBTEST
12512 NEXT, GOTO-PAGE(7), !BUT TABLE HERE
12513 J/BUTSERVICE !SERVICE-H IN BIT<00>
(6275) DCS(0.00.0.0.0.0) BM(0110..00.11..00.01..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.101)

12514
12515
12516
12517
12518 ! - - - - -
12519
12520 !* TEST 510F *
12521 !CHECK THAT SERVICE-H=INTR-HIGH-H*BG-SERVICE(0)H*NOT(FLAG<7>H) IS LOW,
12522 !WHEN INTR-HIGH-H=HIGH, BG-SERVICE(0)H=HIGH, FLAG<7>H=LOW
12523 6610:
12524 TEST510F:
12525 PO, LOAD-ENVA(ZTARGET432), !BIT<00> CLEAR
12526 LOAD-ERROR(TEST510F), !ERROR DIRECTORY KEY
12527 DCS-CTR(C4.) !COMPARE AT TARGET
12528 NEXT, J/ZEROFLAG510E
(6610) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...010.111.110)

12529
12530 6276: !(FREE)
12531 ZEROFLAG510E:
12532 P2, PS+D-[I], !ZERO PS-T-BIT, AND ALL THE OTHERS,
12533 FPS[7-4]+D[7-4]-[I], !ZERO THE FPS
12534 P3, FLAG[8-0]+D[15-8]-[I], !ZERO FLAGS (0 ZEROED ABOVE)
12535 NEXT, J/GOBUT510F
(6276) DCS(0.00.0.0.0.0) BM(1000..00.00..00.01..110..011...0.0.0..0..0...1.1011...0..0000.0...11.000...010.111.111)

12536
12537 6277: !(FREE)
12538 GOBUT510F:
12539 SETUP, RETURN/SCOPE510, !RETURN TO SCOPE LOOP TEST WORD
12540 NEXT, GOTO-PAGE(7), !BUT TABLE HERE
12541 J/BUTSERVICE !SERVICE-H IN BIT<00>
(6277) DCS(0.00.0.0.0.0) BM(0110..00.01..10.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.101)

12542
12543
12544
12545
12546
12547 6300: !(FREE)
12548 SCOPE510:
12549 PO, BUSOIN+EMIT-[I], !KEEP EMIT FOR CONSTANTS
12550 P2, PS+D-[I], !ZERO PS; 0 LEFT ZERO FROM
12551 !PREVIOUS TESTS IF ALL OK
12552 NEXT, BUTD(SCOPE), !NO ERROR: "TEST511A" (+11. WORDS)
12553 J/TEST511A ! ERROR: "SETONES510A" (-20. WORDS)
(6300) DCS(0.00.0.1.0.0) BM(1000..00.00..00.01..010..010...0.0.0..0..0...1.1011...0..0000.0...11.000...110.000.111)

12554
12555

```

```

12556
12557 ! - - - - -
12558
12559 |
12560 |     THIS FIRST SUBROUTINE COPIES:
12561 |         CSP(06) -> FPS<7:4>
12562 |         CSP(05) -> FLAGS<8 +,2:0>, EXFLAG<2:1>
12563 |     THEN RETURNS
12564 |
12565 | 7073:  !(FREE)
12566 | FLAGFPS00:
12567 |     P3-T,  D+CSPD(D06), D(C)+0,           !GET VALUE TO LOAD TO FPS<7:4>H
12568 |     NEXT,  J/FLAGFPS02
12569 | (7073) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...1.1.0..0..0...0.1001...0..0000.0...11.000...000.111.101)
12570 |
12571 | 7075:  !(FREE)
12572 | FLAGFPS02:
12573 |     PO,    BUSDIN+EMIT-[I],
12574 |     P3-T,  FPS[7-4]+D[7-4]-[I],           !LOAD FPS<7:4>H FROM D<7:4>H
12575 |     NEXT,  J/FLAGFPS03
12576 | (7075) DCS(0.00.0.0.0.0) BM(0000..00.00..00.01..100..000...1.0.0..0..0...1.1011...0..0000.0...11.000...000.111.110)
12577 |
12578 | 7076:  !(FREE)
12579 | FLAGFPS03:
12580 |     P3-T,  D+CSPD(D05), D(C)+0,           !GET VALUE TO LOAD TO FLAGS<8:0>H,
12581 |     NEXT,  J/FLAGFPS04                   !EXFLAG<2:1>H
12582 | (7076) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...1.1.0..0..0...0.1010...0..0000.0...11.000...000.111.111)
12583 |
12584 | 7077:  !(FREE)
12585 | FLAGFPS04:
12586 |     PO,    BUSDIN+EMIT-[I],           !KEEP IT ON
12587 |     P3-T,  FLAG[8-0]+D[15-8]-[I],       !LOAD FLAGS FROM D
12588 |     NEXT,  BUTA(RETURN),               !AND RETURN
12589 |     J/BUTERROR7
12590 | (7077) DCS(0.00.0.0.0.0) BM(0000..00.00..00.01..000..001...1.0.0..0..0...1.1011...0..0000.0...11.111...011.111.110)
12591 |
12592 | |
12593 | |     THIS SECOND SUBROUTINE COPIES:
12594 | |         CSP(05) -> PS<15:12>
12595 | |         CSP(06) -> PS<7:4>
12596 | |         CSP(07) -> PS<3:0>
12597 | |     THEN RETURNS
12598 | |
12599 | | 7100:  !(FREE)
12600 | | PSSEQLO0:
12601 | |     P3-T,  D+CSPD(D05), D(C)+0,           !GET VALUE TO LOAD PS<15:12>H
12602 | |     NEXT,  J/PSSEQLO02
12603 | | (7100) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...1.1.0..0..0...0.1010...0..0000.0...11.000...001.000.001)
12604 | |
12605 | | 7101:  !(FREE)
12606 | | PSSEQLO02:
12607 | |     PO,    BUSDIN+EMIT-[I]
12608 | |     P3-T,  PS[15-12]+D[15+13]-[I],       !LOAD PS<15-12>H FROM D<15,13>H
12609 | |     NEXT,  J/PSSEQLO03

```

```

12605 (7101) DCS(0.00.0.0.0.0) BM(0000..00.00..00.01..000..010...1.0.0..0..0...1.1011...0..0000.0...11.000...001.000.010)
12606 7102: !(FREE)
12607 PSSEQL0003:
12608 P3-T, D+CSPD(006), D(C)+0, !GET VALUE TO LOAD PS<7:4>H
12609 NEXT, J/PSSEQL0004
(7102) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...1.1.0..0..0...0.1001...0..0000.0...11.000...001.000.011)
12610 7103: !(FREE)
12611 PSSEQL0004:
12612 PD, BUSDIN+EMIT-[I],
12613 P3-T, PS[7-4]+D[7-4]-[I], !LOAD PS<7:4>H FROM D<7:4>H
12614 NEXT, J/PSSEQL0005
(7103) DCS(0.00.0.0.0.0) BM(0000..00.00..00.01..010..000...1.0.0..0..0...1.1011...0..0000.0...11.000...001.000.100)
12616 7104: !(FREE)
12617 PSSEQL0005:
12618 P3-T, D+CSPD(007), D(C)+0, !GET VALUE TO LOAD TO PS<3:0>H
12619 NEXT, J/PSSEQL0006
(7104) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...1.1.0..0..0...0.1000...0..0000.0...11.000...001.000.101)
12621 7105: !(FREE)
12622 PSSEQL0006:
12623 PD, BUSDIN+EMIT-[I], !KEEP IT ON
12624 P2-T, PS[3-0]+D[3-0]-[I], !LOAD PS<3:0>H FROM D<3:0>H
12625 NEXT, BUTA(RETURN), !AND RETURN
12626 J/BUTERROR7
(7105) DCS(0.00.0.0.0.0) BM(1000..00.00..00.01..000..000...0.0.0..0..0...1.1011...0..0000.0...11.111...011.111.110)

```

12628  
12629  
12630  
12631  
12632  
12633  
12634  
12635  
12636  
12637  
12638  
12639  
12640  
12641  
12642  
12643  
12644  
12645  
12646  
12647  
12648  
12649  
12650  
12651  
12652  
12653

!.PAGE=====

.TOC \* TEST511: MFSS LOGIC TESTS

\*\*\*\*\*

TESTS: 511 A - B UWORDS: 022 + 060

FUNCTIONS:

THE FOLLOWING TESTS VERIFY THAT THE "MF SAME STACK" LOGIC OPERATES CORRECTLY, AND THAT THE "SR6-H" DECODE IS CORRECT.

\*\*\*\*\*

SUMMARY OF "MF SAME STACK H" LOGIC TESTS

12654 MF SAME STACK H = FLAG2-H \* IR8-H \* IR7-H \* IR6-L \* (PS15-H=PS13-H)-H

TEST	PS15,13H	FLAG2H	IR8-6H/SR6H	MFSS-H	DATA	
12657	A1	0-0	1	110-1	1	002601
12658	A2	0-1	1	110-1	0	022600
12659	A3	1-1	1	110-1	1	.22601
12660	A4	1-0	1	110-1	0	.102600
12661	B1	0-0	0	110-1	0	000600
12662	B2	0-0	1	010-0	0	002200
12663	B3	0-0	1	100-0	0	002400
12664	B4	0-0	1	111-0	0	002700

12678 TESTING SUBR USED FOR ABOVE TESTS USES THE DATA AS FOLLOWS:

12679 DATA<15,13> -> PS<15,13>, DATA<10> -> FLAG<2>,  
12680 DATA<8:6> -> IR<8:6>, DATA<0> = EXPECTED MFSS-H OUTPUT

12684 6232: !(FREE)

12685 MFSS01:

12686 P2-U, IR+DBUF-[I], !DONT CARE ABOUT AFFECT, ONLY SET  
12687 P3, DBUF+D-[I], !THESE UCONS UP  
12688 NEXT, J/MFSS02

12689 (6232) DCS(0.00.0.0.0.0) BM(0100..00.00..00.01..000..100...0.0.0..0..0...1.1011...0..0000.0...11.000...011.000.010)

12691 6302: !(FREE)

12692 MFSS02:

12693 P2-T, D+CSP8(B17), D(C)+ALU00, !GET DATA INTO D, D(C)+EXPECTED MFSS-H  
12694 P2-U, IR+DBUF, !IGNORE FOR NOW  
12695 P3, DBUF+0, !GET DATA INTO DBUF, TO GO TO IR NEXT  
12696 NEXT, J/MFSS03

12697 (6302) DCS(0.00.0.0.0.0) BM(1010..11.00..00.00..000..010...0.1.0..0..0...1.1010...0..0000.0...11.000...011.000.011)

12698 6303: !(FREE)

12699 MFSS03:

12700 P0, BUMP-VERIFY, !COUNT  
12701 P2-U, IR+DBUF, !SETUP IR<8:6> FROM DATA<8:6>  
12702 P3, DBUF+0, !IGNORE FOR NOW  
12703 NEXT, J/MFSS04

12704 (6303) DCS(0.00.0.0.0.1) BM(0000..00.00..00.00..000..000...0.0.0..0..0...1.1010...0..0000.0...11.000...011.000.100)

```

12705 6304: !(FREE)
12706 MFSS04:
12707 PO, BUSDIN+EMIT-[[I], !RESET
12708 P3, FLAG[8-0]+D[15-8]-[[I], !SETUP FLAG<2> FROM DATA<10>
12709 PS[15-12]+D[15-13]-[[I], !SETUP PS<15,13> FROM D<15,13>
12710 NEXT, BUTR(D[C]-A), !IF EXPECT MFSS-H=(1), J/MFSS06
12711 J/MFSS05 !IF EXPECT MFSS-H=(0), J/MFSS05

```

(6304) DCS[0.00.0.0.0.0] BM[0000..00.00..00.01..000..011...0.0.0..0..0...1.1011...0..0000.0...01.111...101.101.010]

```

12712 !* COME HERE IF EXPECT MFSS-H=(0)
12713 6552:
12714 MFSS05:

```

```

12715 PO, LOAD-ENUA(MFSSXPECO), !SETUP FOR (0)
12716 LOAD-ERROR(MFSS05), !ERROR DIRECTORY KEY
12717 DCS-CTR(C1.), !COMPARE AT TARGET
12718 BUMP-VERIFY, !COUNT
12719 NEXT, BUTR(MFSS), !TEST MFSS: (0)=MFSSXPECO, (1)=MFSSXPEC1
12720 J/MFSSXPECO
12721

```

(6552) DCS[1.00.1.0.0.1] BM[1110..00.11..01.01..100..101...0.0.0..0..0...0.0000...0..0000.0...01.100...101.100.101]

```

12722 !* COME HERE IF EXPECT MFSS-H=(1)
12723 6553:
12724 MFSS06:

```

```

12725 P1, LOAD-ENUA(MFSSXPEC1), !SETUP FOR (0)
12726 LOAD-ERROR(MFSS06), !ERROR DIRECTORY KEY
12727 DCS-CTR(C1.), !COMPARE AT TARGET
12728 NEXT, BUTR(MFSS), !TEST MFSS: (0)=MFSSXPECO, (1)=MFSSXPEC1
12729 J/MFSSXPECO
12730

```

(6553) DCS[1.00.1.0.0.0] BM[1110..00.11..01.01..100..111...0.0.0..0..0...0.0000...0..0000.0...01.100...101.100.101]

```

12731 !* COME HERE IF MFSS-H TESTS AS A (0)
12732 6545:
12733 MFSSXPECO:

```

```

12734 PO, BUSDIN+EMIT-[[I], !RESET PROC UCON
12735 NEXT, BUTA(RETURN), !AND RETURN
12736 J/BUTERROR6 !*** COMPARE DONE HERE ***
12737

```

(6545) DCS[0.00.0.0.0.0] BM[0000..00.00..00.01..000..000...0.0.0..0..0...1.1001...0..0000.0...11.111...011.111.110]

```

12739 !* COME HERE IF MFSS-H TESTS AS A (1)
12740 6547:
12741 MFSSXPEC1:

```

```

12742 PO, BUSDIN+EMIT-[[I], !RESET PROC UCON
12743 NEXT, BUTA(RETURN), !AND RETURN
12744 J/BUTERROR6 !*** COMPARE DONE HERE ***
12745

```

(6547) DCS[0.00.0.0.0.0] BM[0000..00.00..00.01..000..000...0.0.0..0..0...1.1001...0..0000.0...11.111...011.111.110]

```

12746 ! - - - - -
12747 !
12748 ! THE TESTS ACTUALLY START HERE:
12749 !
12750 ! - - - - -
12751 !
12752 !
12753 !

```

```

12754 6607:
12755 TESTS11A:
12756 PO, LOAD-ERROR(TESTS11A), !ERROR DIRECTORY KEY
12757 P3, CSPD[17]+EMIT, EMIT/002601, !* TEST 511 A1 DATA *
12758 NEXT, J/GOTESTS11A1
(6607) DCS[1.00.0.0.0.0] BM[0000..10.01..01.10..000..001...0.0.0..0..0...0.0000...1..0000.0...11.000...101.101.000]

12759 6550:
12760 GOTESTS11A1:
12761 SETUP, RETURN/TESTS11A2, !GO DO THE TEST
12762 NEXT, CALL[MFSS-TEST]
(6550) DCS[0.00.0.0.0.0] BM[0111..00.00..01.11..100..110...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.010]

12764 7074: !(FREE)
12765 TESTS11A2:
12766 P3, CSPD[17]+EMIT, EMIT/022600, !* TEST 511 A2 DATA *
12767 NEXT, J/GOTESTS11A2
(7074) DCS[0.00.0.0.0.0] BM[0010..10.01..01.10..000..000...0.0.0..0..0...0.0000...1..0000.0...11.000...001.000.111]

12770 7107: !(FREE)
12771 GOTESTS11A2:
12772 SETUP, RETURN/TESTS11A3, !GO DO THE TEST
12773 NEXT, CALL[MFSS-TEST]
(7107) DCS[0.00.0.0.0.0] BM[0111..00.00..10.01..000..110...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.010]

12775 7110: !(FREE)
12776 TESTS11A3:
12777 P3, CSPD[17]+EMIT, EMIT/122601, !* TEST 511 A3 DATA *
12778 NEXT, J/GOTESTS11A3
(7110) DCS[0.00.0.0.0.0] BM[1010..10.01..01.10..000..001...0.0.0..0..0...0.0000...1..0000.0...11.000...001.001.001]

12781 7111: !(FREE)
12782 GOTESTS11A3:
12783 SETUP, RETURN/TESTS11A4, !GO DO THE TEST
12784 NEXT, CALL[MFSS-TEST]
(7111) DCS[0.00.0.0.0.0] BM[0111..00.00..10.01..010..110...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.010]

12786 7112: !(FREE)
12787 TESTS11A4:
12788 P3, CSPD[17]+EMIT, EMIT/102600, !* TEST 511 A4 DATA *
12789 NEXT, J/GOTESTS11A4
(7112) DCS[0.00.0.0.0.0] BM[1000..10.01..01.10..000..000...0.0.0..0..0...0.0000...1..0000.0...11.000...001.001.011]

12792 7113: !(FREE)
12793 GOTESTS11A4:
12794 SETUP, RETURN/SCOPE511A, !GO DO THE TEST
12795 NEXT, CALL[MFSS-TEST]
(7113) DCS[0.00.0.0.0.0] BM[0110..00.01..10.00..001..110...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.010]

```



```

12797
12798
12799 6301: !(FREE)
12800 SCOPE511A:
12801 P3, CSPD(17)+EMIT, EMIT/002601, !RESET DATA FOR TEST 511 A1
12802 NEXT, BUID(SCOPE), !NO ERROR: "TEST511B1" (+1. WORDS)
12803 J/TEST511B1, !ERROR: "GOTEST511A1" (-7. WORDS)
(6301) DCS(0.00.0.1.0.0) BM(0000..10.01..01.10..000..001...0.C.0..0..0..0.0000...1..0000.0...11.000...101.101.001)

12804
12805
12806
12807
12808 ! - - - - -
12809
12810
12811 6551:
12812 TEST511B1:
12813 P3, CSPD(17)+EMIT, EMIT/000600, !* TEST 511 B1 DATA *
12814 NEXT, J/GOTEST511B1
(6551) DCS(0.00.0.0.0.0) BM(0000..10.00..01.10..000..000...0.0.0..0..0..0.0000...1..0000.0...11.000...101.101.100)

12815
12816 6554:
12817 GOTEST511B1:
12818 SETUP, RETURN/TEST511B2, !GO DO THE TEST
12819 NEXT, CALL(MFSS-TEST)
(6554) DCS(0.00.0.0.0.0) BM(0111..00.00..10.00..110..110...0.0.0..0..0..0.0000...0..0000.0...11.100...010.011.010)

12820
12821
12822 7106: !(FREE)
12823 TEST511B2:
12824 P3, CSPD(17)+EMIT, EMIT/002200, !* TEST 511 B2 DATA *
12825 NEXT, J/GOTEST511B2
(7106) DCS(0.00.0.0.0.0) BM(0000..10.01..00.10..000..000...0.0.0..0..0..0.0000...1..0000.0...11.000...001.001.101)

12826
12827 7115: !(FREE)
12828 GOTEST511B2:
12829 SETUP, RETURN/TEST511B3, !GO DO THE TEST
12830 NEXT, CALL(MFSS-TEST)
(7115) DCS(0.00.0.0.0.0) BM(0111..00.00..10.01..1!0..110...0.0.0..0..0..0.0000...0..0000.0...11.100...010.011.010)

12831
12832
12833 7116: !(FREE)
12834 TEST511B3:
12835 P3, CSPD(17)+EMIT, EMIT/002400, !* TEST 511 B3 DATA *
12836 NEXT, J/GOTEST511B3
(7116) DCS(0.00.0.0.0.0) BM(0000..10.01..01.00..000..000...0.0.0..0..0..0.0000...1..0000.0...11.000...001.001.111)

12837
12838 7117: !(FREE)
12839 GOTEST511B3:
12840 SETUP, RETURN/TEST511B4, !GO DO THE TEST
12841 NEXT, CALL(MFSS-TEST)
(7117) DCS(0.00.0.0.0.0) BM(0111..00.00..10.10..000..110...0.0.0..0..0..0.0000...0..0000.0...11.100...010.011.010)

```

```

12842
12843
12844 7120: !(FREE)
12845 TESTS1184:
12846 P3, CSPD(17)+EMIT, EMIT/002700, !* TEST 511 B4 DATA *
12847 NEXT, J/GOTESTS1184
(7120) DCS(0.00.0.0.0.0) BM(0000..10.01..01..11..000..000...0.0.0..0..0...0.0000...1..0000.0...11.000...001.010.001)

```

```

12848
12849 7121: !(FREE)
12850 GOTESTS1184:
12851 SETUP, RETURN/SCOPE511B, !GO DO THE TEST
12852 NEXT, CALL(MFSS-TEST)
(7121) DCS(0.00.0.0.0.0) BM(0110..00.01..10.00..101..110...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.010)

```

```

12853
12854 6305: !(FREE)
12855 SCOPE511B:
12856 P3, CSPD(17)+EMIT, EMIT/000600, !RESET DATA FOR TEST 511 B1
12857 NEXT, BUTD(SCOPE), !NO ERROR: "SETUP512A" (+1. WORDS)
12858 J/SETUP512A !ERROR: "GOTESTS1181" (-7. WORDS)
12859 (6305) DCS(0.00.0.1.0.0) BM(0000..10.00..01.10..000..000...0.0.0..0..0...0.0000...1..0000.0...11.000...101.101.101)

```

12860  
12861  
12862  
12863  
12864  
12865  
12866  
12867  
12868  
12869  
12870  
12871  
12872  
12873  
12874  
12875  
12876  
12877  
12878  
12879  
12880  
12881  
12882  
12883  
12884  
12885  
12886  
12887  
12888  
12889  
12890  
12891

! .PAGE=====

.TOC \* TESTS12: KT SRC/DST ADDRESSING LOGIC TESTS

```

*****
!*
!* TESTS: 512 A - E UNWORDS: 044 + 062
!*
!* FUNCTIONS:
!*
!* THE FOLLOWING TEN TESTS VERIFY THE KT-SRC/DST-ADDRS ROM OUTPUT AND INPUT LINES
!* FUNCTION CORRECTLY, IN RESPECT TO NO STUCK ONE/ZERO CONDITIONS.
!*
*****

```

```

!
! KT SRC/DST LOGIC EQUATIONS: (IMPLEMENTED IN ROM)
!
! KT-SRC-ADDRS-3 = NOT-F2.AND.SR6.AND.PS15
! .OR. F2.AND.SR6.AND.NOT-SM0.AND.PS15
! .OR. F2.AND.SR6.AND.SM0.AND.PS13.AND.NOT-FLTPT
! .OR. SR6.AND.PS15.AND.FLTFT
!
! KT-DST-ADDRS-3 = NOT-F1.AND.DR6.AND.PS.5

```

```

12892 .OR. F1.AND.DR6.AND.NOT-DMO.AND.PS15
12893 .OR. F1.AND.DR6.AND.DMO.AND.PS13.AND.NOT-FLTPT
12894 .OR. DR6.AND.PS15.AND.FLTPT
12895
12896
12897
12898
12899
12900
12901
12902
12903
12904
12905
12906
12907
12908
12909
12910
12911
12912
12913
12914
12915
12916
12917
12918
12919
12920
12921
12922
12923
12924
12925
12926
12927
12928
12929
12930
12931
12932
12933
12934
12935
12936
12937
12938
12939
12940
12941
12942
12943

```

SUMMARY OF KT ASP/BSP SRC/DST STACK POINTER ADDRESSING LOGIC:

TEST NUMB	PS 15:13H	FLAG 2:1H	IR	FLTL	SMDH	SR6H	DMOH	DR6H	KT-SRC ADR-H	KT-DST ADR-H	HOW READ:
A1	1,0	1,1	172206	0	0	0	1	1	0	1	ASPHI[SF]=(02) BSPHI[DF]=(16)
A2											
B1	0,1	1,0	070606	1	1	1	1	1	1	0	ASPHI[DF]=(06) BSPHI[SF]=(16)
B2											
C1	0,1	1,1	134606	1	0	1	1	1	0	1	ASPHI[DF]=(16) BSPHI[SF]=(06)
C2											
D1	1,0	0,1	160612	1	1	1	0	0	1	0	ASPHI[SF]=(16) BSPHI[DF]=(02)
D2											
E1	1,0	1,1	150626	1	1	1	0	1	0	1	ASPHI[DF]=(16) BSPHI[SF]=(06)
E2											

KT SRC/DST STACK POINTER ADDRESS MODE TEST SUBROUTINE:

```

ENTER WITH:  CSP(17) = VALUE TO GO INTO IR, TO SETUP FLTPT/SMD/SR6/DMO/DR6
              CSP(16) = BIT<15,13> -> PS<15,13>
              BIT<10:09> -> FLAG<2:1>
              ** BIT<00> IS AN INTERNAL FLAG TO INDICATE WHICH REGISTER
              TO PUT IN THE SR ON EXIT:
              BIT<00> = (1) -> ASPHI[DF], BIT<00> = (0) -> ASPHI[SF]

```

7114: !(FREE)

KTSRCDST01:

```

P2-T, D+CSPD(D17), D(C)+0, ;INITIAL DATA TO GO INTO IR
NEXT, J/KTSRCDST02

```

(7114) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...0.1.0..0..0...0.0000...0..0000.0...11.000...001.010.011)

7123: !(FREE)

KTSRCDST02:

```

P2-U, IR+DBUF-[I], ;IGNORE FOR NOW
P3, DBUF+D-[I], ;COPY IR DATA FROM D -> DBUF
NEXT, J/KTSRCDST03

```

(7123) DCS(0.00.0.0.0.0) BM(0100..00.00..00.01..000..100...0.0.0..0..0...1.1011...0..0000.0...11.000...001.010.100)

7124: !(FREE)

```

12944 KTSRCDST03:
12945 P2-U, IR+DBUF ;COPY IR DATA FROM DBUF -> IR
12946 P3-T, D+CSPB(816), D(C)+ALU00, ;GET PS/FLAGS/REGISTER SELECT DATA
12947 P3, DBUF+0 ;D(C) = REGISTER SELECT FLAG
12948 NEXT, J/KTSRCDST04
(7124) DCS(0.00.0.0.0.0) BM(1010..11.01..00.00..000..010...1.1.0..0..0...1.1010...0..0000.0...11.000...001.010.101)

12949 7125: !(FREE)
12950 KTSRCDST04:
12951 P0, BUSDIN+EMIT-[I], ;KEEP IT ON
12952 P3, PS(15-12)+D(15#13)-[I], ;SETUP PS<15,13> AS REQUIRED
12953 FLAG(8-0)+D(15-8)-[I], ;SETUP FLAG<2:1> AS REQUIRED
12954 NEXT, J/KTSRCDST04B
12955 (7125) DCS(0.00.0.0.0.0) BM(0000..00.00..00.01..000..011...0.0.0..C..0...1.1011...0..0000.0...11.000...001.010.110)

12956 7126: !(FREE)
12957 KTSRCDST04B:
12958 P2-T, D+CSPD(C052525), SAVE-D(C), ;THIS WORD NEEDED FOR 1 UWORD DELAY FOR ROM TO SETTLE
12959 P3, A#BSPHI(02)+D, ;DATA PATTERN IN SCRATCH PAD ADDRESS (02)
12960 NEXT, BUTR(D(C)-B), ;BIT<3:0> = (05)
12961 ;KEY=(1) -> READ ASPHI[DF] -> SR
12962 ;KEY=(0) -> READ ASPHI[SF] -> SR
12963 (7126) DCS(0.00.0.0.0.0) BM(1010..10.10..00.00..101..111...0.1.0..0..0...0.0111...0..1011.0...10.011...011.011.101)

12964 !ENTER HERE IF D(C) CLEAR, SF SELECTED
12965 7335:
12966 KTSRCDST05:
12967 P2-T, SR+ASPHI(SF), ;USE ASP/SF
12968 NEXT, J/BUTSR3-0 ;AND NOW GO CHECK WHOM WAS READ
12969 (7335) DCS(0.00.0.0.0.0) BM(1111..00.00..11.11..000..000...0.0.1..0..0...0.0000...0..0000.0...11.000...010.111.110)

12970 !ENTER HERE IF D(C) SET, DF SELECTED
12971 7337:
12972 KTSRCDST06:
12973 P2-T, SR+ASPHI(DF), ;USE ASP/DF
12974 NEXT, J/BUTSR3-0 ;AND NOW GO CHECK WHOM WAS READ
12975 (7337) DCS(0.00.0.0.0.0) BM(1111..00.00..11.10..000..000...0.0.1..0..0...0.0000...0..0000.0...11.000...010.111.110)

12976 !* WE ALSO NEED TWO ENTRY POINTS TO READ BSP SF & DF:
12977
12978 7127: !(FREE)
12979 KTSRCDST07:
12980 P2-T, SR+BSPHI(SF), ;USE BSP/SF
12981 NEXT, J/BUTSR3-0 ;AND NOW GO CHECK WHOM WAS READ
12982 (7127) DCS(0.00.0.0.0.0) BM(1010..01.01..00.00..000..000...0.0.1..0..0...0.0000...0..0000.0...11.000...010.111.110)

12983
12984 7130: !(FREE)
12985 KTSRCDST08:
12986 P2-T, SR+BSPHI(DF), ;USE BSP/DF
12987 NEXT, J/BUTSR3-0 ;AND NOW GO CHECK WHOM WAS READ
12988 (7130) DCS(0.00.0.0.0.0) BM(1010..01.00..00.00..000..000...0.0.1..0..0...0.0000...0..0000.0...11.000...010.111.110)
12989
12990

```

```

12991
12992 ! -----
12993 !
12994 !   *** KT SRC/DST ENTERS HERE ***
12995 ! -----
12996
12997
12998
12999 !THESE FIRST TWO WORDS DO SOME PRELIMINARY SETUP OF SCRATCHPAD LOCATIONS (06)/(16)
13000 6555:
13001   SETUP512A:
13002     P2-T,   D+CSPD(C125252), D(C)+0,
13003     P3,     A#BSPHI(06)+0,
13004     NEXT,   J/SETUP512B
(6555) DCS(0.00.0.0.0.0) BM(1010..10.10..00.00..111..000...0.1.0..0..0...0.0110...0..1011.0...11.000...011.000.1111)
!
! KERNAL SP LOCATION:
! BIT<3:0> = (12) = SP-ADDRESS(06)
!
13005
13006 6307: !(FREE)
13007   SETUP512B:
13008     P2-T,   D+CSPD(C000000), D(C)+0,
13009     P3,     A#BSPHI(16)+0,
13010     NEXT,   J/TEST512A1
(6307) DCS(0.00.0.0.0.0) BM(1010..10.10..00.00..011..000...0.1.0..0..0...0.0100...0..1011.0...11.000...101.100.0101)
!
! USER SP LOCATION:
! BIT<3:0> = (00) = SP-ADDRESS(16)
!
13011
13012 !A,BSPHI(02) = (052525)
13013 ! BIT<3:0> = (05) = SP-ADDRESS(02)
13014
13015 !A,BSPHI(12) = (000152)
13016 ! BIT<3:0> = (12) = SP-ADDRESS(12)
13017 ! (NOT USED UNLESS ERROR)
13018
13019
13020 ! -----
13021
13022 !* TEST 512 A1 *
13023
13024 ! TEST 512 A 1-2 SETS UP FOR: ASP-SF-ADDRESS=(02), BSP-DF-ADDRESS=(16)
13025 6542:
13026   TEST512A1:
13027     PO,     LOAD-ENVA(ZTARGET405),
13028             LOAD-ERROR(TEST512A1),
13029             DCS-CTR(C11.),
13030             NEXT, J/SETIR512A1
(6542) DCS(1.00.1.0.0.0) BM(0100..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...101.110.1001)
!
! BIT<3:0> = (05) = SP-ADDRESS(02)
! ERROR DIRECTORY KEY
! COMPARE AT TARGET
!
13031
13032 6564:
13033   SETIR512A1:
13034     PO,     BUMP-VERIFY,
13035             CSPD(17)+EMIT, EMIT/172206,
13036             NEXT, J/SETPSFLAG512A1
(6564) DCS(0.00.0.0.0.1) BM(1111..10.01..00.10..000..110...0.0.0..0..0...0.0000...1..0000.0...11.000...011.001.0001)
!
! COUNT
! SETUP IR: FLTPT/SMD/SR6/DMD/DR6
!
13037
13038 6310: !(FREE)
13039   SETPSFLAG512A1:
13040     P3,     CSPD(16)+EMIT, EMIT/103000,
! BIT<15,13> -> PS<15,13>,

```

```

13041                                     !BIT<10:9> -> FLAGS<2:1>
13042      NEXT, J/GOTESTS12A1              !BIT<00> IS REGISTER KEY (SEE SUBR)
(6310) DCS(0.00.0.0.0.0) BM(1000..10.01..10.00..000..000...0.0.0..0..0...0.0001...1..0000.0...11.000...011.001.001)
13043
13044      6311: !(FREE)
13045      GOTESTS12A1:
13046      SETUP, RETURN/TESTS12A2,        !GO EXEC KT SRC/DST TEST SUBR AT
13047      NEXT, CALL(KTSRCDST)             ! INITIALIZATION POINT
(6311) DCS(0.00.0.0.0.0) BM(0110..00.10..11.00..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.001.100)
13048
13049
13050
13051      !* TEST 512A2 NOW READS THE COMPLEMENTARY REGISTER TO THAT USED IN TESTS12A1, INTO THE SR
13052      6544:
13053      TESTS12A2:
13054      PO, LOAD-ENUA(ZTARGET400),        !BIT<3:0> = (00) = SP-ADDRESS(16)
13055      LOAD-ERROR(TESTS12A2),           !ERROR DIRECTORY KEY
13056      DCS-CTR(C4.)                     !COMPARE AT TARGET
13057      NEXT, J/GOTESTS12A2
(6544) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..000..000...0.0.C..0..0...0.0000...0..0000.0...11.000...011.001.010)
13058
13059      6312: !(FREE)
13060      GOTESTS12A2:
13061      SETUP, RETURN/SCOPE512A,        !READ REGISTER BSPHI(DF) TO SR.
13062      NEXT, CALL(KTDSTBSP)            ! THEN DO BUT(SR3-0)
(6312) DCS(0.00.0.0.0.0) BM(0110..00.01..10.01..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.011.000)
13063
13064
13065      6313: !(FREE)
13066      SCOPE512A:
13067      PO, BUSDIN+EMIT-[I],            !RESET PROC UCON
13068      EN-CLK-IR[15-00],
13069      NEXT, BUTC(SCOPE),                !NO ERROR: "TESTS12B1" (+1. WORDS)
13070      J/TESTS12B1                       ! ERROR: "SETIRS12A1" (-5. WORDS)
(6313) DCS(0.00.0.1.0.0) BM(0000..00.00..00.01..000..100...0.0.0..C..0...1.1001...0..0000.0...11.000...101.110.101)
13071
13072
13073
13074
13075      ! - - - - -
13076
13077      !* TEST 512 B1 *
13078
13079      ! TEST 512 B 1-2 SETS UP FOR: ASP-DF-ADDRESS=(06), BSP-SF-ADDRESS=(16)
13080      6565:
13081      TESTS12B1:
13082      PO, LOAD-ENUA(ZTARGET412),        !BIT<3:0> = (12) = SP-ADDRESS(06)
13083      LOAD-ERROR(TESTS12B1),           !ERROR DIRECTORY KEY
13084      DCS-CTR(C11.),                   !COMPARE AT TARGET
13085      NEXT, J/SETIRS12B1
(6565) DCS(1.00.1.0.0.0) BM(0100..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...101.011.110)
13086

```

```

13087 6536:
13088 SETIRS1281:
13089 PO, BUMP-VERIFY, !COUNT
13090 P3, CSPD[17]+EMIT, EMIT/070606, !SETUP IR: FLTPT/SMD/SR6/DMD/DR6
13091 NEXT, J/SETPSFLAG51281
(6536) DCS[0.00.0.0.0.1] BM[0111..10.00..01.10..000..110...0.0.0..0..0...0.0000...1..0000.0...11.000...011.001.100]

13092
13093 6314: !(FREE)
13094 SETPSFLAG51281:
13095 P3, CSPD[16]+EMIT, EMIT/022001, !BIT<15,13> -> PS<15,13>,
13096 !BIT<10:9> -> FLAGS<2:1>
13097 NEXT, J/GOTEST51281 !BIT<00> IS REGISTER KEY (SEE SUBR)
(6314) DCS[0.00.0.0.0.0] BM[0010..10.01..00.00..000..001...0.0.0..0..0...0.0001...1..0000.0...11.000...011.001.101]

13098
13099 6315: !(FREE)
13100 GOTEST51281:
13101 SETUP, RETURN/TEST51282, !GO EXEC KT SRC/DST TEST SUBR AT
13102 NEXT, CALL[KTSRCDST] ! INITIALIZATION POINT
(6315) DCS[0.00.0.0.0.0] BM[0110..00.10..11.00..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.001.100]

13103
13104
13105 !* TEST 51282 NOW READS THE COMPLEMENTARY REGISTER TO THAT USED IN TEST51281, INTO THE SR
13106
13107 6546:
13108 TEST51282:
13109 PO, LOAD-ENUA(ZTARGET400), !BIT<3:0> = (00) = SP-ADDRESS(16)
13110 LOAD-ERROR(TEST51282), !ERROR DIRECTORY KEY
13111 DCS-CTR(C4.), !COMPARE AT TARGET
13112 NEXT, J/GOTEST51282
(6546) DCS[1.00.1.0.0.0] BM[1011..00.11..11.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...011.001.110]

13113
13114 6316: !(FREE)
13115 GOTEST51282:
13116 SETUP, RETURN/TEST512C1, !READ REGISTER BSPHI[SF] TO SR,
13117 NEXT, CALL[KTSRCSBP] ! THEN DO BUT(SR3-0)
(6316) DCS[0.00.0.0.0.0] BM[0110..00.10..11.01..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.010.111]

13118
13119
13120
13121
13122
13123
13124 ! - - - - -
13125 !* TEST 512 C1 *
13126
13127 ! TEST 512 C 1-2 SETS UP FOR: ASP-DF-ADDRESS=(16), BSP-SF-ADDRESS=(06)
13128
13129 6556:
13130 TEST512C1:
13131 PO, LOAD-ENUA(ZTARGET400), !BIT<3:0> = (00) = SP-ADDRESS(16)
13132 LOAD-ERROR(TEST512C1), !ERROR DIRECTORY KEY
13133 DCS-CTR(C11.), !COMPARE AT TARGET

```

```

13134      NEXT, J/SETIRS12C1
(6556) DCS[1.00.1.0.0.0] BM[0100..00.11..11.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...011.001.111]
13135
13136      6317: !(FREE)
13137      SETIRS12C1:
13138          PO,      BUMP-VERIFY,          !COUNT
13139          P3,      CSPD[17]+EMIT, EMIT/134606, !SETUP IR: FLTPT/SMD/SR6/DMD/DR6
13140      NEXT, J/SETPSFLAG512C1
(6317) DCS[0.00.0.0.0.1] BM[1011..10.10..01.10..000..110...0.0.0..0..0...0.0000...1..0000.0...11.000...011.010.000]
13141
13142      6320: !(FREE)
13143      SETPSFLAG512C1:
13144          P3,      CSPD[16]+EMIT, EMIT/023001, !BIT<15,13> -> PS<15,13>
13145          !BIT<10:9> -> FLAGS<2:1>
13146      NEXT, J/GOTEST512C1
(6320) DCS[0.00.0.0.0.0] BM[0010..10.01..10.00..000..001...0.0.0..0..0...0.0001...1..0000.0...11.000...011.010.001]
13147
13148      6321: !(FREE)
13149      GOTEST512C1:
13150          SETUP,   RETURN/TEST512C2,      !GO EXEC KT SRC/DST TEST SUBR AT
13151          NEXT,    CALL[KTSRCDST]         ! INITIALIZATION POINT
(6321) DCS[0.00.0.0.0.0] BM[0110..00.10..11.10..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.001.100]
13152
13153
13154
13155      !* TEST 512C2 NOW READS THE COMPLEMENTARY REGISTER TO THAT USED IN TEST512C1, INTO THE SR
13156      6566:
13157      TEST512C2:
13158          PO,      LOAD-ENUA(ZTARGET412), !BIT<3:0> = (12) = SP-ADDRESS(06)
13159          LOAD-ERROR(TEST512C2),         !ERROR DIRECTORY KEY
13160          DCS-CTR(C4.),                  !COMPARE AT TARGET
13161      NEXT, J/GOTEST512C2
(6566) DCS[1.00.1.0.0.0] BM[1011..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.010.010]
13162
13163      6322: !(FREE)
13164      GOTEST512C2:
13165          SETUP,   RETURN/SCOPE512C,      !READ REGISTER BSPHI[5F] TO SR,
13166          NEXT,    CALL[KTSRCSBP]         ! THEN DO BUT(SR3-0)
(6322) DCS[0.00.0.0.0.0] BM[0110..00.01..10.10..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.010.111]
13167
13168
13169      6323: !(FREE)
13170      SCOPE512C:
13171          PO,      BUSDIN+EMIT-[I],      !RESET PROC UCON
13172          EN-CLK-IR[15-00],
13173          NEXT,    BUTD(SCOPE),
13174          J/TEST512D1
(6323) DCS[0.00.0.1.0.0] BM[0000..00.00..00.01..000..100...0.0.0..0..0...1.1001...0..0000.0...11.000...101.011.111]
13175
13176
13177
13178

```



KD11-k MICRO V00P-1 00:00:03 12-MAR-77

```

13179 ! - - - - -
13180
13181 !* TEST 512 D1 *
13182
13183 ! TEST 512 D 1-2 SETS UP FOR: ASP-SF-ADDRESS=(16), BSP-DF-ADDRESS=(02)
13184 6537:
13185 TESTS12D1:
13186 PO, LOAD-ENVA(ZTARGET400), !BIT<3:0> = (00) = SP-ADDRESS(16)
13187 LOAD-ERROR(TESTS12D1), !ERROR DIRECTORY KEY
13188 DCS-CTR(C11.), !COMPARE AT TARGET
13189 NEXT J/SETIRS12D1
(6537) DCS(1.00.1.0.0.0) BM(0100..00.11..11.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.010.110)

13190
13191 6626:
13192 SETIRS12D1:
13193 PO, BUMP-VERIFY !COUNT
13194 P3, CSPD(17)+EMIT, EMIT/160612, !SETUP IR: FLTPT/SMD/SR6/DMD/DR6
13195 NEXT J/SETPSFLAG512D1
(6626) DCS(0.00.0.0.0.1) BM(1110..10.00..01.10..001..010...0.0.0..0..0...0.0000...1..0000.0...11.070...011.010.100)

13196
13197 6324: !(FREE)
13198 SETPSFLAG512D1:
13199 P3, CSPD(16)+EMIT, EMIT/101000, !BIT<15,13> -> PS<15,13>,
13200 !BIT<10:9> -> FLAGS<2:1>
13201 NEXT J/GOTESTS12D1 !BIT<00> IS REGISTER KEY (SEE SUBR)
(6324) DCS(0.00.0.0.0.0) BM(1000..10.00..10.00..000..000...0.0.0..0..0...0.0001...1..0000.0...11.000...011.010.101)

13202
13203 6325: !(FREE)
13204 GOTESTS12D1:
13205 SETUP, RETURN/TEST512D2, !GO EXEC KT SRC/DST TEST SUBR AT
13206 NEXT CALL(KTSRCDST) ! INITIALIZATION POINT
(6325) DCS(0.00.0.0.0.0) BM(0110..00.10..11.11..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.001.100)

13207
13208
13209
13210 !* TEST 512D2 NOW READS THE COMPLEMENTARY REGISTER TO THAT USED IN TESTS12D1, INTO THE SR
13211 6576:
13212 TESTS12D2:
13213 PO, LOAD-ENVA(ZTARGET405), !BIT<3:0> = (05) = SP-ADDRESS(02)
13214 LOAD-ERROR(TESTS12D2), !ERROR DIRECTORY KEY
13215 DCS-CTR(C4.) !COMPARE AT TARGET
13216 NEXT J/GOTESTS12D2
(6576) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...011.010.110)

13217
13218 6326: !(FREE)
13219 GOTESTS12D2:
13220 SETUP, RETURN/TEST512E1, !READ REGISTER BSPHI(DF) TO SR,
13221 NEXT CALL(KTDSTBSP) ! THEN DO BUT(SR3-0)
(6326) DCS(0.00.0.0.0.0) BM(0110..00.10..11.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.011.000)

13222
13223
13224

```

```

13225
13226
13227
13228 ! - - - - -
13229
13230 !* TEST 512 E1 *
13231
13232 ! TEST 512 E 1-2 SETS UP FOR: ASP-DF-ADDRESS=(16), BSP-SF-ADDRESS=(06)
13233 6577:
13234 TESTS12E1:
13235     PO,          LOAD-ENUA(ZTARGET400),          !BIT<3:0> = (00) = SP-ADDRESS(16)
13236     LOAD-ERROR(TESTS12E1),          !ERROR DIRECTORY KEY
13237     DCS-CTR(C11.),          !COMPARE AT TARGET
13238     NEXT,        J/SETIRS12E1
(6577) DCS(1.00.1.0.0.0) BM(0100..00.11..11.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...011.010.111)

13239
13240 6327: !(FREE)
13241 SETIRS12E1:
13242     PO,          BUMP-VERIFY,          !COUNT
13243     P3,          CSPD(17)+EMIT, EMIT/150626, !SETUP IR: FLTPT/SMO/SR6/DMD/OR6
13244     NEXT,        J/SETPSFLAG512E1
(6327) DCS(0.00.0.0.0.1) BM(1101..10.00..01.10..010..110...0.0.0..0..0...0.0000...1..0000.0...11.000...011.011.000)

13245
13246 6330: !(FREE)
13247 SETPSFLAG512E1:
13248     P3,          CSPD(16)+EMIT, EMIT/103001, !BIT<15,13> -> PS<15,13>,
13249     NEXT,        J/GOTESTS12E1          !BIT<10:9> -> FLAGS<2:1>
13250 (6330) DCS(0.00.0.0.0.0) BM(1000..10.01..10.00..000..001...0.0.0..0..0...0.0001...1..0000.0...11.000...011.011.001)

13251
13252 6331: !(FREE)
13253 GOTESTS12E1:
13254     SETUP,      RETURN/TESTS12E2,      !GO EXEC KT SRC/DST TEST SUBR AT
13255     NEXT,        CALL(KTSRCDST)        ! INITIALIZATION POINT
(6331) DCS(0.00.0.0.0.0) BM(0110..00.10..10.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.001.100)

13256
13257
13258
13259 !* TEST 512E2 NOW READS THE COMPLEMENTARY REGISTER TO THAT USED IN TESTS12E1, INTO THE SR
13260 6535:
13261 TESTS12E2:
13262     PO,          LOAD-ENUA(ZTARGET412),          !BIT<3:0> = (12) = SP-ADDRESS(06)
13263     LOAD-ERROR(TESTS12E2),          !ERROR DIRECTORY KEY
13264     DCS-CTR(C4.),          !COMPARE AT TARGET
13265     NEXT,        J/GOTESTS12E2
(6535) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.011.010)

13266
13267 6332: !(FREE)
13268 GOTESTS12E2:
13269     SETUP,      RETURN/SCOPE512E,      !READ REGISTER BSPHI(SF) TO SR,
13270     NEXT,        CALL(KTSRCBSP)        ! THEN DO BUT(SR3-0)
(6332) DCS(0.00.0.0.0.0) BM(0110..00.01..10.11..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.010.111)

```

13271  
13272  
13273  
13274  
13275  
13276  
13277  
13278  
13279

6333: !(FREE)

SCOPE SIZE:

PO, BUSDIN+EMIT-(I),  
EN-CLK-IR(15-00),  
NEXT, BUTD(SCOPE),  
J/TEST520A

!RESET PROC UCON

!NO ERROR: "TEST520A" (+1. WORDS)

!ERROR: "SETIR51201" (-11. WORDS)

(6333) DCS(0.00.0.1.0.0) BM(0000..00.00..00.01..000..100...0.0.0..0..0...1.1001...0..0000.0...11.000...110.010.111)

13280  
13281  
13282  
13283  
13284

!.PAGE=====

13285  
13286  
13287

.TOC \* TEST520A-520E: TESTING THE "INSTR BRANCH" ROM

13288  
13289

!\*\*\*\*\*

! \* TESTS: 520A - 520E

! \* UWORDS: 020 + 031

! \* FUNCTIONS:

! \* THE FOLLOWING FIVE TESTS VERIFY THE VALIDITY OF THE INSTRUCTION BRANCH ROM  
! \* INPUTS AND OUTPUTS, IN REGARD TO NO STUCK ONE/ZERO CONDITIONS.

!\*\*\*\*\*

13290  
13291  
13292  
13293  
13294  
13295  
13296  
13297  
13298  
13299

! SUMMARY OF INSTR BRANCH ROM TESTS:

TEST NUMB	IR<15,10:08>H	B-- ?	ON:	PS<3:0>H N Z V C	INSTR BRANCH L
520A	1 0 1 0	BHI	C.IOR.Z=0	0 1 0 0	1, NEGATED
520B	0 1 1 0	BGT	Z ^OR.(N.XOR.V)=0	0 0 0 0	0, ASSERTED
520C	0 1 0 1	BLT	N.XOR.V=1	1 1 1 0	1, NEGATED
520D	1 0 1 1	BLOS	C.IOR.Z=1	0 0 0 1	0, ASSERTED
520E	1 1 1 1	BLO	C=1	0 1 1 0	1, NEGATED

13300  
13301  
13302  
13303  
13304  
13305  
13306  
13307  
13308  
13309  
13310  
13311  
13312  
13313  
13314  
13315  
13316  
13317

! - - - - -

!\*\*\* TEST 520A \*\*\*

!TEST-520-A SETS UP IR<15,10:08>H="1010", PS<3:0>H="0100",  
! AND THEN BUTS ON "INSTR BRANCH L"

13321  
13322  
13323

6627:

TEST520A:

PO, LOAD-ENUA(ZTARGET403),

!NOT ASSERTED

```

13324 LOAD-ERROR(TEST520A),          !ERROR DIRECTORY KEY
13325 DCS-CTR(C8.),                 !COMPARE ENUA:TNUA AT TARGET
13326 NEXT,                          J/UCONS20A
(6627) DCS[1.00.1.0.0.0] BM[0111..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...110.010.100]

13327
13328 6624:
13329 UCONS20A:
13330 SELECT, UCON-PROC                !PROCESSOR UCON:
13331 ENABLE, BUSDIN+EMIT[15-00],      !EMIT ON BUSDIN
13332 EN-CLK-IR[15-00],               !AND CLOCKING IR
13333 PO, BUMP-VERIFY,                 !COUNT
13334 SET-UCON-CONTROL,               !WRITE CONTROLS
13335 NEXT,                          J/SETUPS20A
(6624) DCS[0.00.0.0.0.1] BM[0000..00.00..00.01..000..100...0.0.0..0..0...1.1001...0..0000.0...11.000...011.011.100]

13336
13337 6334: !(FREE)
13338 SETUPS20A:
13339 PO, BUMP-VERIFY,                 !COUNT
13340 EMITC, EMIT/101004,             !PS<3:0>H="0100"
13341 P2-U, IR+EMIT,                  !IR<15,10:08>H="1010"
13342 P3, CSPD[05]+EMIT,
13343 NEXT,                          J/GOTEST520A
(6334) DCS[0.00.0.0.0.1] BM[1000..10.00..10.00..000..100...0.0.0..0..0...1.1010...1..0000.0...11.000...011.011.101]

13344
13345 6335: !(FREE)
13346 GOTEST520A:
13347 SETUP, RETURN/TEST520B,         !RETURN TO START OF NEXT SUBTEST
13348 NEXT, GOTO-PAGE(7),             !GO TO CODE THAT LOADS PS[CC], AND
13349 J/SUCBRTEST01,                   ! THEN BUTS ON THE "INSTR BRANCH" ROM OUTPUT
(6335) DCS[0.00.0.0.0.0] BM[0110..00.11...11.10..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.010.010]

13350
13351
13352
13353
13354 ! - - - - -
13355
13356 !*** TEST 520B ***
13357 !TEST-520-B SETS UP IR<15,10:08>H="0110", PS<3:0>H="0000",
13358 ! AND THEN BUTS ON "INSTR BRANCH L"
13359 6764:
13360 TEST520B:
13361 PO, LOAD-ENUA(ZTARGET402),        !ASSERTED
13362 LOAD-ERROR(TEST520B),            !ERROR DIRECTORY KEY
13363 DCS-CTR(C7.),                    !COMPARE ENUA:TNUA AT TARGET
13364 NEXT,                          J/SETUPS20B
(6764) DCS[1.00.1.0.0.0] BM[1000..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.011.110]

13365
13366 6336: !(FREE)
13367 SETUPS20B:
13368 EMITC, EMIT/003000,               !PS<3:0>H="0000"
13369 P2-U, IR+EMIT,
13370 P3, CSPD[05]+EMIT,
13371 NEXT,                          J/GOTEST520B

```

```

13372 (6336) DCS(0.00.0.0.0.0) BM(0000..10.01..10.00..000..000...0.0.0..0..0...1.1010...1..0000.0...11.000...011.011.111)
13373 6337: !(FREE)
13374 GOTESTS208:
13375     SETUP, RETURN/TEST520C, !RETURN TO START OF NEXT SUBTEST
13376     NEXT, GOTO-PAGE(7) !GO TO CODE THAT LOADS PS(CC), AND
13377     J/SUCBRTEST01 ! THEN BUTS ON THE "INSTR BRANCH" ROM OUTPUT
(6337) DCS(0.00.0.0.0.0) BM(0110..00.11..11.11..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.010.010)

```

```

13378
13379
13380
13381
13382 ! - - - - -
13383
13384 !*** TEST 520C ***
13385 !TEST-520-C SETS UP IR<15,10:08>H="0101", PS<3:0>H="1110",
13386 ! AND THEN BUTS ON "INSTR BRANCH L"
13387 6772:
13388 TEST520C:
13389     PO, LOAD-ENUA(ZTARGET403), !NEGATED
13390     LOAD-ERROR(TEST520C), !ERROR DIRECTORY KEY
13391     DCS-CTR(C7.), !COMPARE ENUA:TNUA AT TARGET
13392     BUMP-VERIFY, !COUNT
13393     NEXT, J/SETUP520C
(6772) DCS(1.00.1.0.0.1) BM(1000..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...011.100.000)

```

```

13394
13395 6340: !(FREE)
13396 SETUP520C:
13397     EMITC, EMIT/002416, !PS<3:0>H="1110"
13398     P2-U, IR+EMIT, !
13399     P3, CSPD(05)+EMIT, !IR<15,10:08>H="0101"
13400     NEXT, J/GOTESTS20C
(6340) DCS(0.00.0.0.0.0) BM(0000..10.01..01.00..001..110...0.0.0..0..0...1.1010...1..0000.0...11.000...011.100.001)

```

```

13401
13402 6341: !(FREE)
13403 GOTESTS20C:
13404     SETUP, RETURN/TEST5200, !RETURN TO START OF NEXT SUBTEST
13405     NEXT, GOTO-PAGE(7) !GO TO CODE THAT LOADS PS(CC), AND
13406     J/SUCBRTEST01 ! THEN BUTS ON THE "INSTR BRANCH" ROM OUTPUT
(6341) DCS(0.00.0.0.0.0) BM(0110..00.11..11.10..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.010.010)

```

```

13407
13408
13409
13410
13411 ! - - - - -
13412
13413 !*** TEST 520D ***
13414 !TEST-520-D SETS UP IR<15,10:08>H="1011", PS<3:0>H="0001",
13415 ! AND THEN BUTS ON "INSTR BRANCH L"
13416 6762:
13417 TEST5200:
13418     PO, LOAD-ENUA(ZTARGET402), !ASSERTED

```

```

13419 LOAD-ERROR(TEST520D), !ERROR DIRECTORY KEY
13420 DCS-CTR(C7.), !COMPARE ENUA:TNUA AT TARGET
13421 NEXT J/SETUPS200
(6762) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.100.010)
13422
13423 6342: !(FREE)
13424 SETUP5200:
13425 EMITC, EMIT/101401, !PS<3:0>H="0001"
13426 P2-U, IR+EMIT, !IR<15,10:08>H="1011"
13427 P3, CSPD(05)+EMIT,
13428 NEXT J/GOTEST5200
(6342) DCS(0.00.0.0.0.0) BM(1000..10.00..11.00..000..001...0.0.0..0..0...1.1010...1..0000.0...11.000...011.100.011)
13429
13430 6343: !(FREE)
13431 GOTEST5200:
13432 SETUP, RETURN/TEST520E, !RETURN TO START OF NEXT SUBTEST
13433 NEXT, GOTO-PAGE(7) !GO TO CODE THAT LOADS PS(CC), AND
13434 J/SUCBRTEST01 ! THEN BUTS ON THE "INSTR BRANCH" ROM OUTPUT
(6343) DCS(0.00.0.0.0.0) BM(0110..00.11..11.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.010.010)
13435
13436
13437
13438
13439 ! - - - - -
13440
13441 !*** TEST 520E ***
13442 !TEST-520-E SETS UP IR<15,10:08>H="1111", PS<3:0>H="0110",
13443 ! AND THEN BUTS ON "INSTR BRANCH L"
13444 6775:
13445 TEST520E:
13446 PO, LOAD-ENUA(ZTARGET403), !NEGATED
13447 LOAD-ERROR(TEST520E), !ERROR DIRECTORY KEY
13448 DCS-CTR(C7.), !COMPARE ENUA:TNUA AT TARGET
13449 NEXT J/SETUPS200
(6775) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...011.100.100)
13450
13451 6344: !(FREE)
13452 SETUP520E:
13453 PO, BUMP-VERIFY, !COUNT
13454 EMITC, EMIT/103406, !PS<3:0>H="0110"
13455 P2-U, IR+EMIT,
13456 P3, CSPD(05)+EMIT, !IR<15,10:08>H="1111"
13457 NEXT J/GOTEST520E
(6344) DCS(0.00.0.0.0.1) BM(1000..10.01..11.00..000..110...0.0.0..0..0...1.1010...1..0000.0...11.000...011.100.101)
13458
13459 6345: !(FREE)
13460 GOTEST520E:
13461 SETUP, RETURN/SCOPE520, !RETURN TO SCOPE LOOP TEST WORD
13462 NEXT, GOTO-PAGE(7) !GO TO CODE THAT LOADS PS(CC), AND
13463 J/SUCBRTEST01 ! THEN BUTS ON THE "INSTR BRANCH" ROM OUTPUT
(6345) DCS(0.00.0.0.0.0) BM(0110..00.01..11.00..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.010.010)
13464

```

```

13465
13466
13467
13468 6346: !(FREE)
13469 SCOPE520:
13470     PD,      BUSDIN+EMIT-[I],      !RESET PROC UCONS
13471           EN-CLK-IR[15-00],
13472     NEXT,    BUTD(SCOPE),           !NO ERROR: "TEST533A" [+4. WORDS]
13473           J/TEST533A                !ERROR: "UCONS20A" [-15. WORDS]
(6346) DCS[0.00.0.1.0.0] BM[0000..00.00..00.01..000..100...0.0.0..0..0...1.1001...0..0000.0...11.000...110.010.101]

```

```

13474
13475
13476
13477 !-----
13478 !
13479 ! TESTING SUBR (COMMON CODE) FOR ABOVE TESTS

```

```

13480
13481 7122: !(FREE)
13482 SUCBRTEST01:
13483     P2-T,    D+CSPD(DOS), D[C]+0,    !GET PATTERN
13484     NEXT,    J/SUCBRTEST02
(7122) DCS[0.00.0.0.0.0] BM[1010..10.00..00.00..000..000...0.1.0..0..0...0.1010...0..0000.0...11.000...001.011.010]

```

```

13485
13486 7132: !(FREE)
13487 SUCBRTEST02:
13488     P2-T,    PS[3-0]+D[3-0]-[I],    !INTO PS[CC]
13489     NEXT,    J/SUCBRTEST03
(7132) DCS[0.00.0.0.0.0] BM[1000..00.00..00.01..000..000...0.0.0..0..0...1.1011...0..0000.0...11.000...001.011.011]

```

```

13490
13491 7133: !(FREE)
13492 SUCBRTEST03:
13493     SELECT,  UCON-PROC,              !LEAVE WITH EMIT ON BUSDIN,
13494     ENABLE,  BUSDIN+EMIT[15-00],    !CLOCK IR ENABLED
13495           EN-CLK-IR[15-00],
13496     PD,      SET-UCON-CONTROL,      !WRITE CONTROLS
13497     NEXT,    J/BUTINSTRBRANCH
(7133) DCS[0.00.0.0.0.0] BM[0000..00.00..00.01..000..100...0.0.0..0..0...1.1001...0..0000.0...1.000...011.101.110]

```

```

13498
13499
13500
13501
13502
13503 !.PAGE=====

```

```

13504
13505
13506 .TOC * TEST533-537: SHIFT TREE
13507
13508 !*****
13509 !*
13510 !* TESTS: 533A - 537A UWORDS: 170 + 044
13511 !*
13512 !* FUNCTIONS: TESTS 533A - 537A VERIFY THE DATA AND CONTROL PATHS
13513 !* OF THE 3 LEVEL BARREL SHIFTER (SHIFT TREE).
13514 !*

```

```

13515 *****
13516
13517
13518
13519
13520
13521
13522 ! - - - - -
13523
13524 !*** TEST 533A ***
13525 !READ D DIRECTLY THRU "D(HI)#D(LO)" PORT OF AMUX(HI#LO), BMUX-CMUX/DIRECT
13526 !IN(0)(052652), OUT(052652)
13527 6625:
13528 TEST533A:
13529     PO,      LOAD-ENVA(ZTARGET402),      !SETUP FOR D = ZERO TEST
13530     LOAD-ERROR(TEST533A),      !ERROR DIRECTORY KEY
13531     DCS-CTR(C6.),      !COMPARE AT TARGET
13532     NEXT,    J/INIT533A
(6625) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...110.110.110)

13533
13534 6666:
13535 INIT533A:
13536     PO,      BUMP-VERIFY,      !COUNT
13537     P3,      CSPD(17)+EMIT,      !GET INITIAL PATTERN FOR D
13538     EMIT/052652,      !"0101 0101 1010 1010"
13539     NEXT,    J/INITD533A
(6666) DCS(0.00.0.0.0.0.1) BM(0101..10.01..01.10..101..010...0.0.0..0..0...0.0000...1..0000.0...11.000...011.100.111)

13540
13541 6347: !(FREE)
13542 INITD533A:
13543     P2-T,    D+CSPD(017),      !INITIAL D =(052652)
13544     D(C)+ALU15,      !SETUP D(C) FOR SHIFT = "0"
13545     NEXT,    J/COMPS33A
(6347) DCS(0.00.0.0.0.0.0) BM(1010..10.00..00.00..000..100...0.1.0..0..0...0.0000...0..0000.0...11.000...011.101.000)

13546
13547 6350: !(FREE)
13548 COMPS33A:
13549     PO,      BUMP-VERIFY,      !COUNT
13550     SETUP,    D-DIRECT,      !AMUX-BMUX-CMUX ALL DIRECT
13551     P2-T,    D+D-SHIFTED-XOR-CSPB(817),      !COMPARE D-SHIFTED:EXPECTED, BITWISE
13552     NEXT,    J/GOBUT533A      !EXPECTED =(052652)
(6350) DCS(0.00.0.0.0.0.1) BM(0110..11.00..01.01..000..000...0.1.0..0..0...0.0000...0..0000.0...11.000...011.101.001)

13553
13554 6351: !(FREE)
13555 GOBUT533A:
13556     SETUP,    RETURN/TEST5338,      !RETURN TO START OF NEXT SUBTEST
13557     NEXT,    GOTO-PAGE(7),      !BUT TABLE IS ON PAGE 7
13558     J/BUTD-IS-ZERO      !GO TEST D IS ALL ZERO
(6351) DCS(0.00.0.0.0.0.0) BM(0110..00.11..00.11..011..11!...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

13559
13560
13561
13562

```



```

13563 ! - - - - -
13564
13565 !*** TEST 5338 ***
13566 !READ D DIRECTLY THRU "D(HI)D(LO)" PORT OF AMUX(HI#LO), BMUX-CMUX/DIRECT
13567 !IN(0)(125125), OUT(125125)
13568 6633:
13569 TEST5338:
13570     PO,      LOAD-EMUA(ZTARGET402),      !SETUP FOR D=ZERO TEST
13571           LOAD-ERROR(TEST5338),        !ERROR DIRECTORY KEY
13572           DCS-CTR(C6.),                !COMPARE AT TARGET
13573     NEXT,    J/INIT5338
(6633) DCS(1.00.1.0.0.0) BM(1001..00.11..1.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.101.0101)

13574
13575 6352: !(FREE)
13576 INIT5338:
13577     P3,      CSPD(16)+EMIT,              !GET INITIAL PATTERN FOR D
13578           EMIT/125125,                !"1010 1010 0101 0101"
13579     NEXT,    J/INIT5338
(6352) DCS(0.00.0.0.0.0) BM(1010..10.10..10.01..010..101...0.0.0..0..0...0.0001...1..0000.0...11.000...011.101.0111)

13580
13581 6353: !(FREE)
13582 INIT5338:
13583     PO,      BUMP-VERIFY,                !COUNT
13584           P2-T,    D+CSPD(016),        !INITIAL D=(125125)
13585           D(C)+ALU07,                !SETUP D(C) FOR SHIFT = "0"
13586     NEXT,    J/COMP5338
(6353) DCS(0.00.0.0.0.1) BM(1010..10.00..00.00..000..011...0.1.0..0..0...0.0001...0..0000.0...11.000...011.101.1001)

13587
13588 6354: !(FREE)
13589 COMP5338:
13590     SETUP,   D-DIRECT,                  !AMUX-BMUX-CMUX ALL DIRECT
13591           P2-T,    D+D-SHIFTED-XOR-CSPB(816), !COMPARE D-SHIFTED:EXPECTED, BITWISE
13592     NEXT,    J/GOBUT5338                !EXPECTED=(125125)
(6354) DCS(0.00.0.0.0.0) BM(0110..11.01..01.01..000..000...0.1.0..0..0...0.0000...0..0000.0...11.000...011.101.1011)

13593
13594 6355: !(FREE)
13595 GOBUT5338:
13596     SETUP,   RETURN/SCOPE5338,          !RETURN TO SCOPE LOOP TEST WORD
13597           NEXT, GOTO-PAGE(7),          !BUT TABLE IS ON PAGE 7
13598           J/BUTD-IS-ZERO                !GO TEST D IS ALL ZERO
(6355) DCS(0.00.0.0.0.0) BM(0110..00.01..11.01..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.0011)

13599
13600 6356: !(FREE)
13601 SCOPE5338:
13602     P3,      CSPD(14)+EMIT, EMIT/000377, !CONSTANT FOR USE BELOW
13603           NEXT, BUTD(SCOPE),          !NO ERROR: "TEST534A" (+1.WORDS)
13604           J/TEST534A                    ! ERROR: "INIT533A" (-9.WORDS)
(6356) DCS(0.00.0.1.0.0) BM(0000..10.00..00.11..111..111...0.0.0..0..0...0.0011...1..0000.0...11.000...110.110.1111)

13605
13606
13607
13608
13609 ! - - - - -

```

```

13610
13611 !*** TEST 534A ***
13612 !READ D THRU "D[LO]#D[HI]" PORT OF AMUX[HI,LO], BMUX-CMUX/DIRECT
13613 !IN(0)(052652), OUT(125125)
13614 6667:
13615 TEST534A:
13616     PO,      LOAD-ENUA(ZTARGET402),      !SETUP FOR D=ZERO TEST
13617           LOAD-ERROR(TEST534A),        !ERROR DIRECTORY KEY
13618           DCS-CTR(CS.),                 !COMPARE AT TARGET
13619           P3,      BUTA(CLR-FLAG-RES-UCON), !EMIT ON BUSDIN
13620     NEXT,    J/INITD534A
(6667) DCS(1.00.1.0.0.0) BM(1010..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.010...110.110.100)

```

```

13621
13622 6664:
13623 INITD534A:
13624     PO,      BUMP-VERIFY,                !COUNT
13625     P2-T,    D+CSPD(D17),                !INITIAL D=(052652)
13626           D[C]+ALU15,                    !SETUP D[C] FOR SHIFT = "0"
13627     NEXT,    J/COMPS34A
(6664) DCS(0.00.0.0.0.1) BM(1010..10.00..00.00..000..100...0.1.0..0..0...0.0000...0..0000.0...11.000...011.101.111)

```

```

13628
13629 6357: !(FREE)
13630 COMPS34A:
13631     SETUP,   D-SWAB,                      !AMUX/SWAB, BMUX-CMUX/DIRECT
13632     P2-T,    D+D-SHIFTED-XOR-CSPB(B16),   !COMPARE D-SHIFTED:EXPECTED, BITWISE
13633     NEXT,    J/GOBUT534A                 !EXPECTED=(125125)
(6357) DCS(0.00.0.0.0.0) BM(0110..11.01..01.01..000..000...0.1.0..0..0...0.0101...0..0000.0...11.000...011.110.000)

```

```

13634
13635 6360: !(FREE)
13636 GOBUT534A:
13637     SETUP,   RETURN/TEST534B,            !RETURN TO START OF NEXT SUBTEST
13638     NEXT,    GOTO-PAGE(7),                !BUT TABLE IS ON PAGE 7
13639           J/BUTD-IS-ZERO                 !GO TEST D IS ALL ZERO
(6360) DCS(0.00.0.0.0.0) BM(0110..00.11..00.11..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

```

```

13640
13641
13642
13643
13644 ! - - - - -
13645

```

```

13646 !*** TEST 534B ***
13647 !READ D THRU "D[LO]#D[HI]" PORT OF AMUX[HI,LO], BMUX-CMUX/DIRECT
13648 !IN(0)(125125), OUT(052652)
13649 6634:
13650 TEST534B:
13651     PO,      LOAD-ENUA(ZTARGET402),      !SETUP FOR D=ZERO TEST
13652           LOAD-ERROR(TEST534B),        !ERROR DIRECTORY KEY
13653           DCS-CTR(CS.),                 !COMPARE AT TARGET
13654           BUMP-VERIFY,                  !COUNT
13655     NEXT,    J/INITD534B
(6634) DCS(1.00.1.0.0.1) BM(1010..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.110.001)

```

```

13656 6361: !(FREE)
13657

```

```

13658 INITD534B:
13659     P2-T,   D+CSPD(016),           !INITIAL D=(125125)
13660     NEXT,   D(C)+ALU07,           !SETUP D(C) FOR SHIFT = "0"
13661     J/COMP534B
(6361) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..011...0.1.0..0..0...0.0001...0..0000.0...11.000...011.110.010)

13662 6362: !(FREE)
13663 COMP534B:
13664     PO,     BUMP-VERIFY,           !COUNT
13665     SETUP,  D-SWAB,                !AMUX/SWAB, BMUX-CMUX/DIRECT
13666     P2-T,   D+D-SHIFTED-XOR-CSPB(817), !COMPARE D-SHIFTED:EXPECTED, BITWISE
13667     NEXT,   J/GOBUT534B           !EXPECTED=(052652)
13668 (6362) DCS(0.00.0.0.0.1) BM(0110..11.00..01.01..000..000...0.1.0..0..0...0.0101...0..0000.0...11.000...011.110.011)

13669 6363: !(FREE)
13670 GOBUT534B:
13671     SETUP,  RETURN/SCOPE534B,      !RETURN TO SCOPE LOOP TEST WORD
13672     NEXT,   GOTO-PAGE(7),          !BUT TABLE IS ON PAGE 7
13673     J/BUTD-IS-ZERO                !GO TEST D IS ALL ZERO
13674 (6363) DCS(0.00.0.0.0.0) BM(0110..00.01..11.10..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

13675 6364: !(FREE)
13676 SCOPE534B:
13677     P3,     CSPD(15)+EMIT, EMIT/177400, !CONSTANT FOR USE BELOW
13678     NEXT,   BUTD(SCOPE),           !NO ERROR: "TEST534C" (+1.WORDS)
13679     J/TEST534C                    !ERROR: "INITD534A" (-7.WORDS)
13680 (6364) DCS(0.00.0.1.0.0) BM(1111..10.11..11.00..000..000...0.0.0..0..0...0.0010...1..0000.0...11.000...110.110.101)

13681 ! - - - - -
13682
13683
13684
13685
13686
13687 !*** TEST 534C ***
13688 !READ D THRU "8*D(C)#D(HI)" PORT OF AMUX(HI#LO), BMUX-CMUX/DIRECT
13689 !IN:1(000000), OUT(177400)
13690 6665:
13691 TEST534C:
13692     PO,     LOAD-EMUA(ZTARGET402),   !SETUP FOR D=ZERO TEST
13693     LOAD-ERROR( TEST534C),          !ERROR DIRECTORY KEY
13694     DCS-CTR(C5.),                  !COMPARE AT TARGET
13695     NEXT,   J/INITD534C
(6665) DCS(1.00.1.0.0.0) BM(1010..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...110.110.010)

13696 6662:
13697 INITD534C:
13698     P2-T,   D+ASPHI(000000),       !INITIAL D=(000000)
13699     NEXT,   D(C)+1,                !SETUP D(C) FOR SHIFT = "1"
13700     J/COMP534C
(6662) DCS(0.00.0.0.0.0) BM(1111..00.00..11.01..100..000...0.1.0..0..0...0.0000...0..0000.0...11.000...011.110.101)

13702 6365: !(FREE)
13703 COMP534C:
13704

```

```

13705                                     !CSP(15)=(177400)="1111 1111 0000 0000"
13706         SETUP, D-RIGHT-8,             !AMUX/8*D(C)#D(HI), BMUX-CMUX/DIRECT
13707         P2-T, D+D-SHIFTED-XOR-CSPB(B15), !COMPARE D-SHIFTED:EXPECTED, BITWISE
13708         NEXT, J/GOBUT534C             !EXPECTED=(177400)
(6365) DCS(0.00.0.0.0.0) BM(0110..11.10..01.01..000..000...0.1.0..0..0...0.0110...0..0000.0...11.000...011.110.110)

13709
13710         6366: !(FREE)
13711         GOBUT534C:
13712         SETUP, RETURN/TEST534D,       !RETURN TO START OF NEXT SUBTEST
13713         NEXT, GOTO-PAGE(7),           !BUT TABLE IS ON PAGE 7
13714         J/BUTD-IS-ZERO                !GO TEST D IS ALL ZERO
(6366) DCS(0.00.0.0.0.0) BM(0110..00.11..00.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

13715
13716
13717
13718
13719
13720
13721         !*** TEST 534D ***
13722         !READ D THRU "8*D(C)#D(L0)" PORT OF AMUX(HI#LO), BMUX-CMUX/DIRECT
13723         !IN(0)(177777), OUT(000377)
13724         6635:
13725         TEST534D:
13726         PO, LOAD-ENJIA(ZTARGET402),    !SETUP FOR D=ZERO TEST
13727         LOAD-ERROR(TEST534D),          !ERROR DIRECTORY KEY
13728         DCS-CTR(CS.),                  !COMPARE AT TARGET
13729         NEXT, J/INITD534D
(6635) DCS(1.00.1.0.0.0) BM(1010..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.110.111)

13730
13731         6367: !(FREE)
13732         INITD534D:
13733         P2-T, D+BSPHI(C177777),        !INITIAL D=(177777)
13734         D(C)+0,                        !SETUP D(C) FOR SHIFT = "0"
13735         NEXT, J/COMP534D
(6367) DCS(0.00.0.0.0.0) BM(1010..01.11..00.00..101..000...0.1.0..0..0...0.0000...0..0000.0...11.000...011.111.000)

13736
13737         6370: !(FREE)
13738         COMP534D:
13739                                     !CSP(14)=(000377)="0000 0000 1111 1111"
13740         SETUP, D-SIGNEXT,             !AMUX/8*D(C)#D(LO), BMUX-CMUX/DIRECT
13741         P2-T, D+D-SHIFTED-XOR-CSPB(B14), !COMPARE D-SHIFTED:EXPECTED, BITWISE
13742         NEXT, J/GOBUT534D             !EXPECTED=(000377)
(6370) DCS(0.00.0.0.0.0) BM(0110..11.11..01.01..000..000...0.1.0..0..0...0.0010...0..0000.0...11.000...011.111.001)

13743
13744         6371: !(FREE)
13745         GOBUT534D:
13746         PO, BUMP-VERIFY,               !COUNT
13747         SETUP, RETURN/SCOPE534D,        !RETURN TO SCOPE LOOP TEST WORD
13748         NEXT, GOTO-PAGE(7),           !BUT TABLE IS ON PAGE 7
13749         J/BUTD-IS-ZERO                !GO TEST D IS ALL ZERO
(6371) DCS(0.00.0.0.0.1) BM(0110..00.01..11.11..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

13750

```

```

13751 6372: !(FREE)
13752 SCOPE534D:
13753 NEXT, BUTD(SCOPE), !NO ERROR: "TEST534E" (+1.WORDS)
13754 J/TEST534E ! ERROR: "EXPEC534C" (-9.WORDS)
(6372) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000..0.0.0..0..0...0.0000...0..0000.0...11.000...110.110.011)

13755
13756
13757
13758
13759 ! - - - - -
13760
13761 !*** TEST 534E ***
13762 !READ D THRU "COUNTER#D(LO)" PORT OF AMUX(HI#LO), BMUX-CMUX/DIRECT
13763 !IN(0)(000125), OUT(125125), CTR(252)
13764 6663:
13765 TEST534E:
13766 PD, LOAD-ENUA(ZTARGET402), !SETUP FOR D =ZERO TEST
13767 LOAD-ERROR(TEST534E), !ERROR DIRECTORY KEY
13768 DCS-CTR(C6.) !COMPARE AT TARGET
13769 NEXT J/LODCNTRS534E
(6663) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...110.110.000)

13770
13771 6660:
13772 LODCNTRS534E:
13773 PD, BUMP-VERIFY, !COUNT
13774 P2, COUNTER+BSPHI(C125252), !PUT A (252) IN BM COUNTER
13775 NEXT J/INITD534E ! (GETS B-BUS(7:0))
(6660) DCS(0.00.0.0.0.1) BM(0000..01.11..00.00..110..000...0.0.0..0..0...0.0000...0..0010.1...11.000...011.111.011)

13776
13777 6373: !(FREE)
13778 INITD534E:
13779 PD, BUMP-VERIFY, !COUNT
13780 P2-T, D+CSPD(14)-AND-ASPHI(C052525), !INITIAL D=(000125), CSP(14)=(000377)
13781 D(C)+ALU15, !SETUP D(C) FOR SHIFT = "0"
13782 NEXT J/COMPS534E
(6373) DCS(0.00.0.0.0.1) BM(1011..10.00..11.01..111..100...0.1.0..0..0...0.0011...0..0000.0...11.000...011.111.100)

13783
13784 6374: !(FREE)
13785 COMPS534E:
13786 SETUP, COUNT#D(LO), !AMUX/COUNTER#D(LO) BMUX-CMUX/DIRECT
13787 P2-T, D+D-SHIFTED-XOR-CSPB(B16), !COMPARE D-SHIFTED:EXPECTED, BITWISE
13788 NEXT J/GOBUT534E !EXPECTED=(125125)
(6374) DCS(0.00.0.0.0.0) BM(0110..11.01..01.01..000..000...0.1.0..0..0...0.0011...0..0000.0...11.000...011.111.101)

13789
13790 6375: !(FREE)
13791 GOBUT534E:
13792 SETUP, RETURN/TEST534F, !RETURN TO START OF NEXT SUBTEST
13793 NEXT, GOTO-PAGE(7) !BUT TABLE IS ON PAGE 7
13794 J/BUTD-IS-ZERO !GO TEST D IS ALL ZERO
(6375) DCS(0.00.0.0.0.0) BM(0110..00.11..00.11..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

13795
13796
13797

```

```

13798
13799 ! - - - - -
13800
13801 !*** TEST 534F ***
13802 !READ D THRU "COUNTER#D(HI)" PORT OF AMUX(HI#LO), BMUX-CMUX/DIRECT
13803 !IN(0)(125000), OUT(052652), CTR(125)
13804 6636:
13805 TEST534F:
13806     PO,          LOAD-ENUA(ZTARGET402),          !SETUP FOR D=ZERO TEST
13807     LOAD-ERROR(TEST534F),          !ERROR DIRECTORY KEY
13808     DCS-CTR(C6.),          !COMPARE AT TARGET
13809     NEXT,        J/LODCNTRS34F
(6636) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.000.000)

13810
13811 6400: !(FREE)
13812 LODCNTRS34F:
13813     P2,          COUNTER+BSPHI(C052525),          !PUT A (125) IN BM COUNTER
13814     NEXT,        J/INITD534F          ! (GETS B-BUS[7-0])
(6400) DCS(0.00.0.0.0.0) BM(0000..01.11..00.00..111..000...0.0.0..0..0...0.0000...0..0010.1...11.000...100.000.001)

13815
13816 6401: !(FREE)
13817 INITD534F:
13818     PO,          BUMP-VERIFY,          !COUNT
13819     P2-T,        D+CSPD[15]-AND-ASPHI(C125252),  !INITIAL D=(125000), CSP(15)=(177400)
13820     D(C)+ALU00,  !SETUP D(C) FOR SHIFT = "0"
13821     NEXT,        J/COMPS34F
(6401) DCS(0.00.0.0.0.1) BM(1011..10.00..11.01..110..010...0.1.0..0..0...0.0010...0..0000.0...11.000...100.000.010)

13822
13823 6402: !(FREE)
13824 COMPS34F:
13825     SETUP,       COUNT#D(HI),          !AMUX/COUNTER#D(HI), BMUX-CMUX/DIRECT
13826     P2-T,        D+D-SHIFTED-XOR-CSPB(B17),    !COMPARE D-SHIFTED:EXPECTED, BITWISE
13827     P3,          BUTA(LAST)          !CLEAR CTR TO (000) DURING P3
13828     NEXT,        J/GOBUT534F          !EXPECTED=(052652)
(6402) DCS(0.00.0.0.0.0) BM(0110..11.00..01.01..000..000...0.1.0..0..0...0.0111...0..0000.0...10.000...100.000.011)

13829
13830 6403: !(FREE)
13831 GOBUT534F:
13832     SETUP,       RETURN/TEST534G,          !RETURN TO START OF NEXT SUBTEST
13833     NEXT,        GOTO-PAGE(7)          !BUT TABLE IS ON PAGE 7
13834     J/BUTD-IS-ZERO          !GO TEST D IS ALL ZERO
(6403) DCS(0.00.0.0.0.0) BM(0110..00.11..11.01..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

13835
13836 ! - - - - -
13837
13838 !*** TEST 534G ***
13839 !READ D THRU "COUNTER#D(LO)" PORT OF AMUX(HI#LO), BMUX-CMUX/DIRECT
13840 !COUNTER AND D SHOULD READ ALL ZERO, AFTER BEING ZAPPED IN TEST-534-F
13841 6752:
13842 TEST534G:
13843     PO,          LOAD-ENUA(ZTARGET402),          !SETUP FOR D=ZERO TEST
13844

```

```

13846 LOAD-ERROR(TEST534G), !ERROR DIRECTORY KEY
13847 DCS-CTR(C4.), !COMPARE AT TARGET
13848 NEXT, J/COUNTERS34G
(6752) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.000.100)
13849
13850 6404: !(FREE)
13851 COUNTERS34G:
13852 P2-1, D+COUNT#0(L0), !READ THE MUX INTO D
13853 NEXT, J/GOBUT534G
(6404) DCS(0.00.0.0.0.0) BM(1111..00.00..01.01..000..000...0.1.0..0..0...0.0011...0..0000.0...11.000...100.000.101)
13854
13855 6405: !(FREE)
13856 GOBUT534G:
13857 PO, BUMP-VERIFY, !COUNT
13858 SETUP, RETURN/SCOPE534G, !RETURN TO SCOPE LOOP TEST WORD
13859 NEXT, GOTO-PAGE(7), !BUT TABLE
13860 J/BUTD-IS-ZERO !GO TEST D IS ALL ZERO
(6405) DCS(0.00.0.0.0.1) BM(0110..00.10..00.00..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)
13861
13862
13863
13864
13865 6406: !(FREE)
13866 SCOPE534G:
13867 NEXT, BUTD(SCOPE), !NO ERROR: "TEST535A" (+1.WORDS)
13868 J/TEST535A ! ERROR: "LOADCNTR534E" (-11.WORDS)
(6406) DCS(0.00 0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.110.001)
13869
13870
13871
13872
13873 ! - - - - -
13874
13875 !*** TEST 535A ***
13876 !READ D THRU "4*D(C)AMUX<15:04>" PORT OF BMUX, AMUX-CMUX/DIRECT
13877 !IN(1)(122645), OUT(175132)
13878 6661:
13879 TEST535A:
13880 PO, LOAD-ENUA(ZTARGET402), !SETUP FOR D=ZERO TEST
13881 LOAD-ERROR(TEST535A), !ERROR DIRECTORY KEY
13882 DCS-CTR(C7.), !COMPARE AT TARGET
13883 BUMP-VERIFY, !COUNT
13884 NEXT, J/INITS35A
(6661) DCS(1.00.1.0.0.1) BM(1000..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...110.101.110)
13885
13886 6656:
13887 INITS35A:
13888 P3, CSPD(151)+EMIT, !GET INITIAL PATTERN FOR D
13889 EMIT/122645, !"1010 0101 1010 0101"
13890 NEXT, J/EXPEC535A
(6656) DCS(0.00.0.0.0.0) BM(1010..10.01..01.10..100..101...0.0.0..0..0...0.0010...1..0000.0...11.000...100.000.111)
13891
13892 6407: !(FREE)

```

```

13893 EXPECT535A:
13894     P3,          CSPD(14)+EMIT,          !GET EXPECTED PATTERN AFTER SHIFT
13895     NEXT,        EMIT/175132,          !"1111 1010 0101 1010"
13896     J/INIT0535A
(6407) DCS(0.00.0.0.0.0) BM(1111..10.10..10.01..011..010...0.0.0..0..0...0.0011...1..0000.0...11.000...100.001.000)

13897
13898 6410: !(FREE)
13899 INITD535A:
13900     P0,          BUMP-VERIFY,          !COUNT
13901     P2-T,        D+CSPD(DIS),          !INITIAL D=(122645)
13902     NEXT,        D(C)+ALUIS,          !SETUP D(C) FOR SHIFT = "1"
13903     J/COMP535A
(6410) DCS(0.00.0.0.0.1) BM(1010..10.00..00.00..000..100...0.1.0..0..0...0.0010...0..0000.0...11.000...100.001.001)

13904
13905 6411: !(FREE)
13906 COMP535A:
13907     SETUP,       D-RIGHT-4,          !AMUX/DIRECT, BMUX/RIGHT-4, CMUX/DIRECT
13908     P2-T,        D+D-SHIFTED-XOR-CSPB(B14), !COMPARE D-SHIFTED:EXPECTED, BITWISE
13909     NEXT,        J/GOBUT535A        !EXPECTED=(175132)
(6411) DCS(0.00.0.0.0.0) BM(0110..11.11..01.01..000..000...0.1.0..0..0...0.1000...0..0000.0...11.000...100.001.010)

13910
13911 6412: !(FREE)
13912 GOBUT535A:
13913     SETUP,       RETURN/TEST535B,    !RETURN TO START OF NEXT SUBTEEST
13914     NEXT,        GOTO-PAGE(7),      !BUT TABLE IS ON PAGE 7
13915     J/BUTD-15-ZERO,                !GO TEST D IS ALL ZERO
(6412) DCS(0.00.0.0.0.0) BM(0110..00.11..00.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

13916
13917
13918
13919
13920 ! - - - - -
13921
13922 !*** TEST 535B ***
13923 !READ D THRU "4*D(C)#AMUX<15:04>" PORT OF BMUX, AMUX-CMUX/DIRECT
13924 !IN(D)(055132), OUT (002645)
13925 6637:
13926 TEST535B:
13927     P0,          LOAD-ENVA(ZTARGET402), !SETUP FOR D=ZERO TEST
13928     LOAD-ERROR(TEST535B),          !ERROR DIRECTORY KEY
13929     DCS-CTR(C6.),                  !COMPARE AT TARGET
13930     BUMP-VERIFY,                    !COUNT
13931     NEXT,        J/EXPECT535B
(6637) DCS(1.00.1.0.0.1) BM(1001..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.001.011)

13932
13933 6413: !(FREE)
13934 EXPECT535B:
13935     P3,          CSPD(14)+EMIT,          !GET EXPECTED PATTERN AFTER SHIFT
13936     NEXT,        EMIT/002645,          !"0000 0101 1010 0101"
13937     J/INIT0535B
(6413) DCS(0.00.0.0.0.0) BM(0000..10.01..01.10..100..101...0.0.0..0..0...0.0011...1..0000.0...11.000...100.001.100)

13938
13939 6414: !(FREE)

```



```

13940 INITD535B:
13941     P2-T, D+NOT-CSPB(815),           !INITIAL D=(055132)
13942     D(C)+ALU15,           !SETUP D(C) FOR SHIFT = "0"
13943     NEXT J/COMP535B
(6414) DCS(0.00.0.0.0.0) BM(0111..11.10..11.01..101..100...0.1.0..0..0...0.0000...0..0000.0...11.000...100.001.101)

13944
13945 6415: !(FREE)
13946 COMP535B:
13947     SETUP, D-RIGHT-4           !AMUX/DIRECT, BMUX/RIGHT-4, CMUX/DIRECT
13948     P2-T, D+D-SHIFTED-XOR-CSPB(814), !COMPARE D-SHIFTED:EXPECTED, BITWISE
13949     NEXT J/GOBUT535B         !EXPECTED=(002645)
(6415) DCS(0.00.0.0.0.0) BM(0110..11.11..01.01..000..000...0.1.0..0..0...0.1000...0..0000.0...11.000...100.001.110)

13950
13951 6416: !(FREE)
13952 GOBUT535B:
13953     PO, BUMP-VERIFY           !COUNT
13954     SETUP, RETURN/SCOPE535B, !RETURN TO SCOPE LOOP TEST WORD
13955     NEXT, GOTO-PAGE(7),      !BUT TABLE IS ON PAGE 7
13956     J/BUTD-IS-ZERO         !GO TEST D IS ALL ZERO
(6416) DCS(0.00.0.0.0.1) BM(0110..00.10..00.01..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

13957
13958 6417: !(FREE)
13959 SCOPE535B:
13960     P3, CSPD(16)+EMIT, EMIT/114631, !CONSTANT FOR USE BELOW
13961     NEXT, BUTD(SCOPE),      !NO ERROR: "TEST536A" (+1.WORDS)
13962     J/TEST536A             ! ERROR: "INIT535A" (-10.WORDS)
(6417) DCS(0.00.0.1.0.0) BM(1001..10.10..01.10..011..001...0.0.0..0..0...0.0001...1..0000.0...11.000...110.101.111)

13963
13964
13965
13966
13967 ! - - - - -
13968
13969 !*** TEST 536A ***
13970 !READ D THRU "2*D(C)BMUX<15:02>" PORT OF CMUX, AMUX-BMUX/DIRECT
13971 !IN(1)(016161), OUT(143434)
13972 6657:
13973 TEST536A:
13974     PO, LOAD-EMUA(ZTARGET402), !SETUP FOR DD=ZERO TEST
13975     LOAD-ERROR(TE536A),       !ERROR DIRECTORY KEY
13976     DCS-CTR(C7.),           !COMPARE AT TARGET
13977     NEXT J/INIT536A
(6657) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...110.101.100)

13978
13979 6654:
13980 INIT536A:
13981     P3, CSPD(15)+EMIT,       !GET INITIAL PATTERN FOR D
13982     EMIT/016161,           !"0001 1100 0111 0001"
13983     NEXT J/EXPEC536A
(6654) DCS(0.00.0.0.0.0) BM(0001..10.11..00.01..110..001...0.0.0..0..0...0.0010...1..0000.0...11.000...100.010.000)

13984
13985 6420: !(FREE)
13986 EXPEC536A:

```

```

13987      PO,      BUMP-VERIFY,      !COUNT
13988      P3,      CSPD(14)+EMIT,      !GET EXPECTED PATTERN AFTER SHIFT
13989      NEXT,     EMIT/143434,      !"1100 0111 0001 1100"
13990      J/INITD536A
(6420) DCS(0.00.0.0.0.1) BM(1100..10.01..11.00..011..100...0.0.0..0..0...0.0011...1..0000.0...11.000...100.010.001)

13991
13992      6421:  !(FREE)
13993      INITD536A:
13994      P2-T,     D+CSPD(015),      !INITIAL D=(016161)
13995      D(C)+ALU00,      !SETUP D(C) FOR SHIFT = "1"
13996      NEXT,     J/COMPS36A
(6421) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..010...0.1.0..0..0...0.0010...0..0000.0...11.000...100.010.010)

13997
13998      6422:  !(FREE)
13999      COMPS36A:
14000      SETUP,    D-RIGHT-2,      !AMUX-BMUX/DIRECT, CMUX/RIGHT-2
14001      P2-T,     D+D-SHIFTED-XOR-CSPB(B(4)),      !COMPARE D-SHIFTED:EXPECTED, BITWISE
14002      NEXT,     J/GOBUT536A      !EXPECTED =(143434)
(6422) DCS(0.00.0.0.0.0) BM(0110..11.11..01.11..000..000...0.1.0..0..0...0.0000...0..0000.0...11.000...100.010.011)

14003
14004      6423:  !(FREE)
14005      GOBUT536A:
14006      SETUP,    RETURN/TEST536B,      !RETURN TO START OF NEXT SUBTEST
14007      NEXT,     GOTO-PAGE(7)      !BUT TABLE IS ON PAGE 7
14008      J/BUTD-IS-ZERO      !GO TEST D IS ALL ZERO
(6423) DCS(0.00.0.0.0.0) BM(0110..00.11..01.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

14009
14010
14011
14012
14013      ! - - - - -
14014
14015      !*** TEST 536B ***
14016      !READ D THRU "2*D(C)#BMUX<15:02>" PORT OF CMUX, AMUX-BMUX/DIRECT
14017      !IN(0)(161616), OUT(034343)
14018      6640:
14019      TEST536B:
14020      PO,      LOAD-ENUA(ZTARGET402),      !SETUP FOR D=ZERO TEST
14021      LOAD-ERROR(TEST536B),      !ERROR DIRECTORY KEY
14022      DCS-CTR(C6.),      !COMPARE AT TARGET
14023      NEXT,     J/EXPEC536B
(6640) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.010.100)

14024
14025      6424:  !(FREE)
14026      EXPEC536B:
14027      PO,      BUMP-VERIFY,      !COUNT
14028      P3,      CSPD(14)+EMIT,      !GET EXPECTED PATTERN AFTER SHIFT
14029      NEXT,     EMIT/034343,      !"1110 0011 1000 1110"
14030      J/INITD536B
(6424) DCS(0.00.0.0.0.1) BM(0011..10.10..00.11..100..011...0.0.0..0..0...0.0011...1..0000.0...11.000...100.010.101)

14031
14032      6425:  !(FREE)
14033      INITD536B:

```

```

14034 P2-T, D+NOT-CSPB(B15), !INITIAL D=(161616)
14035 D(C)+AL000, !SETUP D(C) FOR SHIFT = "0"
14036 NEXT, J/COMPS36B
(6425) DCS(0.00.0.0.0.0) BM(0111..11.10..11.01..101..010...0.1.0..0..0...0.0000...0..0000.0...11.000...100.010.110)

```

```

14037 6426: !(FREE)
14038 COMPS36B:
14039 PO, BUMP-VERIFY, !COUNT
14040 SETUP, D-RIGHT-2, !AMUX-BMUX/DIRECT, CMUX/RIGHT-2
14041 P2-T, D+D-SHIFTED-XOR-CSPB(B14), !COMPARE D-SHIFTED:EXPECTED, BITWISE
14042 NEXT, J/GOBUTS36B !EXPECTED=(034343)
14043 (6426) DCS(0.00.0.0.0.1) BM(0110..11.11..01.11..000..000...0.1.0..0..0...0.0000...0..0000.0...11.000...100.010.111)

```

```

14044 6427: !(FREE)
14045 GOBUTS36B:
14046 SETUP, RETURN/SCOPE536B, !RETURN TO SCOPE LOOP TEST WORD
14047 NEXT, GOTO-PAGE(7) !BUT TABLE IS ON PAGE 7
14048 J:3UTD-IS-ZERO !GO TEST D IS ALL ZERO
14049 (6427) DCS(0.00.0.0.0.0) BM(0110..00.10..00.11..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

```

```

14050 6430: !(FREE)
14051 SCOPE536B:
14052 PO, BUMP-VERIFY, !COUNT
14053 P3, CSPD(17)+EMIT, EMIT/031463, !CONSTANT FOR USE BELOW
14054 NEXT, BUTD(SCOPE), !NO ERROR: "TEST536C" (+1.WORDS)
14055 J/TEST536C !ERROR: "INIT536A" (-10.WORDS)
14056 (6430) DCS(0.00.0.1.0.1) BM(0011..10.00..11.00..110..011...0.0.0..0..0...0.0000...1..0000.0...11.000...110.101.101)

```

```

14057
14058
14059
14060
14061 ! - - - - -
14062

```

```

14063 !*** TEST 536C ***
14064 !READ D THRU "D(C)BMUX<15:01>" PORT OF CMUX, AMUX-BMUX/DIRECT
14065 !IN(1)(031463), OUT(114631)
14066 6655:
14067 TEST536C:
14068 PO, LOAD-ENUA(ZTARGET402), !SETUP FOR D=ZERO TEST
14069 LOAD-ERROR(TEST536C), !ERROR DIRECTORY KEY
14070 DCS-CTR(C5.), !COMPARE AT TARGET
14071 BUMP-VERIFY, !COUNT
14072 NEXT, J/INITD536C
(6655) DCS(1.00.1.0.0.1) BM(1010..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...110.101.010)

```

```

14073 6652:
14074 INITD536C:
14075 PO, BUMP-VERIFY, !COUNT
14076 P2-T, D+CSPD(D17), !INITIAL D=(031463)
14077 D(C)+ALU00, !SETUP D(C) FOR SHIFT = "1"
14078 NEXT, J/COMPS36B !"0011 0011 0011 0011"
14079 (6652) DCS(0.00.0.0.0.1) BM(1010..10.00..00.00..000..010...0.1.0..0..0...0.0000...0..0000.0...11.000...100.011.001)
14080

```

```

14081 6431: !(FREE)
14082 COMPS36C:
14083     SETUP, D-RIGHT-1,           !AMUX-BMUX/DIRECT, CMUX/RIGHT-1
14084     P2-T,  D+D-SHIFTED-XOR-CSPB(816), !COMPARE D-SHIFTED:EXPECTED, BITWISE
14085     NEXT,   J/GOBUT536C         !EXPECTED=(114631)
(6431) DCS(0.00.0.0.0.0) BM(0110..11.01..01.10..000..000...0.1.0..0..0...0.0000...0..0000.0...11.000...100.011.010)
                                     !"1001 1001 1001 1001"

14086
14087
14088 6432: !(FREE)
14089 GVBUT536C:
14090     SETUP, RETURN/TEST536D,     !RETURN TO START OF NEXT SUBTEST
14091     NEXT,  GOTO-PAGE(7),        !BUT TABLE IS ON PAGE 7
14092     J/BUTD-IS-ZERO              !GO TEST D IS ALL ZERO
(6432) DCS(0.00.0.C.0.0) BM(0110..00.11..01.00..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

14093
14094
14095
14096
14097 ! - - - - -
14098
14099 !*** TEST 536D ***
14100 !READ D THRU "D(C)BM...<15:01>" PORT OF CMUX, AMUX-BMUX/DIRECT
14101 !IN(0)(146314), OUT(063146)
14102 6641:
14103 TEST536D:
14104     PO,      LOAD-ENUR(ZTARGET402), !SETUP FOR D=ZERO TEST
14105     LOAD-ERROR(TEST536D),         !ERROR DIRECTORY KEY
14106     DCS-CTR(C7.),                 !COMPARE AT TARGET
14107     NEXT,    J/INIT536D
(6641) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.011.011)

14108
14109 6433: !(FREE)
14110 INIT536D:
14111     P3,      CSPD(15)+EMIT,        !GET INITIAL PATTERN FOR D
14112     EMIT/146314,                   !"1100 1100 1100 1100"
14113     NEXT,    J/EXPEC536D
(6433) DCS(0.00.0.0.0.0) BM(1100..10.11..00.11..001..100...0.0.0..0..0...0.0010...1..0000.0...11.000...100.011.100)

14114
14115 6434: !(FREE)
14116 EXPEC536D:
14117     P3,      CSPD(14)+EMIT,        !GET EXPECTED PATTERN AFTER SHIFT
14118     EMIT/063146,                   !"0110 0110 0110 0110"
14119     NEXT,    J/INITD536D
(6434) DCS(0.00.0.0.0.0) BM(0110..10.01..10.01..100..110...0.0.0..0..0...0.0011...1..0000.0...11.000...100.011.101)

14120
14121 6435: !(FREE)
14122 INITD536D:
14123     PO,      BUMP-VERIFY,          !COUNT
14124     P2-T,    D+CSPD(D15),         !INITIAL D=(146314)
14125     D(C)+0,  !SETUP D(C) FOR SHIFT = "0"
14126     NEXT,    J/COMP536D
(6435) DCS(0.00.0.0.0.1) BM(1010..10.00..00.00..000..000...0.1.0..0..0...0.0010...0..0000.0...11.000...100.011.110)

14127

```

```

14128 6436: !(FREE)
14129 COMPS360:
14130 SETUP, D-RIGHT-1, !AMUX-BMUX/DIRECT, CMUX/RIGHT-1
14131 P2-T, D+D-SHIFTED-XOR-CSPB(B14), !COMPARE D-SHIFTED:EXPECTED, BITWISE
14132 NEXT, J/GOBUT5360 !EXPECTED=(063146)
(6436) DCS(0.00.0.0.0.0) BM(0110..11.11..01.10..000..000...0.1.0..0..0...0.0000...0..0000.0...11.000...100.011.111)

14133 6437: !(FREE)
14134 GOBUT5360:
14135 SETUP, RETURN/SCOPE5360, !RETURN TO SCOPE LOOP TEST WORD
14136 NEXT, GOTO-PAGE(7), !BUT TABLE IS ON PAGE 7
14137 J/BUTD-IS-ZERO !GO TEST D IS ALL ZERO
(6437) DCS(0.00.0.0.0.0) BM(0110..00.10..01.00..000...111...J.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

14139 6440: !(FREE)
14140 SCOPE5360:
14141 NEXT, BUTD(SCOPE), !NO ERROR: "TEST536E" (+1.WORDS)
14142 J/TEST536E ! ERROR: "INIT536C" (-11.WORDS)
(6440) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.101.011)

14144 ! - - - - -
14145
14146
14147
14148 !
14149
14150 !*** TEST 536E ***
14151 !READ D THRU BMUX(14:00) SENDMUX(=SR15) PORT OF CMUX, AMUX-BMUX/DIRECT
14152 !IN(0)(146314), OUT(114631), SR=(100000)
14153 6653:
14154 TEST536E:
14155 PO, LOAD-ENVA(ZTARGET402), !SETUP FOR D=ZERO TEST
14156 LOAD-ERROR(TEST536E), !ERROR DIRECTORY KEY
14157 DCS-CTR(C6.), !COMPARE AT TARGET
14158 P3, BUTA(CLR-FLAG-RES-UCON), !SELECT SR-LOAD, SENDMUX PORTS 0123, BUSDIN+EMIT
14159 NEXT, J/LOADSR536E
(6653) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.010...110.101.000)

14160 6650:
14161 LOADSR536E:
14162 PO, BUMP-VERIFY, !COUNT
14163 P2-T, SR+BSPHI(C100000), ! (100000) IN SR(15:00)
14164 NEXT, J/INITD536E
(6650) DCS(0.00.0.0.0.1) BM(1010..01.11..00.00..001..000...0.0.1..0 0...0.0000...0..0000.0...11.000...100.100.001)

14166 6441: !(FREE)
14167 INITD536E:
14168 PO, BUMP-VERIFY, !COUNT
14169 P2-T, D+CSPD(015), !INITIAL D=(146314)
14170 D(C)+0, !SETUP D(C) FOR SHIFT = "0"
14171 NEXT, J/COMPS360
(6441) DCS(0.00.0.0.0.1) BM(1010..10.00..00.00..000..000...0.1.0..0..0...0.0010...0..0000.0...11.000...100.100.010)

14173 6442: !(FREE)
14174

```

```

14175 COMPS36E:
14176     SETUP, D-LEFT-1,          !AMUX-BMUX/DIRECT, CMUX/LEFT-1
14177                                     !BIT<00> = SR<15> = SENDMUX PORT 0 = (1)
14178     P2-T,  D+D-SHIFTED-XOR-CSPB(B16), !COMPARE D-SHIFTED:EXPECTED, BITWISE
14179     NEXT,   J/GOBUT536E        !EXPECTED=(114631)
(6442) DCS(0.00.0.0.0.0) BM(0110..11.01..01.00..000..000...0.1.0..0..0...0.0000...0..0000.0...11.000...100.100.0111)

14180
14181 6443: !(FREE)
14182 GOBUT536E:
14183     SETUP, RETURN/TEST536F,    !RETURN TO START OF NEXT SUBTEST
14184     NEXT,  GOTO-PAGE(7),       !BUT TABLE IS ON PAGE 7
14185     NEXT,  J/BUTD-IS-ZERO     !GO TEST D IS ALL ZERO
(6443) DCS(0.00.0.0.0.0) BM(0110..00.11..01.00..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.0011)

14186
14187
14188
14189
14190 ! - - - - -
14191
14192 !*** TEST 536F ***
14193 !READ D THRU BMUX(14:00) #SEDMUX(=SR15) PORT OF CMUX, AMUX-BMUX/DIRECT
14194 !IN(0)(031463), (063146), SR=(077777)
14195 6642:
14196 TEST536F:
14197     PD,      LOAD-ENVA(ZTARGET402), !SETUP FOR D=ZERO TEST
14198             LOAD-ERROR(TEST536F), !ERROR DIRECTORY KEY
14199             DCS-CTR(C6.),         !COMPARE AT TARGET
14200     NEXT,    J/LOADSRS36F
(6642) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.100.1001)

14201
14202 6444: !(FREE)
14203 LOADSRS36F:
14204     PD,      BUMP-VERIFY          !COUNT
14205     P2-T,    SR+NOT-ASPH(C100000), ! (077777) IN SR <15:00>
14206     NEXT,    J/INITD536F
(6444) DCS(0.00.0.0.0.1) BM(0000..00.00..11.01..001..000...0.0.1..0..0.. 0.0000...0..0000.0...11.000...100.100.1011)

14207
14208 6445: !(FREE)
14209 INITD536F:
14210     P2-T,    D+CSPD(D17),        !INITIAL D=(031463)
14211             D(C)+0,             !SETUP D(C) FOR SHIFT = "0"
14212     NEXT,    J/COMPS36F
(6445) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...0.1.0..0..0...0.0000...0..0000.0...11.000...100.100.1101)

14213
14214 6446: !(FREE)
14215 COMPS36F:
14216     SETUP, D-LEFT-1,          !AMUX-BMUX/DIRECT, CMUX/LEFT-1
14217                                     !BIT<00> = SR<15> = SENDMUX PORT 0 = (0)
14218     P2-T,    D+D-SHIFTED-XOR-CSPB(B14), !COMPARE D-SHIFTED:EXPECTED, BITWISE
14219     NEXT,    J/GOBUT536E        !EXPECTED = (063146)
(6446) DCS(0.00.0.0.0.0) BM(0110..11.11..01.00..000..000...0.1.0..0..0...0.0000...0..0000.0...11.000...100.100.1111)

14220
14221 6447: !(FREE)

```

```

14222 GOBUT536F:
14223     SETUP, RETURN/SCOPE536F,      !RETURN TO SCOPE LOOP TEST WORD
14224     NEXT,  GOTO-PAGE(7)          !BUT TABLE IS ON PAGE 7
14225     J/BUTD-IS-ZERO              !GO TEST D IS ALL ZERO
(6447) DCS(0.00.0.0.0.0) BM(0110..00.10..01.01..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

```

```

14226
14227 6450: !(FREE)
14228 SCOPE536F:
14229     P0, BUMP-VERIFY              !COUNT
14230     P3, CSPD(02)+EMIT           !RES-CON #1: (FOR USE BELOW)
14231     EMITC, SENDMUX-0123-SEL    !SELECT SENDMUX PORTS 0-3
14232     SR-LEFT GUARD-DIS,        !SR GOES LEFT, SR<00>+D(C)
14233     NEXT, BUTD(SCOPE)          !NO ERROR: "SETEMIT537A" (+1.WORDS)
14234     J/SETEMIT537A             !ERROR: "LOADSR536E" (-9.WORDS)
(6450) DCS(0.00.0.1.0.1) BM(0101..10.00..00.00..000..000...0.0.0..0..0...0.1101...1..0000.0...11.000...110.101.001)

```

```

14235
14236
14237
14238
14239 ! - - - - -
14240

```

```

14241 !*** TEST 537A ***
14242 !THIS TEST VALIDATES THE "SENDMUX" INPUTS TO THE SHIFT-TREE.
14243 !EACH 'SENDMUX' OUTPUT IS SET TO 1/0 VALUES, AND THEN READ OUT INTO THE 'SR' WHERE THEY
14244 !ARE ALL SAVED TO CHECK AT ONCE.
14245 6651:
14246 SETEMIT537A:
14247     P0, DCS-CTR(C15.),          !HOLD UP FOR NOW
14248     NEXT, GOTO-PAGE(4),        !XFER
14249     J/LOAD16537A
(6651) DCS(0.00.1.0.0.0) BM(0000..00.00..00.00..000..100...0.0.0..0..0...0.0000...0..0000.0...11.100...101.111.100)

```

```

14250
14251 4574:
14252 LOAD16537A:
14253     P3, CSPD(16)+EMIT,         !BIT03 ONLY SET
14254     EMIT/000010
14255     BUTA(CLR-FLAG-RES-UCON),   !BUSDIN <- EMIT, SR LOAD
14256     NEXT, J/LOAD14537A
(4574) DCS(0.00.0.0.0.0) BM(0000..10.00..00.00..001..000...0.0.0..0..0...0.0001...1..0000.0...11.010...010.111.110)

```

```

14257
14258 4276: !(FREE)
14259 LOAD14537A:
14260
14261     P3, CSPD(14)+EMIT,         !BIT11 ONLY SET
14262     EMIT/004000
14263     NEXT, J/LOAD03537A
(4276) DCS(0.00.0.0.0.0) BM(0000..10.10..00.00..000..000...0.0.0..0..0...0.0011...1..0000.0...11.000...010.111.111)

```

```

14264
14265 4277: !(FREE)
14266 LOAD03537A:
14267     P3, CSPD(03)+EMIT,
14268     EMIT/177252,
!WHAT THE 'SR' SHOULD BE
!AFTER THIS TEST

```

```

14269      NEXT      J/LOAD01537A
(4277) DCS[0.00.0.0.0.0] BM[1111..10.11..10.10..101..010...0.0.0..0.0...0.1100...1..0000.0...11.000...011.000.000]

14270      4300:  !(FREE)
14271      LOAD01537A:
14272      P3,      CSPD[011]+EMIT,
14273      EMITC,    SENDMUX-4567-SEL,
14274      SR-LEFT,  GUARD-DIS,
14275      NEXT,     GOTO-PAGE(7),
14276      J/INITSRS537A
14277      (4300) DCS[0.00.0.0.0.0] BM[0001..10.00..00.00..000..111...0.0.0..0.0...0.1110...1..0000.0...11.100...001.011.001]
!RES-CON #2:
!SELECT SENDMUX PORTS 4-7
!SR GOES LEFT, SR<00>+D[C]
!XFER

14278      7131:  !(FREE)
14279      INITSRS537A:
14280      P2-T,    SR+ALL-ONES,
14281      NEXT,     J/SETRESAS537A
14282      (7131) DCS[0.00.0.0.0.0] BM[1111..00.00..11.01..101..000...0.0.1..0.0...0.0000...0..0000.0...11.000...001.011.101]
!START 'SR' WITH ALL ONES

14283      7135:  !(FREE)
14284      SETRESAS537A:
14285      P2,      RES+CSPD(002),
14286      NEXT,     GOTO-PAGE(6),
14287      J/NEWCTRS537A
14288      (7135) DCS[0.00.0.0.0.0] BM[0000..10.00..00.00..000..110...0.0.0..0.0...0.1101...0..1000.1...11.100...110.100.011]
!LOAD RES W/ RES-CON#1 (SENDMUX-0123)
!XFER

14289      6643:
14290      NEWCTRS537A:
14291      PO,      LOAD-ENUA(SETRESB537A),
14292      LOAD-ERROR(NEWCTRS537A),
14293      DCS-CTR(C14.),
14294      NEXT,     J/AR3-537A
14295      (6643) DCS[1.00.1.0.0.0] BM[0001..00.11..10.11..111..000...0.0.0..0.0...0.0000...0..0000.0...11.000...100.101.001]
!COMPARE POINT #2
!ERROR DIRECTORY KEY
!COMPARE BELOW

14296      6451:  !(FREE)
14297      AR3-537A:
14298      PO,      BUMP-VERIFY,
14299      P2-T,    D+NOT-CSPD(016),
14300      NEXT,     GOTO-PAGE(7),
14301      J/BR3-537A
14302      (6451) DCS[0.00.0.0.0.1] BM[0111..10.00..11.01..101..111...0.1.0..0.0...0.0001...0..0000.0...11.100...001.011.100]
!COUNT
!D=(177767), BIT03="0"
!XFER

14303      7134:  !(FREE)
14304      BR3-537A:
14305      P2-T,    D+D-RIGHT-3,
14306      D[C]+ALU00,
14307      NEXT,     J/CR3-537A
14308      (7134) DCS[0.00.0.0.0.0] BM[1111..00.00..01.00..000..010...0.1.0..0.0...0.1000...0..0000.0...11.000...001.011.111]
!USE SENDMUX PORT 1 = AMUX03 = "0"
!INTO D[C] FOR SR

14309      7137:  !(FREE)
14310      CR3-537A:
14311      P2-T,    SR+SR-LEFT-1,
14312
!SENDMUX OUTPUT INTO SR<00>

```



KD11-K MICRO V00A-1 00:00:03 12-MAR-77

```

14313          D+CSPB(B16), SAVE-D(C),          !D=(000010), BIT03="1"
14314          NEXT      J/DR3-537A
(7137) DCS(0.00.0.0.0.0) BM(1010..11.01..00.00..000..111...0.1.1..0..0...0.0000...0..0000.0...11.000...001.100.000)

14315          7140: !(FREE)
14316          DR3-537A:
14317          P2-T,    D+D-RIGHT-3,          !USE SENDMUX PORT 1 = AMUX03 = "1"
14318          D(C)+ALU00,          !INTO D(C) FOR SR
14319          NEXT      J/AR7-537A
14320          (7140) DCS(0.00.0.0.0.0) BM(1111..00.00..01.00..000..010...0.1.0..0..0...0.1000...0..0000.0...11.000...001.100.001)

14321          7141: !(FREE)
14322          AR7-537A:
14323          P2-T,    SR+SR-LEFT-1,          !SENDMUX OUTPUT INTO SR<00>
14324          D+NOT-ASPHI(C000200), SAVE-D(C), !D=(177577), BIT07="0"
14325          NEXT      J/BR7-537A
14326          (7141) DCS(0.00.0.0.0.0) BM(0000..00.00..11.01..010..111...0.1.1..0..0...0.0000...0..0000.0...11.000...001.100.010)

14327          7142: !(FREE)
14328          BR7-537A:
14329          P2-T,    D+D-RIGHT-7,          !USE SENDMUX PORT 2 = D07 = "0"
14330          D(C)+ALU00,          !INTO D(C) FOR SR
14331          NEXT      J/CR7-537A
14332          (7142) DCS(0.00.0.0.0.0) BM(1111..00.00..01.00..000..010...0.1.0..0..0...0.0110...0..0000.0...11.000...001.100.011)

14333          7143: !(FREE)
14334          CR7-537A:
14335          P2-T,    SR+SR-LEFT-1,          !SENDMUX OUTPUT INTO SR<00>
14336          D+ASPHI(C000200), SAVE-D(C),    !D=(000200), BIT07="1"
14337          NEXT      J/DR7-537A
14338          (7143) DCS(0.00.0.0.0.0) BM(1111..00.00..11.01..010..111...0.1.1..0..0...0.0000...0..0000.0...11.000...001.100.100)

14339          7144: !(FREE)
14340          DR7-537A:
14341          P2-T,    D+D-RIGHT-7,          !USE SENDMUX PORT 2 = D07 = "1"
14342          D(C)+ALU00,          !INTO D(C) FOR SR
14343          NEXT      J/AR11-537A
14344          (7144) DCS(0.00.0.0.0.0) BM(1111..00.00..01.00..000..010...0.1.0..0..0...0.0110...0..0000.0...11.000...001.100.101)

14345          7145: !(FREE)
14346          AR11-537A:
14347          P2-T,    SR+SR-LEFT-1,          !SENDMUX OUTPUT INTO SR<00>
14348          D+NOT-CSPB(B14), SAVE-D(C),    !D=(173777), BIT11="0"
14349          NEXT      J/BR11-537A
14350          (7145) DCS(0.00.0.0.0.0) BM(0111..11.11..11.01..101..111...0.1.1..0..0...0.0000...0..0000.0...11.000...001.100.110)

14351          7146: !(FREE)
14352          BR11-537A:
14353          P2-T,    D+D-RIGHT-11,          !USE SENDMUX PORT 3 = AMUX03 = "0"
14354          D(C)+ALU00,          !INTO D(C) FOR SR
14355          NEXT      J/CR11-537A
14356          (7146) DCS(0.00.0.0.0.0) BM(1111..00.00..01.00..000..010...0.1.0..0..0...0.1110...0..0000.0...11.000...001.100.111)

14357          7147: !(FREE)
14358

```

```

14359 CR11-537A:
14360 P2-T, SR+SR-LEFT-1, !SENDMUX OUTPUT INTO SR<00>
14361 D+CSPB(B14), SAVE-D(C), !D=(004000), BIT11="1"
14362 NEXT, J/DR11-537A
(7147) DCS(0.00.0.0.0.0) BM(1010..11.11..00.00..000..111...0.1.1..0..0...0.0000...0..0000.0...11.000...001.101.000)

14363 7150: !(FREE)
14364 DR11-537A:
14365 P2-T, D+D-RIGHT-11, !USE SENDMUX PORT 3 = AMUX03 = "1"
14366 D(C)+ALU00, !INTO D(C) FOR SR
14367 NEXT, J/ER3-537A
14368 (7150) DCS(0.00.0.0.0.0) BM(1111..00.00..01.00..000..010...0.1.0..0..0...0.1110...0..0000.0...11.000...001.101.001)

14369 7151: !(FREE)
14370 ER3-537A:
14371 P2-T, SR+SR-LEFT-1, !SENDMUX OUTPUT INTO SR<00>
14372 D+NOT-CSPB(B16), SAVE-D(C), !D=(177767), BIT03="0"
14373 NEXT, J/SETRESB537A
14374 (7151) DCS(0.00.0.0.0.0) BM(0111..11.01..11.01..101..111...0.1.1..0..0...0.0000...0..0000.0...11.000...011.111.000)

14375 7370:
14376 SETRESB537A:
14377 P2, RES+CSPD(001), !LOAD RES W/RES-CON#2 (SENDMUX-4567)
14378 NEXT, J/FR3-537A
14379 (7370) DCS(0.00.0.0.0.0) BM(0000..10.00..00.00..000..000...0.0.0..0..0...0.1110...0..1000.1...11.000...001.101.010)

14380 7152: !(FREE)
14381 FR3-537A:
14382 P2-T, D+D-RIGHT-3, !USE SENDMUX PORT 5 = AMUX03 = "0"
14383 D(C)+ALU00, !INTO D(C) FOR SR
14384 NEXT, J/GR3-537A
14385 (7152) DCS(0.00.0.0.0.0) BM(1111..00.00..01.00..000..010...0.1.0..0..0...0.1000...0..0000.0...11.000...001.101.011)

14386 7153: !(FREE)
14387 GR3-537A:
14388 P2-T, SR+SR-LEFT-1, !SENDMUX OUTPUT INTO SR<00>
14389 D+CSPB(B16), SAVE-D(C), !D=(000010), BIT03="1"
14390 NEXT, J/HR3-537A
14391 (7153) DCS(0.00.0.0.0.0) BM(1010..11.01..00.00..000..111...0.1.1..0..0...0.0000...0..0000.0...11.000...001.101.100)

14392 7154: !(FREE)
14393 HR3-537A:
14394 P2-T, D+D-RIGHT-3, !USE SENDMUX PORT 5 = AMUX03 = "1"
14395 D(C)+ALU00, !INTO D(C) FOR SR
14396 NEXT, J/AL1-537A
14397 (7154) DCS(0.00.0.0.0.0) BM(1111..00.00..01.00..000..010...0.1.0..0..0...0.1000...0..0000.0...11.000...001.101.101)

14398 7155: !(FREE)
14399 AL1-537A:
14400 P2-T, SR+SR-LEFT-1, !SENDMUX OUTPUT INTO SR<00>
14401 D+ALL-ONES, !D=(177777)
14402 NEXT, GOTO-PAGE(4), !XFER
14403 J/BL1-537A
14404

```

```

(7155) DCS(0.00.0.0.0.0) BM(1111..00.00..11.01..101..100...0.1.1..0..0...0.0000...0..0000.0...11.100...010.100.010)
14405
14406 4242: !(FREE)
14407 BL1-537A:
14408 P2-T, D+D-LEFT-1, !USE SENDMUX PORT 4 = HARD "0"
14409 D(C)+ALU00, !INTC D(C) FOR SR
14410 NEXT, J/TEST537A
(4242) DCS(0.00.0.0.0.0) BM(1111..00.00..01.00..000..010...0.1.0..0..0...0.0000...0..0000.0...11.000...101.111.010)
14411
14412 4572:
14413 TEST537A:
14414 PO, LOAD-ENUA(ZTARGET402), !SETUP FOR D-IS-ZERO COMPARE
14415 LOAD-ERROR(TEST537A), !ERROR DIRECTORY KEY
14416 DCS-CTR(C4.), !COMPARE AT TARGET
14417 BUMP-VERIFY, !COUNT
14418 P2-T, SR+SR-LEFT-1, !SENDMUX OUTPUT INTO SR<00>
14419 NEXT, J/COMPS37A
(4572) DCS(1.00.1.0.0.1) BM(1011..00.11..11.00..000..010...0.0.1..0..0...0.0000...0..0000.0...11.000...011.000.010)
14420
14421 4302: !(FREE)
14422 COMPS37A:
14423 P2-T, D+SR-XOR-CSPD(D03), !COMPARE RECEIVED:EXPECTED BITWISE
14424 P3, BUTA(CLR-FLAG-RES-UCON), !RESET SR TO LOAD/GUARD-DIS
14425 NEXT, J/GOBUT537A
(4302) DCS(0.00.0.0.0.0) BM(0110..10.00..00.00..000..000...0.1.0..0..0...0.1100...0..0000.0...11.010...011.000.011)
14426
14427 4303: !(FREE)
14428 GOBUT537A:
14429 SETUP, RETURN/SCOPE537A, !RETURN TO SCOPE LOOP TEST WORD
14430 NEXT, GOTO-PAGE(7) !BUT TABLE IS ON PAGE 7
14431 J/BUTD-IS-ZERO !GO TEST D IS ALL ZERO
(4303) DCS(0.00.0.0.0.0) BM(0100..00.01..10.00..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)
14432
14433 4304: !(FREE)
14434 SCOPE537A:
14435 PO, BUMP-VERIFY, !COUNT
14436 BUSDIN+EMIT-[I], !RESET PROC UCON
14437 EN-CLK-IR[15-00],
14438 NEXT, BUTD(SCOPE), !NO ERROR: "TEST551A" (+1.WORD)
14439 J/TEST551A ! ERROR: "LOAD16537A" (-30.WORDS)
(4304) DCS(0.00.0.1.0.1) BM(0000..00.00..00.01..000..100...0.0.0..0..0...1.1001...0..0000.0...11.000...101.111.101)

```

```

14440
14441
14442
14443
14444
14445 !.PAGE=====
14446
14447
14448 .TOC * TEST551: BASE MACHINE DATAPATH COUNTER CAN COUNT
14449
14450
14451 !*****

```

```

14452 !*
14453 !* TESTS: 551 A - C                                UWORDS: 015 + 035
14454 !*
14455 !* FUNCTIONS:
14456 !*
14457 !* THE FOLLOWING THREE TESTS USE A COMMON SUBROUTINE TO TEST THE COUNTING
14458 !* ABILITY OF THE BASE MACHINE DATAPATH COUNTER; USING THE THREE ACTIVE
14459 !* BITS THAT ENABLE COUNTING. ADMITTEDLY THIS SEEMS LIKE OVERKILL, BUT
14460 !* THIS METHOD WAS THE LEAST EXPENSIVE IN TERMS OF NUMBER OF MICROWORDS
14461 !* USED FOR THE TESTING.
14462 !*
14463 !*****
14464
14465
14466 !THE FIRST TEST USES THE ACTIVE BUT(#13) "SR<1:0>#COUNT-IS-377" TO CHECK THE COUNTER
14467 4575:
14468 TEST551A:
14469     PO,          LOAD-ERROR(TEST551A),          !ERROR DIRECTORY KEY
14470                DCS-CTR(C15.),                !HOLD UP FOR NOW
14471                BUMP-VERIFY,                    !BUMP DCS COUNTER
14472     NEXT        J/LOADIN551A
14473 (4575) DCS(1.00.1.0.0.1) BM(0000..00.00..00.00..000..000..0..0.0.0..0..0..0.0000..0..0000.0...11.000...011.000.101)
14474
14475 4305: !(FREE)
14476 LOADIN551A:
14477     P3,          CSPD(17)+EMIT, EMIT/4..,      !INCREMENT FOR D, TO MATCH COUNTER
14478     NEXT        J/SETSR551A
14479 (4305) DCS(0.00.0.0.0.0) BM(0000..10.00..01.00..000..000..0..0.0.0..0..0..0.0000...1..0000.0...11.000...011.000.110)
14480
14481 4306: !(FREE)
14482 SETSR551A:
14483     PO,          BUMP-VERIFY,                  !COUNT
14484     P2-T,        SR+ZERO,                      !KEY IN SR<1:0>=(00) FOR SELECT THIS BUT
14485     NEXT        J/GOTEST551A
14486 (4306) DCS(0.00.0.0.0.1) BM(0011..00.00..00.00..000..000..0..0.0.1..0..0..0.0000...0..0000.0...11.000...101.110.100)
14487
14488 4564:
14489 GOTEST551A:
14490     SETUP,       RETURN/TEST551B,              !GO TO TEST SUBROUTINE
14491     NEXT        CALL(COUNT-TEST)               ! (SEE DESCRIP, FOLLOWING)
14492 (4564) DCS(0.00.0.0.0.0) BM(0100..00.10..11.11..110..100..0..0.0.0..0..0..0.0000...0..0000.0...11.100...011.000.001)
14493
14494 ! - - - - -
14495
14496 !THE SECOND TEST USES THE ACTIVE BUT(#25) "COUNT-IS-377" TO CHECK THE COUNTER
14497 4576:
14498 TEST551B:
14499     PO,          LOAD-ENUA(ZTARGET400),        !BIT<1:0> CLEAR
14500                LOAD-ERROR(TEST551B),        !ERROR DIRECTORY KEY
14501                DCS-CTR(C4.),                !COMPARE AT TARGET

```

```

14502         NEXT, BUTD(SCOPE),           !NO ERROR: "SETSR551B" (+1. WORDS)
14503         J/SETSR551B                   ! ERROR: "GOTEST551A" (-1. WORDS) REPEAT PREV TEST
(4576) DCS(1.00.1.1.0.0) BM(1011..00.11..11.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...101.110.101)

14504
14505         4565:
14506         SETSR551B:
14507         P2-T, SR+SR-PLUS-1,           !KEY IN SR<1:0>=(01) FOR SELECT THIS BUT
14508         NEXT, J/GOTEST551B
(4565) DCS(0.00.0.0.0.0) BM(1001..01.11..00.00..000..000...0.0.1..0..0...0.0000...0..0000.0...11.000...101.110.000)

14509
14510         4560:
14511         GOTEST551B:
14512         SETUP, RETURN/GOTEST551B1,     !RETURN TO START OF NEXT SUBTEST
14513         NEXT, GOTO-PAGE(7),           !BUT TABLE
14514         J/BUTCOUNT-IS-377             !FAKE A TEST
(4560) DCS(0.00.0.0.0.0) BM(0111..00.00..11.01..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.011.110)

14515
14516         7136: !(FREE)
14517         BUTCOUNT-IS-377:
14518         NEXT, BUT(COUNT-IS-377),       !SHOULD TARGET INTO *** ** *00
14519         J/ZTARGET400                 !BIT<1:0> BOTH CLEAR
(7136) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...10.101...100.000.000)

14520
14521
14522         ! - - - - -
14523
14524
14525         ! AND NOW TEST THAT THE ACTUAL BUT/COUNTER WORKS
14526
14527         7157: !(FREE)
14528         GOTEST551A1:
14529         S UP, RETURN/TEST551C,         !GO TO TEST SUBROUTINE
14530         NEXT, CALL(COUNT-TEST)        ! (SEE DESCRIP, FOLLOWING)
(7157) DCS(0.00.0.0.0.0) BM(0100..00.10..11.10..110..100...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.001)

14531
14532
14533
14534
14535         ! - - - - -
14536
14537         !THE THIRD TEST USES THE ACTIVE BUT(#17) "COUNT-IS-377#D(C)" TO CHECK THE COUNTER
14538         4566:
14539         TEST551C:
14540         PD, LOAD-ERROR(TEST551C),       !ERROR DIRECTORY KEY
14541         DCS-CTR(C15.),                 !HOLD UP FOR NOW
14542         NEXT, BUTD(SCOPE),           !NO ERROR: "SETSR551C" (+1. WORDS)
14543         J/SETSR551C                   ! ERROR: "GOTEST551B" (-3. WORDS) REPEAT PREV TEST
(4566) DCS(1.00.1.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...101.110.001)

14544
14545         4561:
14546         SETSR551C:
14547         P2-T, SR+SR-PLUS-1,           !KEY IN SR<1:0>=(10) FOR SELECT THIS BUT

```

```

14548 NEXT, GOTO-PAGE(5), !XFER
14549 J/GOTEST551C !
(4561) DCS(0.00.0.0.0.0) BM(1001..01.11..00.00..000..101...0.0.1..0..0...0.0000...0..0000.0...11.100...110.010.110)
14550
14551 5626:
14552 GOTEST551C:
14553 SETUP, RETURN/SCOPE551 !GO TO TEST SUBROUTINE
14554 NEXT, CALL(COUNT-TEST) ! (SEE DESCRIPTOR, FOLLOWING)
(5626) DCS(0.00.0.0.0.0) BM(0101..00.01..10.00..101..100...0.0.0..0..0...0.0000...0..0000.0...11.100...0!1.000.001)
14555
14556
14557
14558 5305: !(FREE)
14559 SCOPE551:
14560 P0, BUSDIN+EMIT-[I], !RESET PROC UCON
14561 EN-CLK-IR[15-00],
14562 NEXT, BUTD(SCOPE), !NO ERROR: "TEST610A1" (NEXT SECTION)
14563 J/TEST610A1 !ERROR: "GOTEST551C" (-1. WORDS) REPEAT PREV TEST
(5305) DCS(0.00.0.1.0.0) BM(0000..00.00..00.01..000..100...0.0.0..0..0...1.1001...0..0000.0...11.000...110.010.111)
14564
14565
14566
14567 ! -----
14568
14569 !*** B.M. COUNTER TESTING ROUTINE ***
14570
14571 ! THIS ROUTINE USES THE B.M. COUNTER TO COUNT FROM 000-377, WAITING FOR IT
14572 ! TO OVERFLOW AT THE RIGHT MOMENT. AT THE SAME TIME THE D REGISTER HAS BEEN
14573 ! TRACKING THE PROGRESS OF THE COUNTER, WAITING FOR IT TO OVERFLOW, COMPARING
14574 ! COUNT FOR COUNT THE INCREMENTED VALUES, AND GUARANTEEING WE WILL EXIT THE
14575 ! LOOP IF THE COUNTER SOMEHOW NEVER OVERFLOWS.
14576
14577
14578 4301: !(FREE)
14579 COUNTER01:
14580 P2, COUNTER+BSPHI(C000000), !LOAD COUNTER WITH (000) TO START
14581 NEXT, J/COUNTER02 !
(4301) DCS(0.00.0.0.0.0) BM(0000..01.11..00.00..100..000...0.0.0..0..0...0.0000...0..0010.1...11.000...011.001.000)
14582
14583 4310: !(FREE)
14584 COUNTER02:
14585 P2-T, D+ZERO, D[C]+ALUIS, !ZERO D, D[C]
14586 P3, A#BSPHI[17]+D, !ASPHI[17] WILL TRACK THE COUNTER
14587 NEXT, J/COUNTER03 !
(4310) DCS(0.00.0.0.0.0) BM(0011..00.11..00.01..011..100...0.1.0..0..0...0.0000...0..1011.0...11.000...101.010.110)
14588
14589
14590 !*** THE LOOP FOR THE COUNTER TEST ENTERS HERE ***
14591 4526:
14592 COUNTER03:
14593 P2-T, D+A-XOR-B, SAVE-D[C], !COMPARE COUNTER:TRACKER
14594 BUS-A, COUNT#D[L0], !=(COUNTER)#(000)

```

```

14595          BUS-B+BSPhi(R17),          !=(TRACKER)*(000)
14596      NEXT,  BUTR(D(C)-B),          !D(C) HOLDS THE CARRYOUT FROM THE TRACKING REGISTER
14597          J/COUNTER04                !IF (0), GOTO(COUNTER04), CONTINUE TESTING
14598          J/COUNTER04                !IF (1), GOTO(COUNTER11), ERROR, D OVERFLOW BEFORE COUNTER
(4526) DCS(0.00.0.0.0.0) BM(0110..01.11..01.01..011..111...0.1.0..0..0...0.0011...0..0000.0...10.011...101.100.101)

14599          !ENTER HERE FOR CONTINUE TEST
14600
14601      4545:
14602      COUNTER04:
14603          P0,          DCS-CTR(C5,)          !HOLD UP ON DCS COMPARE
14604          NEXT,  BUTR(D14-00-E0-0),          !COMPARE COUNTER:TRACKER, GENERATED ABOVE
14605          J/COUNTER12                !IF EQUAL, GOTO(COUNTER05), CONTINUE TEST
14606          J/COUNTER12                !IF NOT EQUAL, GOTO(COUNTER12), COUNTER*TRACKER ERROR
14607          J/COUNTER12                !IF NOT EQUAL, GOTO(COUNTER12), COUNTER*TRACKER ERROR
(4545) DCS(0.00.1.0.0.0) BM(1010..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...01.101...101.101.101)

14608          !ENTER HERE FOR CONTINUE TEST
14609
14610      4557:
14611      COUNTER05:
14612          P2-T,  D+ZERO, D(C)+ALU15,          !TRY TO FORCE EXIT
14613          NEXT,  BUTR(CASE),          !DETERMINE WHO TO TEST:
14614          J/COUNTER06                !SR=0 -> COUNTER06
14615          J/COUNTER06                !SR=1 -> COUNTER07
14616          J/COUNTER06                !SR=2 -> COUNTER08 (SR=3 NOT USED)
14617          J/COUNTER06                !SR=2 -> COUNTER08 (SR=3 NOT USED)
(4557) DCS(0.00.0.0.0.0) BM(0011..00.00..00.00..000..100...0.1.0..0..0...0.0000...0..0000.0...00.000...101.011.100)

14618          !ENTER HERE FOR SR=0
14619      4534:
14620      COUNTER06:
14621          P2-T,  D+A-PLUS-B, D(C)+COUT15,          !INCREMENT TRACKING REGISTER
14622          BUS-A+ASPhi(R17),          !CONSTANT (400)
14623          BUS-B+CSPD(D17),          !SAVE NEW
14624          P3,          A#BSPhi(17)+D,          !IF SET, GOTO(COUNTER10), B.M. COUNTER OVERFLOWED
14625          NEXT,  BUTA(SR1-0#COUNT-IS-377),          !IF CLEAR, GOTO(COUNTER03), NEXT PASS THRU TEST
14626          J/COUNTER03                !IF CLEAR, GOTO(COUNTER03), NEXT PASS THRU TEST
14627          J/COUNTER03                !IF CLEAR, GOTO(COUNTER03), NEXT PASS THRU TEST
(4534) DCS(0.00.0.0.0.0) BM(1001..10.11..11.01..011..110...0.1.0..0..0...0.0000...0..1011.0...01.011...101.010.110)

14628          !ENTER HERE FOR SR=1
14629      4535:
14630      COUNTER07:
14631          P2-T,  D+A-PLUS-B, D(C)+COUT15,          !INCREMENT TRACKING REGISTER
14632          BUS-A+ASPhi(R17),          !CONSTANT (400)
14633          BUS-B+CSPD(D17),          !SAVE NEW
14634          P3,          A#BSPhi(17)+D,          !IF SET, GOTO(COUNTER10), B.M. COUNTER OVERFLOWED
14635          NEXT,  BUTA(COUNT-IS-377),          !IF CLEAR, GOTO(COUNTER03), NEXT PASS THRU TEST
14636          J/COUNTER03                !IF CLEAR, GOTO(COUNTER03), NEXT PASS THRU TEST
14637          J/COUNTER03                !IF CLEAR, GOTO(COUNTER03), NEXT PASS THRU TEST
(4535) DCS(0.00.0.0.0.0) BM(1001..10.11..11.01..011..110...0.1.0..0..0...0.0000...0..1011.0...10.101...101.010.110)

14638          !ENTER HERE FOR SR=2
14639      4536:
14640      COUNTER08:
14641          P2-T,  D+A-PLUS-B, D(C)+COUT15,          !INCREMENT TRACKING REGISTER
14642          BUS-A+ASPhi(R17),          !
14643          J/COUNTER03                !

```

```

14644          BUS-B+CSPD(D17),          !CONSTANT (400)
14645          P3,      A#BSPHI(17)+D,    !SAVE NEW
14646          NEXT,    BUTA(COUNT-IS-377#D(C)), !IF SET, GOTO(COUNTER10), B.M. COUNTER OVERFLOWED
14647          J/COUNTER09                    !IF CLEAR, GOTO(COUNTER09), NEXT PASS THRU TEST
(4536) DCS(0.00.0.0.0.0) BM(1001..10.11..11.01..011..110...0.1.0..0..0...0.0000...0..1011.0...01.111...101.010.101)

14648
14649
14650          !INTERMEDIATE WORD FOR NEXT PHSS
14651          4525:
14652          COUNTER09:
14653          NEXT,      J/COUNTER03                    !NEXT PASS
(4525) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...101.010.110)

14654
14655
14656
14657          ! - - - - -
14658
14659          !ENTER THIS SECTION WHEN HIT AN END CONDITION:
14660
14661          !ENTER HERE WHEN COUNTER SIGNALS IT HAS OVERFLOWN:
14662          4527:
14663          COUNTER10:
14664          PO,      DCS-CTR(C15.),          !HOLD UP
14665          NEXT,    BUTR(D(C)-B),          !NOW TEST THAT D HAS ALSO OVERFLOWED, AT THE SAME TIME
14666          J/COUNTER10B                    !IF TRUE, GOTO(COUNTER10A), ALL IS OK, END THE TEST
14667          (4527) DCS(0.00.1.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...10.011...101.011.001)
!IF FALSE, GOTO(COUNTER10B), ERROR, COUNTER OVERFLOWED TOO SOON

14668
14669          4533:
14670          COUNTER10A:
14671          PO,      DCS-CTR(C15.),          !ALL, END THE TEST WITH NO ERRORS
14672          NEXT,    BUTA(RETURN),          !
14673          J/BUTERROR4                    !
(4533) DCS(0.00.1.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)

14674
14675          4531:
14676          COUNTER10B:
14677          PO,      DCS-CTR(C0.),          !FORCE ERROR NOW,
14678          NEXT,    BUTA(RETURN),          ! B.M. COUNTER OVERFLOWED TOO SOON
14679          J/BUTERROR4                    !
(4531) DCS(0.00.1.0.0.0) BM(1111..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)

14680
14681          !ENTER HERE WHEN WE GET A "D OVERFLOWED BEFORE COUNTER" CONDITION
14682          4547:
14683          COUNTER11:
14684          PO,      DCS-CTR(C0.),          !FORCE ERROR NOW,
14685          NEXT,    BUTA(RETURN),          ! B.M. COUNTER DIDN'T OVERFLOW ON TIME
14686          J/BUTERROR4                    !
(4547) DCS(0.00.1.0.0.0) BM(1111..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)

14687
14688          !ENTER HERE WHEN WE GET A "TRACKER : COUNTER" MISCOMPARE
14689          4555:
14690          COUNTER12:
14691          PO,      DCS-CTR(C0.),          !FORCE ERROR NOW,

```



14692 NEXT, BUTA(RETURN), ; B.M. COUNTER DIDN'T INCREMENT IN STEP WITH TRACKER  
 14693 J/BUTERROR4 ;  
 (4555) DCS(0.00.1.0.0.0) BM(1111..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)

14694  
14695  
14696  
14697  
14698  
14699  
14700  
14701  
14702  
14703  
14704  
14705  
14706  
14707  
14708  
14709  
14710  
14711  
14712  
14713  
14714  
14715  
14716  
14717  
14718  
14719  
14720  
14721  
14722  
14723  
14724  
14725  
14726  
14727  
14728  
14729  
14730  
14731  
14732  
14733  
14734  
14735  
14736  
14737  
14738  
14739  
14740  
14741  
14742  
14743  
14744

! .PAGE=====

.TOC \* TEST610: CONDITION CODE LOGIC

\*\*\*\*\*

TESTS: 610 A - D UMWORDS: 074 + 070

FUNCTIONS:

THE FOLLOWING EIGHT TESTS EXERCISE THE PSH CC LOGIC (THE NZVC BITS)  
 AND THE ASSOCIATED ROMS/MUXES ETC. TO VERIFY THE ABSCENCE OF STUCK  
 ONE/ZERO CONDITIONS ON ALL LOGIC LINES.

\*\*\*\*\*

SUMMARY OF CC-LOGIC TESTS:

TEST NUMB	ROM ADR	IR DATA	CC CLASS	BYTE-H	D(C)/D-REG	A-SIDE	B-SIDE	MODIFY VBIT-H	PREV NZVC	GENERATED NZVC
A1	665	105200	010	1	1-100000	177700	100100	1	1010	0101
A2	145	105200	010	1	0-100000	077600	000200	0	1001	0110
B1	132	105300	101	1	1-000200	100000	000200	0	0100	1011
B2	253	005300	101	0	0-000000	100200	000200	1	1001	0110
C1	437	072000	111	0	1-000000	000200	000200	0	1010	0111
C2	037	072000	111	0	1-077776	037777	037777	0	1100	0001
D1	216	072000	111	0	0-177400	077600	077600	1	0101	1000
D2	116	072000	111	0	0-100000	000200	100200	0	0101	1000

! - - - - -  
 !\*\*\* TEST 610A1 \*\*\*

5627:  
 TEST610A1:

```

14745      PO,      LOAD-ENVA(ZTARGET405),      !NZVC AFTER = "0101"
14746      LOAD-ERROR(TEST610A1),      !ERROR DIRECTORY KEY
14747      DCS-CTR(C15.),      !COMPARE AT TARGET
14748      BUMP-VERIFY,      !COUNT
14749      NEXT,      J/LOADIR610A1
(5627) DCS(1.00.1.0.0.1) BM(0000..00.11..11.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...101.011.010)

14750      5532:
14751      LOADIR610A1:
14752      PO,      BUMP-VERIFY,      !COUNT
14753      P2-U,      IR+EMIT, EMIT/105200,      !(105200)=INCB, CC-CLASS="010", BYTE-H
14754      NEXT,      J/LOAD01-610A1
14755      (5532) DCS(0.00.0.0.0.0.1) BM(1000..00.10..10.10..000..000...0.0.0..0..0...1.1010...0..0000.0...11.000...011.000.110)

14756      5306: !(FREE)
14757      LOAD01-610A1:
14758      PO,      BUMP-VERIFY,      !COUNT
14759      P3,      CSPD(01)+EMIT, EMIT/100012,      !FOR D(C)=(1), PS(NZVC)="1010" PREVIOUSLY
14760      NEXT,      J/LOAD05-610A1
14761      (5306) DCS(0.00.0.0.0.0.1) BM(1000..10.00..00.00..001..010...0.0.0..0..0...0.1110...1..0000.0...11.000...011.000.111)

14762      5307: !(FREE)
14763      LOAD05-610A1:
14764      P3,      CSPD(05)+EMIT, EMIT/177700,      !A-SIDE DATA
14765      NEXT,      J/LOAD06-610A1
14766      (5307) DCS(0.00.0.0.0.0.0) BM(1111..10.11..11.11..000..000...0.0.0..0..0...0.1010...1..0000.0...11.000...011.001.000)

14767      5310: !(FREE)
14768      LOAD06-610A1:
14769      P3,      CSPD(06)+EMIT, EMIT/100100,      !B-SIDE DATA
14770      NEXT,      J/PSCC-DC610A1
14771      (5310) DCS(0.00.0.0.0.0.0) BM(1100..10.00..00.01..000..000...0.0.0..0..0...0.1001...1..0000.0...11.000...011.001.001)

14772      5311: !(FREE)
14773      PSCC-DC610A1:
14774      SETUP,      RETURN/SETBUS610A1,      !EXEC SUBR WHICH:
14775      NEXT,      CALL(SETUPPSCC#DC)      !(1) CSP(10)<3:0> -> PS(NZVC)
14776      (5311) DCS(0.00.0.0.0.0.0) BM(0101..00.01..10.01..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.101.110)
      !(2) CSP(10)<15> -> D(C)

14778      5312: !(FREE)
14779      SETBUS610A1:
14780      P2-T,      SR+CSPD(D05),      !GET CONSTANT FOR A-SIDE WHEN CC SET
14781      NEXT,      J/DOIT610A1
14782      (5312) DCS(0.00.0.0.0.0.0) BM(1010..10.00..00.00..000..000...0.0.1..0..0...0.1010...0..0000.0...11.000...011.001.011)

14783      5313: !(FREE)
14784      DOIT610A1:
14785      SETUP,      SET-CC,      !FOR CLOCKING CC-S IN NEXT UWORD
14786      MODIFY-VBIT,      !EXTRA CC ROM INPUT
14787      P2-T,      D+A-PLUS-B, SAVE-D(C),      !D=(100000), D(C)=(1)
14788      BUS-A+SR,      !A=(177700)
14789      BUS-B+CSPD(D06),      !B=(100100)
14790

```

```

14791      NEXT      J/GETIT610A1
(5313) DCS(0.00.0.0.0.0) BM(1001..10.00..00.10..000..111...0.1.0..0..1...0.1001...0..0000.0...11.000...011.001.100)
14792
14793      5314: !(FREE)
14794      GETIT610A1:
14795          P2-T,    CLK-CC,                                !PS(NZVC) GENERATED ABOVE LATCHED HERE
14796          SETUP,  RETURN/TEST610A2,                       !EXEC SUBR WHICH:
14797          NEXT,   CALL(PSCCTOSR3-0)                         ! PS<3:0> -> SR<3:0> J/BUT(SR3-0)
(5314) DCS(0.00.0.0.0.0) BM(0101..00.11..00.10..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.110.010)
14798
14799
14800
14801      ! - - - - -
14802
14803      !*** TEST 610A2 ***
14804
14805      5622:
14806      TEST610A2:
14807          PD,      LOAD-ENVA(ZTARGET406),                   !NZVC AFTER = "0110"
14808          LOAD-ERROR(TEST610A2),                             !ERROR DIRECTORY KEY
14809          DCS-CTR(C13.),                                     !COMPARE AT TARGET
14810          NEXT,    J/LOAD06-610A2
(5622) DCS(1.00.1.0.0.0) BM(0010..00.11..11.00..000..110...0.0.0..0..0...0.0000...0..0000.0...11.000...011.001.101)
14811
14812      5315: !(FREE)
14813      LOAD06-610A2:
14814          PD,      BUMP-VERIFY,                               !COUNT
14815          P3,      CSPD(06)+EMIT, EMIT/077600,              !A-SIDE DATA
14816          NEXT,    J/LOAD01-610A2
(5315) DCS(0.00.0.0.0.1) BM(0111..10.11..11.10..000..000...0.0.0..0..0...0.1001...1..0000.0...11.000...011.001.110)
14817
14818      5316: !(FREE)
14819      LOAD01-610A2:
14820          P3,      CSPD(01)+EMIT, EMIT/000011,              !FOR D(C)=(0), PS(NZVC)="1001" PREVIOUSLY
14821          NEXT,    J/PSCC-DC610A2
(5316) DCS(0.00.0.0.0.0) BM(0000..10.00..00.00..001..001...0.0.0..0..0...0.1110...1..0000.0...11.000...011.001.111)
14822
14823      5317: !(FREE)
14824      PSCC-DC610A2:
14825          SETUP,  RETURN/SETBUS610A2,                       !EXEC SUBR WHICH:
14826          NEXT,   CALL(SETUPPSCC#DC)                         ! (1) CSP(10)<3:0> -> PS(NZVC)
14827          ! (2) CSP(10)<15> -> D(C)
(5317) DCS(0.00.0.0.0.0) BM(0101..00.01..10.10..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.101.110)
14828
14829      5320: !(FREE)
14830      SETBUS610A2:
14831          P2-T,    SR=CSPD(006),                             !GET CONSTANT FOR A-SIDE WHEN CC SET
14832          NEXT,   J/DOIT610A2
(5320) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...0.0.1..0..0...0.1001...0..0000.0...11.000...011.010.001)
14833
14834      5321: !(FREE)
14835      DOIT610A2:

```

```

14836 SETUP, SET-CC, !FOR CLOCKING CC-S IN NEXT UWORD
14837 NOT-MODIFY-VBIT, !EXTRA CC ROM INPUT
14838 P2-T, D+A-PLUS-B, SAVE-D(C), !D=(100000), D(C)=(0)
14839 BUS-A+SR, !A=(077600)
14840 BUS-B+BSPHI(C000200), !B=(000200)
14841 NEXT, J/GETIT610A2
(5321) DCS(0.00.0.0.0.0) BM(1001..01.11..00.00..010..111...0.1.0..0..1...0.0000...0..0000.0...11.000...011.010.010)

```

```

14842 5322: !(FREE)
14843 GETIT610A2:
14844 P2-T, CLK-CC, !PS(NZVC) GENERATED ABOVE LATCHED HERE
14845 SETUP, RETURN/SCOPE610A, !EXEC SUBR WHICH:
14846 NEXT, CALL(PSCCTOSR3-0) ! PS<3:0> -> SR<3:0>, J/BUT(SR3-0)
14847 (5322) DCS(0.00.0.0.0.0) BM(0101..00.01..10.10..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.110.010)

```

```

14848 5323: !(FREE)
14849 SCOPE610A:
14850 PO, BUMP-VERIFY, !COUNT
14851 NEXT, BUTD(SCOPE), !NO ERROR: "TEST610B1" (+1. WORDS)
14852 J/TEST610B1 ! ERROR: "LOADIR610A1" (-15. WORDS)
14853 (5323) DCS(0.00.0.1.0.1) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...101.011.011)

```

```

14856
14857
14858
14859
14860
14861
14862
14863
14864 ! - - - - -
14865

```

```

14866 !*** TEST 610B1 ***
14867
14868 5533:
14869 TEST610B1:
14870 PO, LOAD-ENVA(ZTARGET413), !NZVC AFTER = "1011"
14871 LOAD-ERROR(TEST610B1), !ERROR DIRECTORY KEY
14872 DCS-CTR(C13.), !COMPARE AT TARGET
14873 NEXT, J/LOADIR610B1
(5533) DCS(1.00.1.0.0.0) BM(0010..00.11..11.00..001..011...0.0.0..0..0...0.0000...0..0000.0...11.000...101.011.100)

```

```

14874 5534:
14875 LOADIR610B1:
14876 PO, BUMP-VERIFY, !COUNT
14877 P2-U, IR+EMIT, EMIT/105300, !(105300)=DECB, CC-CLASS="101", BYTE-H
14878 NEXT, J/LOAD01-610B1
14879 (5534) DCS(0.00.0.0.0.1) BM(1000..00.10..10.11..000..000...0.0.0..0..0...1.1010...0..0000.0...11.000...011.010.100)

```

```

14880 5324: !(FREE)
14881 LOAD01-610B1:
14882 P3, CSPD(01)+EMIT, EMIT/100004, !FOR D(C)=(1), PS(NZVC)="0100" PREVIOUSLY
14883

```

```

14884      NEXT      J/PSCC-DC61081
(5324) DCS(0.00.0.0.0.0) BM(1000..10.00..00.00..000..100...0.0.0..0..0...0.1110...1..0000.0...11.000...011.010.101)
14885
14886      5325:  !(FREE)
14887      PSCC-DC61081:
14888          SETUP,  RETURN/SETBUSA61081,          !EXEC SUBR WHICH:
14889          !      (1) CSP(10)<3:0> -> PS(NZVC)
14890      NEXT      CALL(SETUPPSCC#DC)          !      (2) CSP(10)<15> -> D(C)
(5325) DCS(0.00.0.0.0.0) BM(0101..00.01..10.10..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.101.110)
14891
14892      5326:  !(FREE)
14893      SETBUSA61081:
14894          P2-T,   SR+BSPHI(C100000),          !GET CONSTANT FOR A-SIDE WHEN CC SET
14895      NEXT      J/DOIT61081
(5326) DCS(0.00.0.0.0.0) BM(1010..01.11..00.00..001..000...0.0.1..0..0...0.0000...0..0000.0...11.000...011.010.111)
14896
14897      5327:  !(FREE)
14898      DOIT61081:
14899          SETUP,  SET-CC,          !FOR CLOCKING CC-S IN NEXT UWORD
14900          NOT-MODIFY-VBIT,        !EXTRA CC ROM INPUT
14901          P2-T,   D+NOT-A-AND-B, SAVE-D(C),    !D=(000200), D(C)=(1)
14902          BUS-A+SR,                !A=(100000)
14903          BUS-B+BSPHI(C000200),    !B=(000200)
14904      NEXT      J/GETIT61081
(5327) DCS(0.00.0.0.0.0) BM(0010..01.11..00.00..010..111...0.1.0..0..1...0.0000...0..0000.0...11.000...011.011.000)
14905
14906      5330:  !(FREE)
14907      GETIT61081:
14908          P2-T,   CLK-CC,          !PS(NZVC) GENERATED ABOVE LATCHED HERE
14909          SETUP,  RETURN/TEST61082,        !EXEC SUBR WHICH:
14910      NEXT      CALL(PSCCTOSR3-0)        ! PS<3:0> -> SR<3:0>, J/BUT(SR3-0)
(5330) DCS(0.00.0.0.0.0) BM(0101..00.11..00.01..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.110.010)
14911
14912
14913
14914      ! - - - - -
14915
14916      !*** TEST 61082 ***
14917
14918      5612:
14919      TEST61082:
14920          PO,   LOAD-ENVA(ZTARGET406),        !NZVC AFTER = "0110"
14921          LOAD-ERROR(TEST61082),            !ERROR DIRECTORY KEY
14922          DCS-CTR(C14.),                    !COMPARE AT TARGET
14923      NEXT      J/LOADIR61082
(5612) DCS(1.00.1.0.0.0) BM(0001..00.11..11.00..000..110...0.0.0..0..0...0.0000...0..0000.0...11.000...011.011.001)
14924
14925      5331:  !(FREE)
14926      LOADIR61082:
14927          PO,   BUMP-VERIFY,          !COUNT
14928          P2-U,  IR+EMIT, EMIT/005300,    ! (005300)=DEC, CC-CLASS="101", NOT-BYTE-H
14929      NEXT      J/LOAD007-61082

```

```

(5331) DCS(0.00.0.0.0.1) BM(0000..00.10..10.11..000..000...0.0.0..0..0...1.1010...0..0000.0...f1.000...011.011.010)
14930
14931 5332: !(FREE)
14932 LOAD07-610B2:
14933 P3, CSPD(07)+EMIT, EMIT/100200, !A-SIDE DATA
14934 NEXT, J/LOAD01-610B2
(5332) DCS(0.00.0.0.0.0) BM(1000..10.00..00.10..000..000...0.0.0..0..0...0.1000...1..0000.0...11.000...011.011.011)
14935
14936 5333: !(FREE)
14937 LOAD01-610B2:
14938 P3, CSPD(01)+EMIT, EMIT/000011, !FOR D(C)=(0), PS(NZVC)="1001" PREVIOUSLY
14939 NEXT, J/PSCC-DC610B2
(5333) DCS(0.00.0.0.0.0) BM(0000..10.00..00.00..001..001...0.0.0..0..0...0.1110...1..0000.0...11.000...011.011.100)
14940
14941 5334: !(FREE)
14942 PSCC-DC610B2:
14943 SETUP, RETURN/SETBUS610B2, !EXEC SUBR WHICH:
14944 NEXT, CALL[SETUPPSCC#DC] ! (1) CSP(10)<3:0> -> PS(NZVC)
14945 ! (2) CSP(10)<15> -> D(C)
(5334) DCS(0.00.0.0.0.0) BM(0101..00.01..10.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.101.110)
14946
14947 5335: !(FREE)
14948 SETBUS610B2:
14949 P2-T, SR+CSPD(007), !GET CONSTANT FOR A-SIDE WHEN CC SET
14950 NEXT, J/DOIT610B2
(5335) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...0.0.1..0..0...0.1000...0..0000.0...11.000...011.011.110)
14951
14952 5336: !(FREE)
14953 DOIT610B2:
14954 SETUP, SET-CC, !FOR CLOCKING CC-S IN NEXT UWORD
14955 MODIFY-VBIT, !EXTRA CC ROM INPUT
14956 P2-T, D+NOT-A-AND-B, SAVE-D(C), !D=(000000), D(C)=(0)
14957 BUS-A+SR, !A=(100200)
14958 BUS-B+BSPHI(C000200), !B=(000200)
14959 NEXT, J/GETIT610B2
(5336) DCS(0.00.0.0.0.0) BM(0010..01.11..00.10..010..111...0.1.0..0..1...0.0000...0..0000.0...11.000...011.011.111)
14960
14961 5337: !(FREE)
14962 GETIT610B2:
14963 P2-T, CLK-CC, !PS(NZVC) GENERATED ABOVE LATCHED HERE
14964 SETUP, RETURN/SCOPE610B, !EXEC SUBR WHICH:
14965 NEXT, CALL[PSCCTOSR3-0] ! PS<3:0> -> SR<3:0> J/BUT(SR3-0)
(5337) DCS(0.00.0.0.0.0) BM(0101..00.01..11.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.110.010)
14966
14967
14968 5340: !(FREE)
14969 SCOPE610B:
14970 PO, BUMP-VERIFY, !COUNT
14971 NEXT, BUTD(SCOPE), !NO ERROR: "TEST610C1" (+1. WORDS)
14972 J/TEST610C1, ! ERROR: "LOADIR610B1" (-14. WORDS)
14973
(5340) DCS(0.00.0.1.0.1) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...101.011.101)

```

14974  
14975  
14976  
14977  
14978  
14979  
14980  
14981  
14982  
14983  
14984  
14985  
14986  
14987  
14988  
14989  
14990  
14991  
14992  
14993  
14994  
14995  
14996  
14997  
14998  
14999  
15000  
15001  
15002  
15003  
15004  
15005  
15006  
15007  
15008  
15009  
15010  
15011  
15012  
15013  
15014  
15015  
15016  
15017  
15018  
15019  
15020  
15021

! - - - - -

!\*\*\* TEST 610C1 \*\*\*

```

5535:
TEST610C1:
    PO,          LOAD-ENUA(ZTARGET407),          !NZVC AFTER = "0111"
                LOAD-ERROR(TEST610C1),         !ERROR DIRECTORY KEY
                DCS-CTR(C13.),                 !COMPARE AT TARGET
    NEXT,        J/LOADIR610C1
(5535) DCS(1.00.1.0.0.0) BM(0010..00.11..11.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.000...101.100.100)

```

```

5544:
LOADIR610C1:
    PO,          BUMP-VERIFY,                   !COUNT
                IR+EMIT, EMIT/072000,          !(072000)=ASH, CC-CLASS="111", NOT-BYTE-H
    NEXT,        J/LOAD01-610C1
(5544) DCS(0.00.0.0.0.1) BM(0111..00.01..00.00..000..000...0.0.0..0..0...1.1010...0..0000.0...11.000...011.100.001)

```

```

5341: !(FREE)
LOAD01-610C1:
    P3,          CSPD(01)+EMIT, EMIT/100012,    !FOR D(C)=(1), PS(NZVC)="1010" PREVIOUSLY
    NEXT,        J/PSCC-DC610C1
(5341) DCS(0.00.0.0.0.0) BM(1000..10.00..00.00..001..010...0.0.0..0..0...0.1110...1..0000.0...11.000...011.100.010)

```

```

5342: !(FREE)
PSCC-DC610C1:
    SETUP,       RETURN/SETBUSA610C1,          !EXEC SUBR WHICH:
                ! (1) CSP(10)<3:0> -> PS(NZVC)
    NEXT,        CALL[SETUPPSCC#DC]           ! (2) CSP(10)<15> -> D(C)
(5342) DCS(0.00.0.0.0.0) BM(0101..00.01..11.00..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.101.110)

```

```

5343: !(FREE)
SETBUSA610C1:
    P2-T,        SR+BSPHI(C000200),           !GET CONSTANT FOR A-SIDE WHEN CC SET
    NEXT,        J/DOIT610C1
(5343) DCS(0.00.0.0.0.0) BM(1010..01.11..00.00..010..000...0.0.1..0..0...0.0000...0..0000.0...11.000...011.100.100)

```

```

5344: !(FREE)
DOIT610C1:
    SETUP,       SET-CC,                      !FOR CLOCKING CC-S IN NEXT UWORD
                NOT-MODIFY-VBIT,              !EXTRA CC ROM INPUT
    P2-T,        D+A-XOR-B, SAVE-D(C),        !D=(000000), D(C)=(1)
                BUS-A+SR,                     !A=(000200)
                BUS-B+BSPHI(C000200),        !B=(000200)

```

```

15022      NEXT      J/GETIT610C1
(5344) DCS(0.00.0.0.0.0) BM(0110..01.11..00.00..010..111...0.1.0..0..1...0.0000...0..0000.0...11.000...011.100.101)
15023
15024      5345:  !(FREE)
15025      GETIT610C1:
15026          P2-T,   CLK-CC,           !PS(NZVC) GENERATED ABOVE LATCHED HERE
15027          SETUP,  RETURN/TEST610C2, !EXEC SUBR WHICH:
15028          NEXT,   CALL[PSCC10SR3-0] ! PS<3:0> -> SR<3:0> J/BUT(SR3-0)
(5345) DCS(0.00.0.0.0.0) BM(0101..00.11..00.00..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.110.010)
15029
15030
15031
15032
15033      ! - - - - -
15034
15035      !*** TEST 610C2 ***
15036
15037      5602:
15038      TEST610C2:
15039          PO,      LOAD-ENUA(ZTARGET401), !NZVC AFTER = "0001"
15040          LOAD-ERROR(TEST610C2), !ERROR DIRECTORY KEY
15041          DCS-CTR(C13.), !COMPARE AT TARGET
15042          NEXT,     J/LOAD05-610C2
(5602) DCS(1.00.1.0.0.0) BM(0010..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...011.100.110)
15043
15044      5346:  !(FREE)
15045      LOAD05-610C2:
15046          PO,      BUMP-VERIFY, !COUNT
15047          P3,      CSPD(05)+EMIT, EMIT/037777, !A AND B SIDE DATA
15048          NEXT,     J/LOAD01-610C2
(5346) DCS(0.00.0.0.0.1) BM(0011..10.11..11.11..111..111...0.0.0..0..0...0.1010...1..0000.0...11.000...011.100.111)
15049
15050      5347:  !(FREE)
15051      LOAD01-610C2:
15052          P3,      CSPD(01)+EMIT, EMIT/100014, !FOR D(C)=(1), PS(NZVC)="1100" PREVIOUSLY
15053          NEXT,     J/PSCC-DC610C2
(5347) DCS(0.00.0.0.0.0) BM(1000..10.00..00.00..001..100...0.0.0..0..0...0.1110...1..0000.0...11.000...011.101.000)
15054
15055      5350:  !(FREE)
15056      PSCC-DC610C2:
15057          SETUP,  RETURN/SETBUSA610C2, !EXEC SUBR WHICH:
15058          NEXT,   CALL[SETUPPSCC#DC] ! (1) CSP(10)<3:0> -> PS(NZVC)
15059          ! (2) CSP(10)<15> -> D(C)
(5350) DCS(0.00.0.0.0.0) BM(0101..00.01..11.01..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.101.110)
15060
15061      5351:  !(FREE)
15062      SETBUSA610C2:
15063          P2-T,   SR+CSPD(D05), !GET CONSTANT FOR A-SIDE WHEN CC SET
15064          NEXT,   J/DOIT610C2
(5351) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...0.0.1..0..0...0.1010...0..0000.0...11.000...011.101.010)
15065
15066      5352:  !(FREE)

```



```

15067 DOIT610C2:
15068     SETUP, SET-CC,                !FOR CLOCKING CC-S IN NEXT UWORD
15069     NOT-MODIFY-VBIT,             !EXTRA CC ROM INPUT
15070     P2-T,  D+A-PLUS-B, SAVE-D(C), !D=(077776), D(C)=(1)
15071     BUS-A+SR,                     !A=(037777)
15072     BUS-B+CSPO(D05),              !B=(037777)
15073     NEXT,  J/GETIT610C2
(5352) DCS(0.00.0.0.0.0) BM(1001..10.00..00.00..000..111...0.1.0..0..1...0.1010...0..0000.0...11.000...011.101.011)

15074
15075 5353: !(FREE)
15076 GETIT610C2:
15077     P2-T,  CLK-CC,                !PS(NZVC) GENERATED ABOVE LATCHED HERE
15078     SETUP, RETURN/SCOPE610C,      !EXEC SUBR WHICH:
15079     NEXT,  CALL(PSCTOSR3-0)        ! PS<3:0> -> SR<3:0>, J/BUT(SR3-0)
(5353) DCS(0.00.0.0.0.0) BM(0101..00.01..11.01..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.110.010)

15080
15081
15082
15083 5354: !(FREE)
15084 SCOPE610C:
15085     PO,  BUMP-VERIFY,              !COUNT
15086     NEXT, BUTD(SCOPE),             !NO ERROR: "TEST610D1" (+1. WORDS)
15087     J/TEST610D1                    !ERROR: "LOADIR610C1" (-13. WORDS)
(5354) DCS(0.00.0.1.0.1) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...101.100.101)

15088
15089
15090
15091
15092
15093
15094
15095
15096 ! - - - - -
15097
15098 !*** TEST 610D1 ***
15099
15100 5545:
15101 TEST610D1:
15102     PO,  LOAD-ENVA(ZTARGET410),     !NZVC AFTER = "1000"
15103     LOAD-ERROR(TEST610D1),         !ERROR DIRECTORY KEY
15104     DCS-CTR(C12.)                  !COMPARE AT TARGET
15105     NEXT, J/LOAD001-610D1
(5545) DCS(1.00.1.0.0.0) BM(0011..00.11..11.00..001..000...0.0.0..0..0...0.0000...0..0000.0...11.000...101.001.000)

15106
15107 5510:
15108 LOAD001-610D1:
15109     P3,  CSPO(011)+EMIT, EMIT/000005, !FOR D(C)=(0), PS(NZVC)="0101" PREVIOUSLY.
15110     NEXT, J/PSCC-DC610D1
(5510) DCS(0.00.0.0.0.0) BM(0000..10.00..00.00..000..101...0.0.0..0..0...0.1110...1..0000.0...11.000...011.101.101)

15111
15112 5355: !(FREE)
15113 PSCC-DC610D1:
15114     SETUP, RETURN/SETBUSA610D1,    !EXEC SUBR WHICH:

```

```

15115                                     !(1) CSP(10)<3:0> -> PS[NZVC]
15116 NEXT, CALL[SETUPPSCC#DC]           !(2) CSP(10)<15> -> D[C]
(5355) DCS[0.00.0.0.0.0] BM[0101..00.01..11.01..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.101.110]

15117
15118 5356: !(FREE)
15119 SETBUSAB1001:
15120 P2-T, SR+CSPD(006),                !GET CONSTANT FOR A-SIDE WHEN CC SET
15121 NEXT, J/DOIT61001
(5356) DCS[0.00.0.0.0.0] BM[1010..10.00..00.00..000..000...0.0.1..0..0...0.1001...0..0000.0...11.000...011.101.111]

15122
15123 5357: !(FREE)
15124 DOIT61001:
15125 SETUP, SET-CC,                       !FOR CLOCKING CC-S IN NEXT UWORD
15126        MODIFY-VBIT,                 !EXTRA CC ROM INPUT
15127 P2-T, D+A-PLUS-B, SAVE-D[C],      !D=(177400), D[C]=(0)
15128        BUS-A+SR,                   !A=(077600)
15129        BUS-B+CSPD(006),           !B=(077600)
15130 NEXT, J/GETIT61001
(5357) DCS[0.00.0.0.0.0] BM[1001..10.00..00.10..000..111...0.1.0..0..1...0.1001...0..0000.0...11.000...011.110.000]

15131
15132 5360: !(FREE)
15133 GETIT61001:
15134 P2-T, CLK-CC,                       !PS[NZVC] GENERATED ABOVE LATCHED HERE
15135 SETUP, RETURN/TEST61002,          !EXEC SUBR WHICH:
15136 NEXT, CALL[PSCCTOSR3-U]          ! PS<3:0> -> SR<3:0>, J/BUT(SR3-0)
(5360) DCS[0.00.0.0.0.0] BM[0101..00.11..00.10..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.110.010]

15137
15138
15139 ! - - - - -
15140
15141
15142 !** TEST 61002 **
15143
15144 5624:
15145 TEST61002:
15146 PO, LOAD-ENUA(ZTARGET410),         !NZVC AFTER = "1000"
15147        LOAD-ERROR(TEST61002),     !ERROR DIRECTORY KEY
15148        DCS-CTR(.11.),              !COMPARE AT TARGET
15149 NEXT, J/PSCC-DC61002
(5624) DCS[1.00.1.0.0.0] BM[0100..00.11..11.00..001..000...0.0.0..0..0...0.0000...0..0000.0...11.000...011.110.001]

15150
15151 5361: !(FREE)
15152 PSCC-DC61002:
15153 SETUP, RETURN/SETBUSAB1002,
15154 NEXT, CALL[SETUPPSCC#DC]           !(1) CSP(10)<3:0> -> PS[NZVC]
15155                                     !(2) CSP(10)<15> -> D[C]
(5361) DCS[0.00.0.0.0.0] BM[0101..00.01..11.10..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.101.110]

15156
15157 5362: !(FREE)
15158 SETBUSAB1002:
15159 P2-T, SR+BSPHI(C000200),           !GET CONSTANT FOR A-SIDE WHEN CC SET

```

```

15160      NEXT      J/DOIT61002
(5362) DCS(0.00.0.0.0.0) BM(1010..01.11..00.00..010..000...0.0.1..0..0...0.0000...0..0000.0...11.000...011.110.011)
15161
15162      5363:  !(FREE)
15163      DOIT61002:
15164          SETUP,  SET-CC,                !FOR CLOCKING CC-S IN NEXT UWORD
15165          NOT-MODIFY-VBIT,              !EXTRA CC ROM INPUT
15166          P2-T,    D+NOT-A-AND-B, SAVE-D(C), !D=(100000), D(C)=(0)
15167          BUS-A+SR,                        !A=(000200)
15168          BUS-B+CSPD(007),                !B=(100200)
15169      NEXT      J/GETIT61002
(5363) DCS(0.00.0.0.0.0) BM(0010..10.00..00.00..000..111...0.1.0..0..1...0.1000...0..0000.0...11.000...011.110.100)
15170
15171      5364:  !(FREE)
15172      GETIT61002:
15173          P2-T,    CLK-CC,                !PS(NZVC) GENERATED ABOVE LATCHED HERE
15174          SETUP,  RETURN/SCOPE6100,      !EXEC SUBR WHICH:
15175      NEXT      CALL(PSCCT0SR3-0)        ! PS(3:0) -> SR(3:0), J/BUT(SR3-0)
(5364) DCS(0.00.0.0.0.0) BM(0101..00.01..11.10..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.110.010)
15176
15177
15178
15179      5365:  !(FREE)
15180      SCOPE6100:
15181          PO,    BUSDIN+EMIT-[I],         !RESET PROC UCONS
15182          EN-CLK-IR[15-00],
15183          NEXT,  BUTD(SCOPE),             !NO ERROR: "TEST620A" (+6. WORDS)
15184          J/TEST620A,                    !ERROR: "LOAD01-61001" (-10. WORDS)
(5365) DCS(0.00.0.1.0.0) BM(0000..00.00..00.01..000..100...0.0.0..0..0...1.1001...0..0000.0...11.000...101.001.001)
15185
15186
15187
15188
15189
15190
15191
15192      ! - - - - -
15193      !
15194      !   THESE TWO SUBROUTINES ARE USED IN THE ABOVE CC-LOGIC TESTS:
15195
15196      7156:  !(FREE)
15197      SETUPPSCC#DC:
15198          P2-T,    D+CSPD(D01), D(C)+ALU15, !SET D(C) FROM BIT<15>
15199      NEXT      J/SETUPPSCC#DC02
(7156) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..100...0.1.0..0..0...0.1110...0..0000.0...11.000...001.110.001)
15200
15201      7161:  !(FREE)
15202      SETUPPSCC#DC02:
15203          P2,    PS(3-0)+D(3-0)-[I],      !SET PS(CC) FROM BIT<03:00>
15204          BUSDIN+PS-[I],                  !SETUP BUSDIN TO READ PS
15205          NEXT,  BUTA(RETURN),            !AND RETURN
15206          J/BUTERROR7,                    !ERROR IF HERE
(7161) DCS(0.00.0.0.0.0) BM(1000..00.00..10.01..000..000...0.0.0..0..0...1.1011...0..0000.0...11.111...011.111.110)

```

```

15207
15208
15209
15210
15211 7162: !(FREE)
15212 PSCCTOSR3-0:
15213     P3,     CSPD(03)+BUSDIN,      !GET ENABLED PS INTO CSP
15214     NEXT,   J/PSCCTOSR3-0AA      !
(7162) DCS(0.00.0.0.0.0) BM(0000..10.00..00.00..000..000...0.0.0..0..0...0.1100...1..0000.0...11.000...001.110.011)

15215
15216 7163: !(FREE)
15217 PSCCTOSR3-0AA:
15218     P2-T,   SR+CSPD(003),        !MOVE PS(CC) TO SR(3:0)
15219     NEXT,   J/PSCCTOSR3-0BB      !
(7163) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...0.0.1..0..0...0.1100...0..0000.0...11.000...001.110.100)

15220
15221 7164: !(FREE)
15222 PSCCTOSR3-0BB:
15223     P0,     BUSDIN+EMIT-(!),      !RESET EMIT FOR CONSTANTS
15224     EN-CLK-IR(15-00),           !AND IR LOADING
15225     NEXT,   J/BUTSR3-0          !AND GO TEST THE BITS
(7164) DCS(0.00.0.0.0.0) BM(0000..00.00..00.01..000..100...0.0.0..0..0...1.1001...0..0000.0...11.000...010.111.110)

```

```

15226
15227
15228
15229
15230
15231 !.PAGE=====

```

```

15234 .TOC * TEST620-624: TESTING UBREAK AND JAMUPP

```

```

15236 !*****
15237 !*
15238 !* TESTS: 620 - 624 UWORDS: 101 + 133
15239 !*
15240 !* FUNCTIONS:
15241 !*
15242 !* THE FOLLOWING GROUP OF TEST FORCES A JAMUPP CONDITION VIA THE UBREAK FACILITY.
15243 !* TESTS ARE THEN PERFORMED ON JAMUPP RELATED SIDE EFFECTS:
15244 !*
15245 !* ALL JAM CONDITION BITS CAN BE RESET; ACTIVE BUT ROM, CSP WRITE BIT ARE CLEARED
15246 !* IN WORD WHICH JAM OCCURS; NON-INTERNAL-JAM IS SET; CUA IS LOCKED, UNLOCKED BY
15247 !* BUTA(TRACK); PREFETC:4#JAM SET/CLEARED.
15248 !*
15249 !*****

```

```

15253 ! - - - - -
15254 !*** TEST 620 ***
15255 ! - - - - -
15256 ! - - - - -
15257

```

```

15258
15259 !* PART A ***
15260 !TEST-620-A DOES A 'CLR-JAM-ERRORS' UCON-I-O FUNCTION, THEN CHECKS 'INIT JAM'=0
15261 5511:
15262 TEST620A:
15263     PO,      LOAD-ENUA(ZTARGET406),      ! EXPECTED ADDRESS AFTER 'BUT'
15264           LOAD-ERROR(TEST620A),        ! ERROR DIRECTORY KEY
15265           DCS-CTR(C5.),                 ! COMPARE AT TARGET
15266     NEXT,    J/BCERC620A
(5511) DCS[1.00.1.0.0.0] BM[1010..00.11..11.00..000..110...0.0.0..0..0...0.0000...0..0000.0...11.000...101.111.000]

15267
15268 5570:
15269 BCERC620A:
15270     PO,      BUMP-VERIFY,                ! COUNT
15271           P2,  CLR-JAM-ERRORS-[I],      ! CLEAR OUT ERROR REGISTERS
15272     NEXT,    J/GOBUT620A
(5570) DCS[0.00.0.0.0.1] BM[0100..00.00..10.00..000..000...0.0.0..0..0...1.1011...0..0000.0...11.000...011.110.110]

15273
15274 5366: !(FREE)
15275 GOBUT620A:
15276     SETUP,   RETURN/TEST620B,           ! RETURN TO START OF NEXT SUBTEST
15277     NEXT,    GOTO-PAGE(7),              ! BUT TABLE IS ON PAGE 7
15278           J/BUTINITJAM,                 ! GO DO MULTIPLE 'BUT' ON INIT JAM
(5366) DCS[0.00.0.0.0.0] BM[0101..00.11..00.10..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.011.110]

15279
15280 ! - - - - -
15281
15282 !* PART B *
15283 !TEST-620-B CHECKS THAT 'OTHER-JAM-H'=0 ALSO
15284 5623:
15285 TEST620B:
15286     PO,      LOAD-ENUA(ZTARGET401),      ! BIT<01> CLEAR
15287           LOAD-ERROR(TEST620B),        ! ERROR DIRECTORY KEY
15288           DCS-CTR(C3.),                 ! COMPARE AT TARGET
15289     NEXT,    J/GOBUT620B
(5623) DCS[1.00.1.0.0.0] BM[1100..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...011.110.111]

15291
15292 5367: !(FREE)
15293 GOBUT620B:
15294     SETUP,   RETURN/TEST620C,           ! RETURN TO START OF NEXT SUBTEST
15295     NEXT,    GOTO-PAGE(7),              ! BUT TABLE IS ON PAGE 7
15296           J/BUTOTHERJAM,                ! GO DO BUT ON OTHER-JAM-H SIGNAL
(5367) DCS[0.00.0.0.0.0] BM[0101..00.11..00.10..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.101.010]

15297
15298 ! - - - - -
15299
15300 !* PART C *
15301 !TEST-620-C CHECKS THAT STATUS MUX PORT 2 (JAM REG) READS (001000) WHEN RESET
15302 ! "CLEAR-JAM-ERRORS" FUNCTION CLEARS BITS<15,13,11,8-2,0> OF JAM-REG
15303 ! "CLR-YELLOW-ZONE" FUNCTION CLEARS BIT<12> OF JAM-REG
15304 !

```

15305 ! BIT(9) IS ACTIVE LOW, READS AS "1"  
 15306 ! BITS(10,1) READ "0" SINCE NO MCS PRESENT  
 15307 ! BIT(14) IS "0" ALWAYS

5621:

TEST620C:

15308 PO, LOAD-ENVA(ZTARGET402), ! SETUP FOR D<15:00>=0 TEST RESULT  
 15309 LOAD-ERROR(TEST620C), ! ERROR DIRECTORY KEY  
 15310 DCS-CTR(C11.), ! COMPARE AT TARGET  
 15311 BUTA(CLR-FLAG-RES-UCON), ! PUT EMIT ONTO BUSDIN, CLEAR OUT I-O UCON  
 15312 J/MASK620C  
 15313 NEXT  
 15314 (5621) DCS(1.00.1.0.0.0) BM(0100..00.11..11.00. 000..010...0.0.0..0..0...0.0000...0..0000.0...11.010...011.111.000)

5370: !(FREE)

MASK620C:

15318 PO, BUMP-VERIFY, ! COUNT  
 15319 P3, CSPD(04)+EMIT, ! DON'T NEED TO MASK ANYTHING  
 15320 EMITC, EMIT/17777  
 15321 NEXT, J/GETJAM620C  
 (5370) DCS(0.00.0.0.0.1) BM(1111..10.11..11.11..111..111...0.0.0..0..0...0.1011...1..0000.0...11.000...011.111.001)

5371: !(FREE)

GETJAM620C:

15325 SETUP, RETURN/TEST621A, ! GO EXECUTE SUBR WHICH:  
 15326 PO, BUMP-VERIFY, ! COUNT  
 15327 NEXT, CALL(CLRJAM00) ! (JAMREG)-XOR-CSP(02)/(001000) -> D, BUT(D=ZERO)  
 (5371) DCS(0.00.0.0.0.1) BM(0101..00.11..00.01..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.101)

15335 !\*\*\* TEST 621 \*\*\*  
 15336 !CAUSE A MICROBREAK JAM, AND CHECK ALL THE APPROPRIATE SIGNALS ARE SET

!\* PART A \*

15341 !TEST-621-A CAUSES A MICROBREAK JAM AT A SPECIFIC MICROADDRESS, AND CHECKS  
 15342 !THAT THE MICROCODE JAMS TO LOCATION (4777) IMMEDIATELY

5617:

TEST621A:

15345 PO, LOAD-ENVA(4777), ! SETUP JAMUPP ADDRESS  
 15346 LOAD-ERROR(TEST621A), ! ERROR DIRECTORY KEY  
 15347 DCS-CTR(C9.), ! COMPARE AT JAMUPP WORD  
 15348 BUTA(CUA-TRACK), ! RESET CUA TRACKING IF HASN'T BEEN  
 15349 NEXT, J/CSP1L621A  
 (5617) DCS(1.00.1.0.0.0) BM(0110..00.10..01.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.001...0:1.111.010)

5372: !(FREE)

CSP1L621A:

15353 PO, BUMP-VERIFY, ! COUNT

```

15354      EMITC,  EMIT/100377,
15355      P2-U,   IR+EMIT
15356      P3,     CSPD(05)+EMIT,
15357      NEXT,   J/SETBRK621A
(5372) DCS(0.00.0.0.0.0.1) BM(1000..10.00..00.11..111..111...0.0.0..0..0...1.1010...1..0000.0...11.000...011.111.011)
15358
15359      5373:  !(FREE)
15360      SETBRK621A:
15361      SELECT, UCON-PROC
15362      ENABLE,  EN-C1K-UBREAK(11-00),
15363      BUSDIN+EMIT(15-00),
15364      P0,     SET-UCON-CONTROL,
15365      NEXT,   J/LOADBRK621A
15366      (5373) DCS(0.00.0.0.0.0) BM(0000..01.00..00.01..000..000...0.0.0..0..0...1.1001...0..0000.0...11.000...011.111.100)
15367
15368      5374:  !(FREE)
15369      LOADBRK621A:
15370      P2-T,   SR+ZERO
15371      UBREAK+BUSDIN(11-00),
15372      EMITML/5522
15373      NEXT,   J/SETRET621A
(5374) DCS(0.00.0.0.0.0) BM(0011..00.10..11.01..010..010...0.0.1..0..0...1.1010...0..0000.0...11.000...011.111.101)
15374
15375      5375:  !(FREE)
15376      SETRET621A:
15377      P0,     BUMP-VERIFY
15378      P3,     CSPD(00)+EMIT, RETURN/TEST621B,
15379      NEXT,   J/SETFLG621A
(5375) DCS(0.00.0.0.0.0.1) BM(0101..10.11..00.01..011..000...0.0.0..0..0...0.1111...1..0000.0...11.000...011.111.111)
15380
15381      5377:  !(FREE)
15382      SETFLG621A:
15383      P2-T,   D+CSPD(D05), D(C1)+0,
15384      P3,     ASPHI(16)+0,
15385      NEXT,   J/LOADFLG621A
(5377) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..011..000...0.1.0..0..0...0.1010...0..1001.0...11.000...100.000.000)
15386
15387      5400:  !(FREE)
15388      LOADFLG621A:
15389      P0,     BUSDIN+EMIT-(11),
15390      BUMP-VERIFY
15391      P3,     FLAG(8-0)+D(15-8)-(11),
15392      NEXT,   J/SETSR621A
(5400) DCS(0.00.0.0.0.0.1) BM(0000..00.00..00.01..000..001...0.0.0..0..0...1.1011...0..0000.0...11.000...100.000.001)
15393
15394      5401:  !(FREE)
15395      SETSR621A:
15396      P2-T,   SR+CSPD(D05),
15397      NEXT,   J/UBRK621A
(5401) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...0.0.1..0..0...0.1010...0..0000.0...11.000...101.010.010)
15398
15399      5522:
15400      UBRK621A:

```

```

! A BRANCH INSTR, SO 'PREFETCH H' = "0"
! FLAG(8)="1" FOR UBREAK EN; EXFLAG(2:1)="11", FOR ACTIVE BUT YES
! BIT00="1" FOR JAMUPP ROUTINE
! NOTE CSPD00(05)/UCON-OPERATION BIT OVERLAP

```

```

! PROCESSOR UCON:
! FOR UBREAK REG LOAD
! AND KEEP EMIT ON BUSDIN
! WRITE UCON REGISTER

```

```

! ZERO SR(00) TO PREVENT SPURIOUS UBREAKS FROM GETTING THRU
! LOAD MICROBREAK REGISTER
! WITH SELECTED ADDRESS FROM EMIT

```

```

! COUNT
! RETURN ADDRESS FOR AFTER JAMUPP

```

```

! GET VALUES TO LOAD INTO FLAGS
! AND SAVE IN ASP FOR COMPARE LATER

```

```

! KEEP IT ON
! COUNT
! ENABLE U-BREAK, FLAG 8; SET EXFLAGS FOR LATER

```

```

! MAKE SR(00)=1, FOR JAMUPP EXPECTED
! (FOR UBREAK JAM EXPECTED)

```

```

!*** MICROBREAK HERE ***

```

```

15401      P3-T,  D+ZERO, D(C)+ALU15,      !UWORD LATCH FOR CLK-D NOT CLEARED, CLK-D SHOULD HAPPEN
15402      BUTA(CLR-FLAG-RES-UCON),          !JAMUPP CLEAR SHOULD ZAP ACTIVE BUT LATCH, BUTA(CLR-...) SHOULDN'T HA
15403      !CUA GETS LOCKED WITH UADDR OF THIS UWORD (5522) ON JAMUPP
15404      !PREFETCH#JAM(1)H GETS PREFETCH-H (=0) AT JAMUPP
15405      P3,    CSPD(05)+EMIT, EMIT/030004, !LATCH HOLDING "WRCSP" BIT SHOULD GET ZAPPED ON JAMUPP,
15406      !THIS WRITE SHOULDN'T HAPPEN
15407      NEXT  J/ERROR621A                 !THIS UPF SHOULD NOT BE USED
(5522) DCS[0.00.0.0.0.0] BM[0011..10.00..00.00..000..100...1.1.0..0..0...0.1010...1..0000.0...11.010...110.001.101]

```

```

15408
15409
15410      ! (4777) JAMUPP001: ***COMPARE ENABLED ABOVE DONE HERE***
15411      ! THIS WORD TESTS SR<00>, WHICH SHOULD BE SET
15412      ! IF SR<00>=1, GOTO(JAMUPP002B), IF SR<00>=0, GOTO(JAMUPP003) [ERROR]
15413      ! P3-T,  SR+D  SAVE OLD D IN SR, FOR NOW
15414
15415      ! (4757) JAMUPP002B: P2-T,  D+CSPD(00)  GET RETURN ADDRESS INTO D
15416
15417      ! (4XXX) JAMUPP002C: P0,    RETURN+D  LOAD RETURN ADDRESS
15418      ! P2-T,  D+SR  RESTORE OLD D FROM SR
15419
15420      ! (7XXX) JAMUPP002D: P2-T,  SR+ZERO,  ZERO OUT SR, JAMUPPS NOW ILLEGAL
15421      ! NEXT,  BUTA(RETURN)  AND NOW RETURN
15422
15423

```

!EXECUTE THE FOLLOWING WORD ONLY IF NO JAMUPP OCCURRED:

```

15424      5615:
15425      ERROR621A:
15426      P0,    LOAD-ENUA(0000)              !FORCE ERROR
15427      LOAD-ERROR(ERROR621A),             !ERROR DIRECTORY KEY
15428      DCS-CTR(C0.),                       !FORCE COMPARE AT P3-T
15429      BUMP-VERIFY,                         !COUNT
15430      NEXT  J/TEST621B
15431      (5615) DCS[1.00.1.0.0.1] BM[1111..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...110.001.011]

```

! \* PART B \*

!TEST-621-B CHECKS THAT D WAS ZEROED; IE, P3 PULSE WAS SUPPRESSED IN JAM WORD, BUT UWORD LATCH FOR CLK-D BIT NOT ZAPPED, SO BIT WAS SAVED FOR EXECUTE LATER

```

15439      5613:
15440      TEST621B:
15441      P0,    LOAD-ENUA(ZTARGET402),        !SETUP FOR D = ZERO TEST
15442      LOAD-ERROR(TEST621B),               !ERROR DIRECTORY KEY
15443      DCS-CTR(C3.),                       !COMPARE AT TARGET
15444      NEXT  J/GOBUT621B
(5613) DCS[1.00.1.0.0.0] BM[1100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.000.010]

```

5402: !(FREE)

```

15446      GOBUT621B:
15447      SETUP,  RETURN/TEST621C,           !RETURN TO START OF NEXT SUBTEST
15448      NEXT,  GOTO-PAGE(7),               !BUT TABLE ON PAGE 7
15449      J/BUTD-IS-ZERO                       !BUT ON D CONTENTS
15450

```



(5402) DCS(0.00.0.0.0.0) BM(0101..00.11..00.01..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

15451  
15452  
15453  
15454  
15455  
15456  
15457  
15458  
15459  
15460  
15461  
15462

! \* PART C \*  
! TEST-621-C CHECKS THAT 'INIT JAM' STILL OFF

5611:  
TEST621C:  
PO, LOAD-ENUA(ZTARGET406), !BIT<00> CLEAR  
LOAD-ERROR(TEST621C), !ERROR DIRECTORY KEY  
DCS-CTR(C4.), !COMPARE AT TARGET  
NEXT J/GOBUT621C

(5611) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..000..110...0.0.0..0..0...0.0000...0..0000.0...11.000...100.000.011)

15463  
15464  
15465  
15466  
15467  
15468

5403: !(FREE)  
GOBUT621C:  
SETUP, RETURN/TEST621D, !RETURN TO START OF NEXT SUBTEST  
NEXT, GOTO-PAGE(7), !BUT TABLE ON PAGE 7  
J/BUTINITJAM !GO TEST INIT JAM

(5403) DCS(0.00.0.0.0.0) BM(0101..00.11..00.00..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.011.110)

15469  
15470  
15471  
15472  
15473  
15474  
15475  
15476  
15477  
15478  
15479  
15480

! \* PART D \*  
! TEST-621-D CHECKS THAT 'OTHER-JAM-H'=1, WAS SET FOR UBREAK

5607:  
TEST621D:  
PO, LOAD-ENUA(ZTARGET403), !BIT<01> SET  
LOAD-ERROR(TEST621D), !ERROR DIRECTORY KEY  
DCS-CTR(C3.), !COMPARE AT TARGET  
NEXT J/GOBUT621D

(5607) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...100.000.100)

15481  
15482  
15483  
15484  
15485  
15486  
15487

5404: !(FREE)  
GOBUT621D:  
SETUP, RETURN/TEST621E, !RETURN TO START OF NEXT SUBTEST  
PO, BUMP-VERIFY, !COUNT  
NEXT, GOTO-PAGE(7), !BUT TABLE ON PAGE 7  
J/BUTOTHERJAM !GO TEST 'OTHER-JAM-H' IS SET

(5404) DCS(0.00.0.0.0.1) BM(0101..00.11..00.00..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.101.010)

15488  
15489  
15490  
15491  
15492  
15493  
15494  
15495  
15496  
15497

! \* PART E \*  
! TEST-621-E CHECKS UBREAK ONLY BIT SET FROM UBREAK JAMUPP IN STATUS MUX PORT 2 (JAM REG)

5605:  
TEST621E:  
PO, LOAD-ENUA(ZTARGET402), !SETUP FOR D = ZERO COMPARE  
LOAD-ERROR(TEST621E), !ERROR DIRECTORY KEY

```

15498          DCS-CTR(C10.),          !COMPARE AT TARGET
15499      NEXT      J/EXPEC621E
(5605) DCS(1.00.1.0.0.0) BM(0101..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.000.101)

15500
15501      5405:  !(FREE)
15502      EXPEC621E:
15503          P3,      CSPD(02)+EMIT,          !WHAT WE EXPECT TO SEE IN JAM IS:
15504          EMITC,   EMIT/001001,          ! (001001),  UBREAK H IN BIT00
15505      NEXT      J/GETJAM621E
(5405) DCS(0.00.0.0.0.0) BM(0000..10.00..10.00..000..001...0.0.0..0..0...0.1101...1..0000.0...11.000...100.000.110)

15506
15507      5406:  !(FREE)
15508      GETJAM621E:
15509          SETUP,  RETURN/TEST621F,          !GO TO SUBR WHICH:
15510      NEXT      CALL(JAMTOD)          ! (JAMREG)-XOR-CSP(02) -> D, BUT(D=ZERO)
(5406) DCS(0.00.0.0.0.0) BM(0101..00.11..00.00..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.110)

15511
15512
15513
15514      ! - - - - -
15515
15516      !* PART F *
15517      !TEST-621-F CHECKS THAT THE RIGHT CUA WAS LOCKED, AND SHORT TERM FLAGS NOT CLEARED (IE,
15518      ! BUTA(CLR-FLAG-...) IN UBREAK WORD DIDN'T CLEAR EXFLAG1), INDICATING JAMUPP CLEAR L
15519      ! DID IN FACT ZAP THE ACTIVE BUT ROM LATCH. ALSO PREFETCH*JAM(1)H GETS PREFETCH-H="0".
15520      5603:
15521      TEST621F:
15522          PO,      LOAD-ENUA(ZTARGET402),          !SETUP FOR D = ZERO COMPARE
15523          LOAD-ERROR(TEST621F),          !ERROR DIRECTORY KEY
15524          DCS-CTR(C10.),          !COMPARE AT TARGET
15525      NEXT      J/EXPEC621F
(5603) DCS(1.00.1.0.0.0) BM(0101..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.000.111)

15526
15527      5407:  !(FREE)
15528      EXPEC621F:
15529          PO,      BUMP-VERIFY          !COUNT
15530          P3,      CSPD(02)+EMIT,          ! (USE MASK OF ALL 1'S FROM BEFORE)
15531          EMITC,   EMIT/055226          !WHAT THE CUA-EXFLAG-FOVP PORT OF HBMUX SHOULD BE
15532      NEXT      J/GETCUA621F          !CUA=(5522), EXFLAG<2:1>="11", PREFETCH*JAM(1)H="0"
(5407) DCS(0.00.0.0.0.1) BM(0101..10.10..10.10..010..110...0.0.0..0..0...0.1101...1..0000.0...11.000...100.001.000)

15533
15534      5410:  !(FREE)
15535      GETCUA621F:
15536          SETUP,  RETURN/TEST621G,          !GO TO SUBR WHICH:
15537          PO,      BUMP-VERIFY          !COUNT
15538      NEXT      CALL(CUATOD)          ! (CUA)-XOR-CSP(02) -> D, BUT(D=ZERO)
(5410) DCS(0.00.0.0.0.1) BM(0101..00.11..11.10..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.111)

15539
15540
15541
15542
15543

```

```

15544
15545 ! - - - - -
15546
15547 !* PART G *
15548 !TEST-621-G CHECKS THAT 'PREFETCH#JAM(1)H' GOT 'PREFETCH-H'="0" AFTER JAMUPP
15549 5766:
15550 TEST621G:
15551     PO,          LOAD-ENUA(ZTARGET401),          !BIT01 CLEAR
15552     LOAD-ERROR(TEST621G),          !ERROR DIRECTORY KEY
15553     DCS-CTR(C3.),          !COMPARE AT TARGET
15554     NEXT,        J/GOBUT621G
(5766) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...100.001.001)

15555
15556 5411: !(FREE)
15557 GOBUT621G:
15558     SETUP,      RETURN/TEST621H,          !RETURN TO START OF NEXT SUBTEST
15559     PO,          BUMP-VERIFY,          !COUNT
15560     NEXT,       GOTO-PAGE(7),          !BUT TABLE ON PAGE 7
15561     J/BUTPREFETCHJAM          !PREFETCH#JAM(1)H IN BIT01
(5411) DCS(0.00.0.0.0.1) BM(0101..00.11..11.01..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.110.000)

15562
15563
15564
15565
15566 ! - - - - -
15567
15568 !* PART H *
15569 !TEST-621-H CHECKS THAT CSP(05) DID NOT GET WRITTEN IN THE JAMMED WORD ABOVE,
15570 ! AND IS IN FACT, EQUAL TO THE SAVED COPY OF ITS CONTENTS, IN ASPHI(16)
15571 5754:
15572 TEST621H:
15573     PO,          LOAD-ENUA(ZTARGET402),          !SETUP FOR D=ZERO COMPARE
15574     LOAD-ERROR(TEST621H),          !ERROR DIRECTORY KEY
15575     DCS-CTR(C4.),          !COMPARE AT TARGET
15576     BUMP-VERIFY,          !COUNT
15577     NEXT,        J/COMP621H
(5754) DCS(1.00.1.0.0.1) BM(1011..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.001.010)

15578
15579 5412: !(FREE)
15580 COMP621H:
15581     P2-T,       D=CSP0(05)-XOR-ASPHI(R16),          !COMPARE CURRENT:SAVED, SHOUL BE SAME
15582     NEXT,       GOTO-PAGE(7),          !XFER
15583     J/GOBUT621H
(5412) DCS(0.00.0.0.0.0) BM(0110..10.00..11.00..011..111...0.1.0..0..0...0.1010...0..0000.0...11.100...001.110.000)

15584
15585 7160: !(FREE)
15586 GOBUT621H:
15587     SETUP,      RETURN/TEST622A,          !RETURN TO START OF NEXT SUBTEST
15588     NEXT,       GOTO-PAGE(7),          !BUT TABLE ON PAGE 7
15589     J/BUTD-IS-ZERO          !TEST FOR EQUALITY
(7160) DCS(0.00.0.0.0.0) BM(0101..00.11..00.00..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

15590

```

```

15591
15592
15593 ! -----
15594
15595 !*** TEST 622 ***
15596 !DO A 'CLR-JAM-ERRORS' FUNCTION TO CLEAR OUT SET BITS
15597 !MAKE SURE THAT 'START-DELAY', 'CLR-NPR-TIMEOUT' DON'T CLEAR THEM
15598
15599 ! -----
15600
15601 !* PART A *
15602 !TEST-622-A CHECKS THAT 'START-DELAY', 'CLR-NPR-TIMEOUT' DON'T AFFECT
15603 !THE 'CLR-JAM-ERRORS' FUNCTION'
15604 5601:
15605 TEST622A:
15606     PO,      LOAD-ENUA(ZTARGET403),      !SETUP FOR BUT ON 'OTHER-JAM-H'=(1)
15607             LOAD-ERROR(TEST622A),      !ERROR DIRECTORY KEY
15608             DCS-CTR(C5.),              !COMPARE AT TARGET
15609     NEXT,    J/BCIFC622A
(5601) DCS(1.00.1.0.0.0) BM(1010..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...100.001.011)

15610
15611 5413: !(FREE)
15612 BCIFC622A:
15613     PO,      BUMP-VERIFY,              !COUNT
15614     P2,      START-DELAY-[[1],         !DO A 'START-DELAY'
15615     NEXT,    J/CNST0622A
(5413) DCS(0.00.0.0.0.1) BM(0100..00.00..01.00..000..000...0.0.0..0..0...1.1011...0..0000.0...11.000...100.001.100)

15616
15617 5414: !(FREE)
15618 CNST0622A:
15619     P2,      CLR-NPR-TIMEOUT-[[1],     !DO A 'CLR-NPR-TIMEOUT'
15620     NEXT,    J/GOBUT622A              !NEITHER OF THESE SHOULD ASSERT 'CLR-JAM-ERRORS'
(5414) DCS(0.00.0.0.0.0) BM(0100..00.00..11.00..000..000...0.0.0..0..0...1.1011...0..0000.0...11.000...100.001.101)

15621
15622 5415: !(FREE)
15623 GOBUT622A:
15624     SETUP,   RETURN/TEST622B,         !RETURN TO START OF NEXT SUBTEST
15625     NEXT,    GOTO-PAGE(7),            !BUT TABLE ON PAGE 7
15626             J/BUTOTHERJAM           !TEST 'OTHER-JAM-H' STILL SET
(5415) DCS(0.00.0.0.0.0) BM(0101..00.10..11.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.101.010)

15627
15628
15629 ! -----
15630
15631 !* PART B*
15632 !TEST-622-B NOW CHECKS THAT 'CLR-JAM-ERRORS' DOES JUST THAT
15633 5577:
15634 TEST622B:
15635     PO,      LOAD-ENUA(ZTARGET401),      !SETUP FOR BUT ON 'OTHER-JAM-H'=(0)
15636             LOAD-ERROR(TEST622B),      !ERROR DIRECTORY KEY
15637             DCS-CTR(C4.),              !COMPARE AT TARGET
15638     NEXT,    J/BCERC622B
(5577) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...100.001.110)

```

```

15639
15640 5416: !(FREE)
15641 BCERC622B:
15642 P2, CLR-JAM-ERRORS-[1], !DO IT
15643 NEXT, J/GOBUT622B
(5416) DCS[0.00.0.0.0.0] BM[0100..00.00..10.00..000..000...0.0.0..0..0...1.1011...0..0000.0...11 000...100.001.111]

```

```

15644
15645 5417: !(FREE)
15646 GOBUT622B:
15647 SETUP, RETURN/TEST622C, !RETURN TO START OF NEXT SUBTEST
15648 NEXT, GOTO-PAGE(7), !BUT TABLE ON PAGE 7,
15649 J/BUTOTHERJAM !TEST 'OTHER-JAM-H' NOW CLEAR
(5417) DCS[0.00.0.0.0.0] BM[0101..00.10..11.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.101.010]

```

```

15650
15651 ! - - - - -
15652
15653

```

```

15654 !* PART C *
15655 !TEST-622-C TESTS THAT STATUS MUX PORT 2 (JAM REG) READS (001000) WHEN RESET
15656 5575:
15657 TEST622C:
15658 P0, LOAD-ENUA(ZTARGET402), !SETUP FOR D = ZERO TEST
15659 LOAD-ERROR(TEST622C), !ERROR DIRECTORY KEY
15660 DCS-CTR(C10.), !COMPARE AT TARGET
15661 P3, BUTA(CLR-FLAG-RES-UCON), !PUT EMIT ON BUSDIN, CLEAR SHORT TERM FLAGS
15662 NEXT, J/GETJAM622C
(5575) DCS[1.00.1.0.0.0] BM[0101..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.010...100.010.000]

```

```

15663
15664 5420: !(FREE)
15665 GETJAM622C:
15666 SETUP, RETURN/TEST623, !GO TO SUBR WHICH:
15667 NEXT, CALL[CLRJAMTOD] ! (JAMREG)-XOR-CSP(02)/(001000) -> D, BUT(D=ZERO)
(5420) DCS[0.00.0.0.0.0] BM[0101..00.10..11.11..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.101]

```

```

15668
15669
15670
15671
15672
15673
15674
15675

```

```

15676 ! - - - - -
15677

```

```

15677 !*** TEST 623 ***
15678 !TEST-623 CHECKS THAT BUTA(CUA-TRACK) RESTARTS CUA TRACKING
15679 5573:
15680 TEST623:
15681 P0, LOAD-ENUA(ZTARGET402), !SETUP FOR D = ZERO TEST
15682 LOAD-ERROR(TEST623), !ERROR DIRECTORY KEY
15683 DCS-CTR(C10.), !COMPARE AT TARGET
15684 P3, BUTA(CUA-TRACK), !RESET TO TRACKING CUA MODE
15685 NEXT, J/EXPEC623
(5573) DCS[1.00.1.0.0.0] BM[0101..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.001...100.010.001]

```

```

15686
15687 5421: !(FREE)
15688 EXPECT623:
15689 PO, BUMP-VERIFY, !COUNT
15690 P3, CSPQ(02)+EMIT, ! (PREV MASK)
15691 EMITC, EMIT/073734, ! CUA IS WORD WHICH LOADS CUA INTO CSP, ONLY
15692 NEXT, J/GETCUA623, ! EXFLAG<1> CLEARED FROM BUTA(CLR-FLAG-...), ABOVE
(5421) DCS(0.00.0.0.0.0.1) BM(0111..10.01..11.11..011..100...0.0.0..0..0...0.1101...1..0000.0...11.000...00.010.010)

15693
15694 5422: !(FREE)
15695 GETCUA623:
15696 SETUP, RETURN/SCOPE623, !GO TO SUBR WHICH:
15697 NEXT, CALL(CUAT00) ! CUA=(WORD WHICH READS CUA), SINCE TRACKING RESET
15698 (5422) DCS(0.00.0.0.0.0.0) BM(0101..00.10..00.10..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.111)

15699
15700
15701 5423: !(FREE)
15702 SCOPE623:
15703 PO, BUSDIN+EMIT-[I], !RESET PROC UCONS
15704 EN-CLK-IR[15-00], !IR GETS JUNK, BUT DON'T CARE
15705 P3, FLAG[8-0]+D[15-8]-[I], !RESET FLAGS TO ALL ZERO
15706 NEXT, BUTD(SCOPE), !NO ERROR: "TEST624A" [+1. WORDS]
15707 J/TEST624A ! ERROR: "BCERC620A" [-48. WORDS]
(5423) DCS(0.00.0.1.0.0.0) BM(0000..00.00..00.01..000..101...0.0.0..0..0...1.1011...0..0000.0...11.000...101.111.001)

15708
15709
15710
15711
15712 ! -----
15713
15714 !*** TEST 624 ***
15715 !DO ANOTHER UBREAK JAM, CHECK JAM OCCURS, AND THAT P2 IS SEEN IN JAM WORD
15716
15717 ! -----
15718
15719
15720 !* PART A *
15721 !TEST-624-A CAUSES A UBREAK JAM, AND CHECKS TO SEE IT OCCURS
15722
15723 5571:
15724 TEST624A:
15725 PO, LOAD-ENUA(4777), !SETUP JAMUPP ADDRESS
15726 LOAD-ERROR(TEST624A), !ERROR DIRECTORY KEY
15727 DCS-CTR(C9.), !COMPARE AT JAMUPP WORD
15728 NEXT, J/LODIR624A
(5571) DCS(1.00.1.0.0.0.0) BM(0110..00.10..01.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.000...101.110.010)

15729
15730 5562:
15731 LODIR624A:
15732 P2-U, IR+EMIT, !IR PATTERN ASSERTS PREFETCH-H:
15733 EMIT/030603, ! PREFETCH-H=(DOPH*MOVBL*SMOH*SR7L*DMOH*DR7L)
15734 P3, FLAG[8-0]+D[15-8], !FLAGS GET D, WHICH IS ZERO

```

```

15735      NEXT,      J/SETBRK624A      !      + (SOPH*NEGL*SBCL*ROAL*ASRL*DMOH*DR7L)
(5562) DCS(0.00.0.0.0.0) BM(0011..00.00..01.10..000..011...0.0.0..0..0...1.1010...0..0000.0...11.000...100.010.100)
15736
15737      5424:      !(FREE)
15738      SETBRK624A:
15739      SELECT,     UCON-PROC,          !ENABLE PROCESSOR UCON:
15740      ENABLE,     EN-CLK-UBREAK(11-00), ! EN UBREAK LOAD,
15741      BUSDIN+EMIT(15-00),             ! KEEP EMIT ON BUSDIN
15742      PO,         SET-UCON-CONTROL,    !WRITE CONTROLS
15743      BUMP-VERIFY,             !COUNT
15744      NEXT,      J/LOOBRK624A
(5424) DCS(0.00.0.0.0.1) BM(0000..01.00..00.01..000..000...0.0.0..0..0...1.1001...0..0000.0...11.000...100.010.101)
15745
15746      5425:      !(FREE)
15747      LOOBRK624A:
15748      PO,         BUMP-VERIFY,          !COUNT
15749      P2-T,       UBREAK+BUSDIN(11-00), !LOAD UBREAK REGISTER
15750      EMITML/6255,             ! WITH SELECTED ADDRESS FROM EMIT
15751      NEXT,      J/SETRET624A
(5425) DCS(0.00.0.0.0.1) BM(0000..00.11..00.10..101..101...0.0.0..0..0...1.1010...0..0000.0...11.000...100.010.110)
15752
15753      5426:      !(FREE)
15754      SETRET624A:
15755      P3,         CSPD(00)+EMIT, RETURN/TEST624B, !RETURN ADDRESS FOR AFTER JAMUPP
15756      NEXT,      GOTO-PAGE(6),         ! RETURN TO START OF NEXT SUBTEST
15757      J/SETD624A
(5426) DCS(0.00.0.0.0.0) BM(0101..10.10..11.10..101..110...0.0.0..0..0...0.1111...1..0000.0...11.100...011.000.110)
15758
15759      6306:      !(FREE)
15760      SETD624A:
15761      P2-T,       D+CSPD(D05) D(C)+ALU15,      !SETUP D WITH VALUE TO GO INTO FLAGS<8:0>
15762      NEXT,      J/SETFLG624A
(6306) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..100...0.1.0..0..0...0.1010...0..0000.0...11.000...100.101.011)
15763
15764      6453:      !(FREE)
15765      SETFLG624A:
15766      PO,         BUSDIN+EMIT-[1],          !KEEP IT ON
15767      P3,         FLAG(8-0)+D(15-8)-[1],    !SET UBRK FLAG<8>
15768      NEXT,      J/SETSR624A
(6453) DCS(0.00.0.0.0.0) BM(0000..00.00..00.01..000..001...0.0.0..0..0...1.1011...0..0000.0...11.000...100.101.100)
15769
15770      6454:      !(FREE)
15771      SETSR624A:
15772      P2-T,       SR+CSPD(D05),             !SET SR<00>=1 FOR UBRK JAMUPP EXPECTED
15773      NEXT,      J/UBRK624A
(6454) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...0.0.1..0..0...0.1010...0..0000.0...11.000...010.101.101)
15774
15775      6255:
15776      UBRK624A:
15777      P2-T,       D+ZERO, D(C)+ALU15,      !*** MICROBREAK HERE ***
15778      !P2 PULSE SHOULD OCCUR, SO D SHOULD BE ZEROED.
15779      !CUA GETS LOCKED WITH UADDR OF THIS UWORD (6255) ON JAMUPP
15780      !PREFETCH*JAM(1)H GETS PREFETCH-H (= "1") AT JAMUPP

```

```

15780 P3, FLAG(8-0)+0(15-8), !ZERO ALL THE FLAGS IN THIS UWORD. THE LBREAK SHOULD
15781 !STILL OCCUR, HOWEVER, AS IT LOOKS AT THE 'NUA'.
15782 NEXT, J/ERROR624A !OPF SHOULD NOT BE USED.
(6255) DCS(0.00.0.0.0.0) BM(0011..00.00..00.00..000..100...0.1.0..0..0...1.1010...0..0000.0...11.000...101.110.111)

```

```

15783
15784
15785 ! (4777) JAMUPP001: ***COMPARE ENABLED ABOVE DONE HERE***
15786 THIS WORD TESTS SR<00>, WHICH SHOULD BE SET
15787 IF SR<00>=1, GOTO(JAMUPP002B), IF SR<00>=0, GOTO(JAMUPF003) [ERROR]
15788 P3-T, SR+0 SAVE OLD D IN SR, FOR NOW
15789
15790 ! (4757) JAMUPP002B: P2-T, D+CSPD(00) GET RETURN ADDRESS INTO D
15791
15792 ! (4XXX) JAMUPP002C: PD, RETURN+D LOAD RETURN ADDRESS
15793 P2-T, D+SR RESTORE OLD D FROM SR
15794
15795 ! (7XXX) JAMUPP002D: P2-T, SR+ZERO, ZERO OUT SR, JAMUPPS NOW ILLEGAL
15796 NEXT, BUTA(RETURN) AND NOW RETURN
15797
15798
15799
15800

```

!EXECUTE THE FOLLOWING WORD ONLY IF NO JAMUPP OCCURRED:

6567:

ERROR624A:

```

15804 PD, LOAD-ENUA(0005), !FORCE ERROR
15805 LOAD-ERROR(ERROR624A), !ERROR DIRECTORY KEY
15806 DCS-CTR(C0.), !FORCE COMPARE AT P3-T
15807 BUMP-VERIFY, !COUNT
15808 NEXT, GOTO-PAGE(5), !XFER TO 5
15809 J/TEST624B

```

```

(6567) DCS(1.00.1.0.0.1) BM(1111..00.00..00.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.100...101.110.101)

```

! \* PART B \*

!TEST-624-B CHECKS THAT D WAS ZEROED, IE, P2 DID OCCUR IN JAM WORD

5565:

TEST624B:

```

15818 PD, LOAD-ENUA(ZTARGET434), !SETUP FOR IR=ZERO TEST
15819 LOAD-ERROR(TEST624B), !ERROR DIRECTORY KEY
15820 DCS-CTR(C6.), !COMPARE AT TARGET
15821 NEXT, J/ZEROIR624B

```

```

(5565) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..011..100...0.0.0..0..0...0.0000...0..0000.0...11.000...100.010.111)

```

5427: !(FREE)

ZEROIR624B:

```

15825 SETUP, RETURN/TEST624C, !PUT THE ZEROES IN D INTO THE IR,
15826 NEXT, CALL(DINTOIR-5) ! TO NEGATE PREFETCH-H (000000)=HALT

```

```

(5427) DCS(0.00.0.0.0.0) BM(0101..00.11..11.11..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

```

15827



15828  
15829  
15830  
15831  
15832  
15833  
15834  
15835  
15836  
15837  
15838  
15839  
15840  
15841  
15842  
15843  
15844  
15845  
15846  
15847  
15848  
15849  
15850  
15851  
15852  
15853  
15854  
15855  
15856  
15857  
15858  
15859  
15860  
15861  
15862  
15863  
15864  
15865  
15866  
15867  
15868  
15869  
15870  
15871  
15872  
15873  
15874

! - - - - -

!\* PART C \*  
!TEST-624-C CHECKS THAT THE RIGHT CUA WAS LOCKED,  
! THAT THE EXFLAG(2:1)H READ AS "00" AND  
! THAT PREFETCH\*JAM(1)H = "1" FROM TEST-624-A, EVEN THOUGH TEST-624-B  
! RESET THE IR SO THAT PREFETCH-H="0"

5776:  
TEST624C:  
PO, LOAD-ENUA(ZTARGET402), !SETUP FOR D = ZERO COMPARE  
LOAD-ERROR(TEST624C), !ERROR DIRECTORY KEY  
DCS-CTR(C10.), !COMPARE AT TARGET  
NEXT, J/EXPEC624C  
(5776) DCS(1.00.1.0.0.0) BM(0101..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0. .11.00C...100.011.000)

5430: !(FREE)  
EXPEC624C:  
P3, CSPD(02)+EMIT, !(USE MASK OF ALL 1'S FROM BEFORE)  
EMIT/062551 !WHAT THE CUA-EXFLAG-FOVP PORT OF HBMUX SHOULD BE  
NEXT, J/GETCUA624C !CUA=(6255), EXFLAG="00", PREFETCH\*JAM(1)H="1"  
(5430) DCS(0.00.0.0.0.0) BM(0110..10.01..01.01..101..001...0.0.0..0..0...0.1101...1..0000.0...11.000...100.011.001)

5431: !(FREE)  
GETCUA624C:  
SETUP, RETURN/TEST624D, !GO TO SUBR WHICH:  
PO, BUMP-VERIFY, !COUNT  
NEXT, CALL(CUAT00) ! (CUA)-XOR-CSP(02) -> D, BUT(D=ZERO)  
(5431) DCS(0.00.0.0.0.1) BM(0101..00.11..11.01..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.111)

! - - - - -

!\* PART D \*  
!TEST-624-D CHECKS THAT 'PREFETCH\*JAM(1)H' GOT 'PREFETCH-H'="1" AFTER JAMUPP

5756:  
TEST624D:  
PO, LOAD-ENUA(ZTARGET403), !BIT01 SET  
LOAD-ERROR(TEST624D), !ERROR DIRECTORY KEY  
DCS-CTR(C3.), !COMPARE AT TARGET  
NEXT, J/GOBUT624D  
(5756) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...100.011.010)

5432: !(FREE)  
GOBUT624D:  
SETUP, RETURN/CLEAR624, !RETURN TO SCOPE LOOP TEST WORD  
NEXT, GOTO-PAGE(7), !BUT TABLE ON PAGE 7  
J/BUTPREFETCHJAM !PREFETCH\*JAM(1)H IN BIT01  
(5432) DCS(0.00.0.0.0.0) BM(0111..00.00..11.10..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.110.000)

```

15875
15876
15877 7165: !(FREE)
15878 CLEAR624:
15879     SETUP, RETURN/SCOPE624, !RETURN FOR SCOPE LOOP TEST
15880     NEXT, CALL(CLEAR-I-0-B) !CLEAR OUT WHAT I'VE DONE
(7165) DCS(0.00.0.0.0.0) BM(0101..00.01..00.10..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.011.000)

```

```

15881
15882
15883 5224: !(FREE)
15884 SCOPE624:
15885     P0, BUMP-VERIFY, !COUNT
15886     BUSDIN+EMIT-(I), !RESET PROC UCON
15887     EN-CLK-IR(15-00),
15888     P3, BUTA(CUA-TRACK), !AND START CUA TRACKING AGAIN
15889     NEXT, BUTD(SCOPE), !NO ERROR: "TEST701A" [+1. WORDS]
15890     J/TEST701A !ERROR: "SETIR624A" [-18 WORDS]
(5224) DCS(0.00.0.1.0.1) BM(0000..00.00..00.01..000..100...0.0.0..0..0...1.1001...0..0000.0...11.001...101.110.011)

```

```

15891
15892
15893
15894
15895
15896
15897
15898 !.PAGE=====
15899

```

```

15900
15901 .TOC * TEST701-702: LOAD/READ THE BA, FULL 18. BITS
15902

```

```

15903 !*****
15904 !
15905 ! TESTS: 701A - 702B UWORDS: 033 + 042
15906 !
15907 ! FUNCTIONS:
15908 !
15909 ! THIS NEXT SET OF SIX TESTS CHECKS THE "BA" REGISTER RELATED FUNCTIONS:
15910 ! SPECIFICALLY,
15911 ! 1) LOAD/READ OF BA<15:00>, FROM BUS-A -> BA -> STATUS-MUX/PBA-PORT
15912 ! 2) BUT ON "BA<00>"
15913 ! 3) LOAD READ OF BA<17:16>, FROM BUS-B -> BA -> STATUS-MUX/SERVICE-PORT
15914 ! THE EFFECT OF PARTICULAR KT-UCON-FUNCTIONS/CONSOLE-18.-BIT-MODE
15915 ! IN THIS SITUATION IS ALSO EXAMINED FOR FUNCTIONALITY.
15916 !
15917 !*****

```

```

15918
15919
15920
15921 ! - - - - -

```

```

15922
15923 !TEST-701-A CHECKS THAT BA<15:00> CAN BE LOADED, AND READ BACK WITH (052525)
15924 5563:
15925 TEST701A:
15926     P0, LOAD-ENUA(ZTARGET402), !SETUP FOR D=ZERO COMPARE

```

KD11-K MICRO V00A-1 00:00:03 12-MAR-77

```

15927 LOAD-ERROR(TEST701A), !ERROR DIRECTORY KEY
15928 DCS-CTR(C12.), !COMPARE AT TARGET
15929 BUMP-VERIFY, !COUNT
15930 NEXT, J/MASK701A
(5563) DCS(1.00.1.0.0.1) BM(0011..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.111.110)

```

```

15931 5476:
15932 MASK701A:
15933 P3, CSPD(04)+EMIT, EMIT/177777, !MASK TO READ ALL THE BITS
15934 NEXT, GOTO-PAGE(7), !XFER
15935 J/EXPECT701A
(5476) DCS(0.00.0.0.0.0) BM(1111..10.''.11.11..111..111...0.0.0..0..0...0.1011...1..0000.0...11.100...001.110.110)

```

```

15937 7166: !(FREE)
15938 EXPECT701A:
15939 P3, CSPD(02)+EMIT, EMIT/052525, !EXPECTED DATA TO BE READ OUT OF PBA AFTER LOAD:
15940 NEXT, J/LOADBA701A !"0101 0101 0101 0101"
15941 (7166) DCS(0.00.0.0.0.0) BM(0101..10.01..01.01..010..101...0.0.0..0..0...0.1101...1..0000.0...11.000...001.111.000)

```

```

15942 7170: !(FREE)
15943 LOADBA701A:
15944 P1, BA+ASPHI(C052525), !LOAD BA<15:00> WITH PATTERN
15945 NEXT, J/GOTEST701A
(7170) DCS(0.00.0.0.0.0) BM(0000..00.00..11.01..111..000...0.0.0..1..0...0.0000...0..0000.0...11.000...001.111.001)

```

```

15947 7171: !(FREE)
15948 GOTEST701A:
15949 SETUP, RETURN/TEST701B, !GO TO SUBR THAT:
15950 NEXT, CALL(PBAT00) ! (PBA)-XOR-CSP(02)->0, BUT(0-IS-ZERO)
15951 (7171) DCS(0.00.0.0.0.0) BM(0101..00.10..10.10..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.001)

```

```

15952
15953
15954
15955
15956
15957
15958 ! - - - - -
15959

```

```

15960 !TEST-701-B CHECKS THAT BA<00> WAS LOADED WITH A (1)
15961 5523:
15962 TEST701B:
15963 PO, LOAD-ENUA(ZTARGET403), !BIT<00> SET
15964 LOAD-ERROR(TEST701B), !ERROR DIRECTORY KEY
15965 DCS-CTR(C3.), !COMPARE AT TARGET
15966 NEXT, J/GOBUT701B
(5523) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...100.011.100)

```

```

15967 5434: !(FREE)
15968 GOBUT701B:
15969 SETUP, RETURN/TEST701C, !RETURN TO START OF NEXT SUBTEST
15970 NEXT, GOTO-PAGE(7), !BUT TABLE
15971 J/BUTBA00 !BA<00>H IN BIT<00>
15972

```

(5434) DCS(0.00.0.0.0.0) BM(0110..00.11..11.11..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.101.001)

15973  
15974  
15975  
15976  
15977  
15978  
15979  
15980  
15981  
15982  
15983  
15984  
15985

! - - - - -

!TEST-701-C CHECKS THAT BA<15:00> CAN BE LOADED, AND READ BACK WITH (125252)

6770:

TEST701C:

PO,

LOAD-ENVA(ZTARGET402),

!SETUP FOR D=ZERO COMPARE

LOAD-ERROR(TEST701C),

!ERROR DIRECTORY KEY

DCS-CTR(C11.),

!COMPARE AT TARGET

NEXT

J/EXPEC701C

(6770) DCS(1.00.1.0.0.0) BM(0100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.101.101)

15986  
15987  
15988  
15989  
15990

6455: !(FREE)

EXPEC701C:

P3,

CSPD(02)+EMIT, EMIT/125252,

!EXPECTED DATA TO BE READ OUT OF PBA AFTER LOAD:

NEXT

J/LOADBA701C

!"1010 1010 1010 1010"

(6455) DCS(0.00.0.0.0.0) BM(1010..10.10..10.10..101..010...0.0.0..0..0...0.1101...1..0000.0...11.000...100.101.110)

15991  
15992  
15993  
15994  
15995

6456: !(FREE)

LOADBA701C:

P1,

BA+ASPHI(C125252),

!LOAD BA<15:00> WITH PATTERN

NEXT

J/GOTEST701C

(6456) DCS(0.00.0.0.0.0) BM(0000..00.00..11.01..110..000...0.0.0..1..0...0.0000...0..0000.0...11.000...100.101.111)

15996  
15997  
15998  
15999  
16000

6457: !(FREE)

GOTEST701C:

SETUP,

RETURN/TEST701D,

!GO TO SUBR THAT:

NEXT

CALL(PBATOD)

!(PBA)-XOR-CSP(02)->D, BUT(D-IS-ZERO)

(6457) DCS(0.00.0.0.0.0) BM(0101..00.10..10.01..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.001)

16001  
16002  
16003  
16004  
16005  
16006

! - - - - -

!TEST-701-D CHECKS THAT BA<00> WAS LOADED WITH A (0)

5512:

TEST701D:

PO,

LOAD-ENVA(ZTARGET402),

!BIT<00> CLEAR

LOAD-ERROR(TEST701D),

!ERROR DIRECTORY KEY

DCS-CTR(C3.),

!COMPARE AT TARGET

NEXT

J/GOBUT701D

(5512) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.011.101)

16016  
16017  
16018  
16019

5435: !(FREE)

GOBUT701D:

SETUP,

RETURN/SCOPE701,

!RETURN TO SCOPE LOOP TEST WORD

KD11-K MICRO V00A-1 00:00:03 12-MAR-77

```

16020 NEXT, GOTO-PAGE(7), !BUT TABLE
16021 J/BUTBA00 !E'00'H IN BIT<00>
(5435) DCS(0.00.0.0.0.0) BM(0101..00.10..00.11..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.101.001)

```

```

16022
16023
16024
16025
16026 5436: !(FREE)
16027 SCOPE701:
16028 PO, BUMP-VERIFY, !COUNT
16029 NEXT, BUTD(SCOPE), !NO ERROR: "TEST702A" (+1. WORDS)
16030 J/TEST702A !ERROR: "MASK701A" (-12. WORDS)
(5436) DCS(0.00.0.1.0.1) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...100.111.111)

```

```

16031
16032
16033
16034
16035 ! - - - - -
16036

```

```

16037 :TEST-702-A CHECKS THAT BA<17:16> CAN BE LOADED, AND READ BACK WITH "01"
16038 : WHEN IN 18. BIT CONSOLE MODE AND KT-NO-RELOCATE MODE
16039 5477:
16040 TEST702A:
16041 PO, LOAD-ENUA(ZTARGET402), !SETUP FOR D=ZERO COMPARE
16042 LOAD-ERROR(TEST702A), !ERROR DIRECTORY KEY
16043 DCS-CTR(C13.), !COMPARE AT TARGET
16044 NEXT, J/MASK702A
(5477) DCS(1.00.1.0.0.0) BM(0010..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.111.100)

```

```

16045
16046 5474:
16047 MASK702A:
16048 P3, CSPD(04)+EMIT, EMIT/001400, !MASK TO READ ONLY BITS<9:8>
16049 NEXT, J/EXPEC702A
(5474) DCS(0.00.0.0.0.0) BM(0000..10.00..11.00..000..000...0.0.0..0..0...0.1011...1..0000.0...11.000...100.011.111)

```

```

16050
16051 5437: !(FREE)
16052 EXPEC702A:
16053 P3, CSPD(02)+EMIT, EMIT/000400, !EXPECTED DATA TO BE READ OUT OF SERVICE<9:8> AFTER LOAD:
16054 NEXT, J/SETLED702A !"0000 0001 0000 0000"
(5437) DCS(0.00.0.0.0.0) BM(0000..10.00..01.00..000..000...0.0.0..0..0...0.1101...1..0000.0...11.000...100.100.001)

```

```

16055
16056 5441: !(FREE)
16057 SETLED702A:
16058 P3, SET-CONSOLE-LED, !ENTER INTO 18. BIT MODE FOR PBA<17:16> READ
16059 NEXT, J/SETKT702A
(5441) DCS(0.00.0.0.0.0) BM(0100..00.00..00.00..100..001...0.0.0..0..0...1.1011...0..0000.0...11.000...100.100.010)

```

```

16060
16061 5442: !(FREE)
16062 SETKT702A:
16063 PO, BUMP-VERIFY, !COUNT
16064 BUSDIN+SERVICE-[I], !READ SERVICE PORT BITS<9:8>
16065 KT-NO-RELOCATE-[I], !SETUP KT FOR BA<17:16> LOADABILITY
16066 NEXT, J/LOADBA702A

```

```

16067 (5442) DCS(0.00.0.0.0.1) BM(0101..01.00..00.00..000..010...0.0.0..0..0...1.1001...0..0000.0...11.000...100.100.011)
16068 5443: !(FREE)
16069 LOADBA702A:
16070 P1, BA+BSPHI(C052525), !LOAD BA<17:16> WITH PATTERN "01"
16071 NEXT, J/GOTEST702A
(5443) DCS(0.00.0.0.0.0) BM(0000..01.11..00.00..111..000...0.0.0..1..0...0.0000...0..0000.0...11.000...100.100.100)
16072 5444: !(FREE)
16073 GOTEST702A:
16074 SETUP, RETURN/TEST702B, !GO TO SUBR THAT:
16075 NEXT, CALL(GETPROC DAT) ! (SERVICE<9:8>)-XOR-CSP(02)->D, BUT(D-IS-ZERO)
16076 (5444) DCS(0.00.0.0.0.0) BM(0110..00.11..11.10..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.111.011)
16077
16078
16079
16080
16081
16082
16083
16084 ! - - - - -
16085
16086 !TEST-702-B CHECKS THAT BA<17:16> CAN BE LOADED, AND READ BACK WITH "10"
16087 ! WHEN IN 18. BIT CONSOLE MODE AND KT-NO-RELOCATE MODE
16088 6761:
16089 TEST702B:
16090 PO, LOAD-ENVA(ZTARGET402), !SETUP FOR D=ZERO COMPARE
16091 LOAD-ERROR(TEST702B), !ERROR DIRECTORY KEY
16092 DCS-CTR(C11.), !COMPARE AT TARGET
16093 NEXT, J/EXPEC702B
(6761) DCS(1.00.1.0.0.0) BM(0100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.110.000)
16094 6460: !(FREE)
16095 EXPEC702B:
16096 P3, CSPD(02)+EMIT, EMIT/001000, !EXPECTED DATA TO BE READ OUT OF SERVICE<9:8> AFTER LOAD:
16097 NEXT, J/SETKT702B !"0000 0010 0000 0000"
16098 (6460) DCS(0.00.0.0.0.0) BM(0000..10.00..10.00..000..000...0.0.0..0..0...0.1101...1..0000.0...11.000...100.110.001)
16099 6461: !(FREE)
16100 SETKT702B:
16101 PO, BUSDIN+SERVICE-[I], !READ SERVICE PORT BITS<9:8>
16102 KT-NO-RELOCATE-[I], !SETUP KT FOR BA<17:16> LOADABILITY
16103 NEXT, J/LOADBA702B
16104 (6461) DCS(0.00.0.0.0.0) BM(0101..01.00..00.00..000..010...0.0.0..0..0...1.1001...0..0000.0...11.000...100.110.010)
16105 6462: !(FREE)
16106 LOADBA702B:
16107 PO, BUMP-VERIFY, !COUNT
16108 P1, BA+BSPHI(C125252), !LOAD BA<17:16> WITH PATTERN "10"
16109 NEXT, J/GOTEST702B
16110 (6462) DCS(0.00.0.0.0.1) BM(0000..01.11..00.00..110..000...0.0.0..1..0...0.0000...0..0000.0...11.000...100.110.011)
16111 6463: !(FREE)
16112

```

```

16113 GOTEST7029:
16114     SETUP, RETURN/SCOPE702,           !GO TO SUBR THAT:
16115     NEXT,  CALL[GETPROCAT]           ! (SERVICE(9:8))-XOR-CSP(02)->D, BUT(D-IS-ZERO)
(6463) DCS(0.00.0.0.0.0) BM(0101..00.10..00.11..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.111.011)

```

```

16116
16117
16118
16119     S433:  !(FREE)
16120     SCOPE702:
16121     P3,    CSPD(04)+EMIT, EMIT/177777, !RESET MASK FOR SUBSEQUENT TESTS
16122     NEXT,  BUTD(SCOPE),                !NO ERROR: "TEST710A" (+1. WORDS)
16123     J/TEST710A                          ! ERROR: "MASK702A" (-11. WORDS)
(5433) DCS(0.00.0.1.0.0) BM(1111..10.11..11.11..111..111...0.0.0..0..0...0.1011...1..0000.0...11.000...100.111.101)

```

16124  
16125  
16126  
16127  
16128  
16129  
16130  
16131  
16132  
16133  
16134  
16135  
16136  
16137  
16138  
16139  
16140  
16141  
16142  
16143  
16144  
16145  
16146  
16147  
16148  
16149  
16150  
16151  
16152  
16153  
16154  
16155  
16156  
16157  
16158  
16159  
16160  
16161  
16162

!.PAGE=====

.TOC \* TEST710-722: BUS FUNCTION DECODE, BUS ERROR CONDITIONS

```

!*****
!*
!* TESTS: 710A - 722C                                UWORDS: 133 + 274
!*
!* FUNCTIONS:
!*
!* WE GET SNEAKY HERE AND TEST THE FULL BUS DECODE/STATUS MUX LOGIC, AND THE BUS ERROR
!* CONDITIONS (ODD ADDRESS, INTERNAL ADDRESS, ETC) ALMOST WITHOUT EVER GOING OUT ON THE
!* BUS (WE DO ONCE, FOR SSYN TIMEOUT).
!*
!*****

```

! - - - - -

! TESTING 'DATO', 'ODD-ADDRESS' JAMUPP, 18./16. BIT IO-PAGE DECODE

! - - - - -

```

!TEST-710-A FIRST ATTEMPTS TO DO A BUS "DATO" FUNCTION, TRYING TO FORCE AN "ODD ADDRESS"
! ABORT/JAMUPP

```

```

16156 S475:
16157 TEST710A:
16158     PO,    LOAD-ENUA(4777),           !JAMUPP ADDRESS
16159     LOAD-ERROR(TEST710A),          !ERROR DIRECTORY KEY
16160     DCS-CTR(C6.),                  !COMPARE JUST AFTER BUS CYCLE UWORD, AT JAM
16161     NEXT,  J/LOADRET710A
(5475) DCS(1.00.1.0.0.0) BM(1001..00.10..01.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.000...100.111.010)

```

```

16163 5472:
16164 LOADRET710A:
16165 P3, CSPD(00)+EMIT, RETURN/TEST710B, !RETURN AFTER SUCCESSFUL JAM TO NEXT TEST
16166 NEXT, GOTO-PAGE(4) !XFER
16167 J/LOADADR710A
(5472) DCS(0.00.0.0.0.0) BM(0110..10.11..01.00..100..100...0.0.0..0..0...0.1111...1..0000.0...11.100...011.000.1111)

```

```

16168 4307: !(FREE)
16169 LOADADR710A:
16170 P3, CSPD(16)+EMIT, EMIT/160001, !"DIAGNOSTIC" UNIBUS I/O PAGE ADDRESS; ODD BYTE
16171 NEXT, J/SETJAM710A
16172 (4307) DCS(0.00.0.0.0.0) BM(1110..10.00..00.00..000..001...0.0.0..0..0...0.0001...1..0000.0...11.000...011.001.010)

```

```

16173 4312: !(FREE)
16174 SETJAM710A:
16175 P2-T, SR+CSPD(016), !SET BIT(00)=(1) FOR JAMUPP EXPECTED
16176 NEXT, J/BUSFCN710A
16177 (4312) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...0.0.1..0..0...0.0001...0..0000.0...11.000...011.001.011)

```

```

16178 4313: !(FREE)
16179 BUSFCN710A:
16180 P1, BA+SR, BSPHI(C177777), !SET BA(17:00)=(760001), ALTERED TO(160001)
16181 !SINCE WE'VE LEFT 18. BIT MODE (IN LOADING BA, THAT IS)
16182 P2-T, D+ZERO, !DO A BUS "DAT0", SHOULD GET ODD ADDRESS ABORTED
16183 P3, DAT0, !GO DELAY
16184 NEXT, J/NEXT710A
16185 (4313) DCS(0.00.0.0.0.0) BM(0011..01.11..00.00..101..000...0.1.0..1..0...1.0010...0..0000.0...11.000...100.100.101)

```

```

16186 !** AT THIS POINT JAMUPP SHOULD OCCUR **
16187 !** CLASSIC FLOW (4777) -> (4757) -> (7XXX) -> (4XXX), AND THEN WE'RE BACK HERE **
16188 !** RETURN TO ADDRESS LEFT IN CSP(00) **
16189
16190 !*** END UP HERE IF NO JAMUPP ***
16191 4445:
16192 NEXT710A:
16193 SETUP, RETURN/TEST710A, !FORCE A SCOPE LOOP ON THIS TEST, BUT FIRST
16194 NEXT, GOTO-PAGE(7), ! MUST DELAY A FEW MICROWORDS FOR BUS
16195 J/BUTD-IS-ZERO ! ERROR TO TAKE EFFECT (IGNORE "BUT" OUTCOME HERE)
16196 (4445) DCS(0.00.0.0.0.0) BM(0101..00.10..01.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

```

```

16197 !*** END UP HERE IF JAMUPP ***
16198
16199 ! - - - - -
16200
16201 !
16202 !
16203 !
16204 !TEST-710-B CHECKS THAT THE RIGHT JAM (ODD ADDRESS) IS INDICATED IN THE JAMREG:
16205 !
16206 !
16207 !
16208 !
16209 !

```

BIT:	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FCN:	000	0	SSYN	YEL	RED	WCS	POW	MEM	SSYN	CACH	ILL	MGT	RED	000	WCS	UBRK
ADR			TIME	ZON	ZON	PAR	DIS	PAR	TIME	ERR	ADR	ABT	ZON	ADR	PAR	TRAP
(101004)	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0



```

16210
16211 6644:
16212 TEST710B:
16213 PO, LOAD-ENUA(ZTARGET402), !SETUP FOR D=ZERO COMPARE
16214 LOAD-ERROR(TEST710B), !ERROR DIRECTORY KEY
16215 DCS-CTR(C10.), !COMPARE AT TARGET
16216 NEXT, J/GOTEST710B
(6644) DCS(1.00.1.0.0.0) BM(0101..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.110.100)

```

```

16217
16218 6464: !(FREE)
16219 GOTEST710B:
16220 SETUP, RETURN/TEST710C, !GO TO SUBR WHICH:
16221 NEXT, CALL(00DJAMTOD) ! (JAMREG)-XOR-(101004) -> D, BUT(D=ZERO) [000-ADDRESS]
(6464) DCS(0.00.0.0.0.0) BM(0110..00.11..01.00..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.100)

```

```

16222
16223
16224
16225 ! - - - - -
16226
16227 !TEST-710-C CHECKS THAT THE RIGHT BUS FUNCTION DECODE / PBA<17:16> ARE INDICATED IN SERVICE REG:
16228 !NOTE PBA<17:16> READ "TRUE" (OUT OF BA) WHEN IN 18. BIT CONSOLE MODE
16229
16230 BIT: 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
16231 FCN: DATI BG 0 NPR DATOB DATO PBA PBA HIB LOB TAG CON FLT POW CACH YEL
16232 (002340) 0 SERV 0 TIME 0 0 1 0 0 1 1 1 0 0 0 0 0
16233
16234
16235
16236
16237
16238
16239
16240

```

```

16235 6645:
16236 TEST710C:
16237 PO, LOAD-ENUA(ZTARGET402), !SETUP FOR D=ZERO COMPARE
16238 LOAD-ERROR(TEST710C), !ERROR DIRECTORY KEY
16239 DCS-CTR(C10.), !COMPARE AT TARGET
16240 NEXT, J/EXPEC710C
(6645) DCS(1.00.1.0.0.0) BM(0101..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.110.101)

```

```

16241
16242 6465: !(FREE)
16243 EXPEC710C:
16244 P3, CSPD(02)+EMIT, EMIT/002340, !DATO(1)H SET, PBA<17:16>="00" IN 18. BIT CONSOLE MODE
16245 NEXT, J/GOTEST710C ! IN SERVICE REG
(6465) DCS(0.00.0.0.0.0) BM(0000..10.01..00.11..100..000...0.0.0..0..0...0.1101...1..0000.0...11.000...100.110.110)

```

```

16246
16247 6466: !(FREE)
16248 GOTEST710C:
16249 SETUP, RETURN/TEST710D, !GO TO SUBR WHICH:
16250 NEXT, CALL(SERVICETOD) ! (SERVICE)-XOR-CSP(02) -> D, BUT(D-IS-ZERO)
(6466) DCS(0.00.0.0.0.0) BM(0110..00.10..11.11..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.000)

```

```

16251
16252
16253
16254 ! - - - - -
16255
16256 !TEST-710-D CHECKS THAT PBA<17:16> ARE INDICATED IN SERVICE REG:

```

KD11-K MICRO V00A-1 00:00:03 12-MAR-77

16257 ! NOTE PBA<17:16> READ AS BA<15:13>="111" WHEN IN 16. BIT MODE

16258	BIT:	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
16259	FCN:	DATI	BC	0	NPR	DATOB	DATO	PBA	PBA	HIB	LOB	TAG	CON	ELT	POW	CACH	YEL
16260			SERV	0	TIME			17	16	ERR	ERR	ERR	SERV	SERV	FAIL	ERR	ZON
16261		(003740)	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0

```

16264 6574:
16265 TEST7100:
16266     PO,          LOAD-ENUA(ZTARGET402),      !SETUP FOR D=ZERO COMPARE
16267                 LOAD-ERROR(TEST7100),      !ERROR DIRECTORY KEY
16268                 DCS-CTR(C11.),             !COMPARE AT TARGET
16269     NEXT        J/EXPEC7100
(6574) DCS(1.00.1.0.0.0) BM(0100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.110.111)

```

```

16270 6467: !(FREE)
16271 EXPEC7100:
16272     PO,          BUMP-VERIFY                !COUNT
16273                 CSPD(02)+EMIT, EMIT/003740, !DATO(1)H SET, PBA<17:16>="11" IN 16. BIT MODE
16274     P3,          J/CLEAR7100              ! IN SERVICE REG
16275     NEXT
(6467) DCS(0.00.0.0.0.0.1) BM(0000..10.01..11.11..100..000...0.0.0..0..0...0.1101...1..0000.0...11.000...100.111.000)

```

```

16276 6470: !(FREE)
16277 CLEAR7100:
16278     P3,          CLR-JAM-ERRORS-(1),        !CLEAR OUT JAM-REG, FOR USE IN NEXT TEST
16279                 CLEAR-CONSOLE-LED,        !BACK TO 16. BIT MODE
16280     NEXT        J/GOTEST7100
(6470) DCS(0.00.0.0.0.0) BM(0100..00.00..10.00..010..001...0.0.0..0..0...1.1011...0..0000.0...11.000...100.111.001)

```

```

16282 6471: !(FREE)
16283 GOTEST7100:
16284     SETUP,       RETURN/TEST710E          !GO TO SUBR WHICH:
16285     NEXT,        CALL(SERVICET00)         ! (SERVICE)-XOR-CSP(02) -> D, BUT(D-IS-ZERO)
16286     DCS(0.00.0.0.0.0) BM(0110..00.10..11.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.000)
(6471) DCS(0.00.0.0.0.0) BM(0110..00.10..11.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.000)

```

16287 ! - - - - -

16288 !TEST-710-E CHECKS THAT THE CLEAR-JAM-ERRORS CLEARED THE JAM REG TO (001000)

16289

16290

16291

```

16292 6575:
16293 TEST710E:
16294     PO,          LOAD-ENUA(ZTARGET402),      !SETUP FOR D=ZERO COMPARE
16295                 LOAD-ERROR(TEST710E),      !ERROR DIRECTORY KEY
16296                 DCS-CTR(C10.),             !COMPARE AT TARGET
16297     NEXT        J/GOTEST710E
(6575) DCS(1.00.1.0.0.0) BM(0101..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.111.010)

```

```

16299 6472: !(FREE)
16300 GOTEST710E:
16301     PO,          BUMP-VERIFY                !COUNT
16302     SETUP,       RETURN/SCOPE710,         !GO TO SUBR WHICH:
16303                 ! CSP(02) (- (001000)
16304

```

```

16305      NEXT      CALL(CLRJAMTOD)      ! (JAMREG)-XOR-CSP(02) -> D, BUT(D-IS-ZERO)
(6472) DCS(0.00.0.0.0.0.1) BM(0101..00.10..01.00..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.101)
16306
16307
16308
16309      5445:      !(FREE)
16310      SCOPE710:
16311          P2-U,      IR+EMIT, EMIT/123456,      !A "CMP-BYTE" INSTR. FOR NEXT TEST SERIES
16312          NEXT,      BUTD(SCOPE),      !NO ERROR: "TEST711A" (+1. WORDS)
16313          J/TEST711A      ! ERROR: "LOADRET710A" (-17. WORDS)
(5445) DCS(0.00.0.1.0.0) BM(1010..00.01..11.00..101..110...0.0.0..0..0...1.1010...0..0000.0...11.000...100.111.011)
16314
16315
16316
16317
16318
16319
16320      ! -----
16321      ! TESTING 'DATOB*BYTE', 'SSYN TIMEOUT' JAMUPP
16322
16323      ! -----
16324
16325      ! TEST-711-A DOES A BUS "DATOB*BYTE" FUNCTION, TRYING TO FORCE AN "SSYN TIMEOUT" ABORT/JAMUPP
16326      5473:
16327      TEST711A:
16328          PO,          LOAD-ENVA(4777),          !JAMUPP ADDRESS
16329          LOAD-ERROR(TEST711A),          !ERROR DIRECTORY KEY
16330          DCS-CTR(C4.),          !COMPARE JUST AFTER BUS CYCLE UWORD, AT JAM
16331          NEXT,      J/LOADRET711A
16332      (5473) DCS(1.00.1.0.0.0) BM(1011..00.10..01.11..111..111...0.0.0..0..0...0.000...0..0000.0...11.000...100.111.000)
16333
16334      5470:
16335      LOADRET711A:
16336          PO,          BUMP-VERIFY,          !COUNT
16337          P3,          CSPD(00)+EMIT, RETURN/TEST711B,      !RETURN AFTER SUCCESSFUL JAM TO NEXT TEST
16338          NEXT,      GOTO-PAGE(4),          !XFER
16339          NEXT,      J/BUSFCN711A
(5470) DCS(0.00.0.0.0.0.1) BM(0110..10.10..10.11..100..100...0.0.0..0..0...0.1111...1..0000.0...11.100...011.001.001)
16340
16341      4311:      !(FREE)
16342      BUSFCN711A:
16343          !NOTE:      BA=(160001) FROM PREVIOUS TEST
16344          P2-T,      SR+CSPB(B16),          !SET BIT<00>=(1) FOR JAMUPP EXPECTED
16345          P3,          DATOB,          !DO A BUS "DATOB*BYTE", SHOULD GET SSYN TIMEOUT ABORTED
16346          ! SINCE WERE USING THE "DIAGNOSTIC" UNIBUS TIMEOUT ADDR
16347          NEXT,      J/NEXT711A          !GO DELAY
(4311) DCS(0.00.0.0.0.0) BM(1010..11.01..00.00..000..000...0.0.1..0..0...1.0101...0..0000.0...11.000...100.100.110)
16348
16349      !** AT THIS POINT JAMUPP SHOULD OCCUR **
16350      !** CLASSIC FLOW (4777) -> (4757) -> (7XXX) -> (4XXX), AND THEN WE'RE BACK HERE **
16351      !** RETURN TO ADDRESS LEFT IN CSP(00) **

```

```

16352
16353 !*** END UP HERE IF NO JAMUPP ***
16354 4446:
16355 NEXT711A:
16356     SETUP, RETURN/TEST711A,           !FORCE A SCOPE LOOP ON THIS TEST, BUT FIRST
16357     NEXT,  GOTO-PAGE(7),             ! MUST DELAY A FEW MICROWORDS FOR BUS
16358     J/BUTD-IS-ZERO                   ! ERROR TO TAKE EFFECT (IGNORE "BUT" OUTCOME HERE)
(4446) DCS(0.00.0.0.0.0) BM(0101..00.10..01.11..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

```

```

16359
16360
16361 !*** END UP HERE IF JAMUPP ***
16362
16363
16364 ! - - - - -
16365

```

```

16366 !TEST-711-B CHECKS THAT THE RIGHT JAM (SSYN TIMEOUT) IS INDICATED IN THE JAMREG:
16367
16368 BIT:      15   14   13   12   11   10   09   08   07   06   05   04   03   02   01   00
16369 FCN:      ODD   0   SSYN YEL  RED  WCS  POW  MEM  SSYN CACH ILL  MGT  RED  ODD  WCS  UBRK
16370          ADR   0   TIME ZON  ZON  PAR  DIS  PAR  TIME  ERR  ADR  ABT  ZON  ADR  PAR  TRAP
16371          (021200) 0   0   1   0   0   0   1   0   1   0   0   0   0   0   0
16372
16373 6534:
16374 TEST711B:
16375     PO,      LOAD-ENUA(ZTARGET402),    !SETUP FOR D=ZERO COMPARE
16376     LOAD-ERROR(TEST711B),            !ERROR DIRECTORY KEY
16377     DCS-CTR(C10.),                   !COMPARE AT TARGET
16378     NEXT,   J/EXPEC711B
(6534) DCS(1.00.1.0.0.0) BM(0101..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.111.011)

```

```

16379
16380 6473:  !(FREE)
16381 EXPEC711B:
16382     PO,      BUMP-VERIFY,              !COUNT
16383     P3,      CSPD(02)+EMIT, EMIT/021200, !"SSYN TIMEOUT(1)H" SET
16384     NEXT,   J/GOTEST711B              ! IN JAMREG
(6473) DCS(0.00.0.0.0.1) BM(0010..10.00..10.10..000..000...0.0.0..0..0...0.1101...1..0000.0...11.000...100.111.100)

```

```

16385
16386 6474:  !(FREE)
16387 GOTEST711B:
16388     SETUP,  RETURN/TEST711C,          !GO TO SUBR WHICH:
16389     NEXT,   CALL(JAMTOD)              ! (JAMREG)-XOR-CSP(02) -> D, BUT(D-IS-ZERO)
(6474) DCS(0.00.0.0.0.0) BM(0110..00.10..10.11..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.110)

```

```

16390
16391
16392
16393 ! - - - - -
16394

```

```

16395 !TEST-711-C CHECKS THAT THE RIGHT BUS FUNCTION DECODE / PBA<17:16> ARE INDICATED IN SERVICE REG:
16396 ! BACK IN 16. BIT MODE, SINCE I-0 PAGE, PBA<17:16> READ AS "11"
16397
16398 BIT:      15   14   13   12   11   10   09   08   07   06   05   04   03   02   01   00
16399 FCN:      DATI  BG   0   NPR DATOB DATO PBA  PBA  HIB  LOB  TAG  CON  FLT  POW  CACH YEL

```

```

16400 :          SERV      TIME      17  16  ERR  ERR  ERR SERV SERV FAIL  ERR ZON
16401 :          (005740)  0    0    0    0    1    0    1    1    1    1    0    0    0    0    0
16402
16403 6530:
16404 TEST711C:
16405      PO,          LOAD-ENUA(ZTARGET402),      !SETUP FOR D=ZERO COMPARE
16406          LOAD-ERROR(TEST711C),          !ERROR DIRECTORY KEY
16407          DCS-CTR(C11.),          !COMPARE AT TARGET
16408          J/EXPEC711C
      NEXT,
(6530) DCS(1.00.1.0.0.0) BM(0100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.111.101)

16409
16410 6475: !(FREE)
16411 EXPEC711C:
16412      P3,          CSPD(02)+EMIT, EMIT/005740,      !DATOR(1)H SET, PBA<17:16>="11"
16413      NEXT,          J/GOTEST711C          ! IN SERVICE REG
(6475) DCS(0.00.0.0.0.0) BM(0000..10.10..11.11..100..000...0.0.0..0..0...0.1101...1..0000.0...11.000...100.111.110)

16414
16415 6476: !(FREE)
16416 GOTEST711C:
16417      SETUP,          RETURN/TEST7110,          !GO TO SUBR WHICH:
16418          NEXT,          CALL(CJESERVICETOD)      ! CLR-JAM-ERRORS-[1], FOR NEXT TEST
16419          ! (SERVICE)-XOR-CSP(02) -> 0, BUT(D-IS-ZERO)
(6476) DCS(0.00.0.0.0.0) BM(0110..00.10..10.10..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.111)

16420
16421
16422
16423 ! - - - - -
16424
16425 !TEST-711-D CHECKS THAT THE CLEAR-JAM-ERRORS CLEARED THE JAM REG TO (001000)
16426 6526:
16427 TEST7110:
16428      PO,          LOAD-ENUA(ZTARGET402),      !SETUP FOR D=ZERO COMPARE
16429          LOAD-ERROR(TEST7110),          !ERROR DIRECTORY KEY
16430          DCS-CTR(C10.),          !COMPARE AT TARGET
16431          BUMP-VERIFY          !COUNT
16432          J/GOTEST7110
      NEXT,
(6526) DCS(1.00.1.0.0.1) BM(0101..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.111.111)

16433
16434 6477: !(FREE)
16435 GOTEST7110:
16436      SETUP,          RETURN/SCOPE711,          !GO TO SUBR WHICH:
16437          NEXT,          CALL(CLRJAMTOD)      ! CSP(02) (- (001000)
16438          ! (JAMREG)-XOR-CSP(02) -> 0, BUT(D-IS-ZERO)
(6477) DCS(0.00.0.0.0.0) BM(0101..00.10..01.00..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.101)

16439
16440
16441
16442 5446: !(FREE)
16443 SCOPE711:
16444      NEXT,          BUTD(SCOPE),          !NO ERROR: "TEST712A" (+1. WORDS)
16445          J/TEST712A          ! ERROR: "LOADRET711A" (-11. WORDS)
(5446) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...100.111.001)

```

```

16446
16447
16448
16449
16450
16451
16452 ! -----
16453 ! TESTING 'DATI', 'INTERNAL ADDRESS' JAMUPP
16454 ! -----
16455
16456 !
16457 ! TEST-712-A DOES A BUS "DATI" FUNCTION TO AN INTERNAL ADDRESS, TRYING TO FORCE AN "INTERNAL ADDRESS" ABORT/JAMUPP
16458 5471:
16459 TEST712A:
16460     PD,      LOAD-ENUA(4777),          ! JAMUPP ADDRESS
16461             LOAD-ERROR(TEST712A),    ! ERROR DIRECTORY KEY
16462             DCS-CTR(C5.),            ! COMPARE JUST AFTER BUS CYCLE UWORD, AT JAM
16463     NEXT,    J/LOADRET712A
16464 (5471) DCS(1.00.1.0.0.0) BM(1010..00.10..01.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.000...100.110.110)

16465 5466:
16466 LOADRET712A:
16467     PD,      BUMP-VERIFY,             ! COUNT
16468     P3,      CSPD(00)+EMIT, RETURN/TEST712B, ! RETURN AFTER SUCCESSFUL JAM TO NEXT TEST
16469     NEXT,    GOTO-PAGE(4),           ! XFER
16470             J/GENADR712A
16471 (5466) DCS(0.00.0.0.0.1) BM(0101..10.10..10.00..101. 100...0.0.0..0..0...0.1111...1..0000.0...11.100...011.001.100)

16472 4314: !(FREE)
16473 GENADR712A:
16474     P2-T,    D+NOT-ASPHI(C000001),    ! MAKE D=(177776), INTERNAL ADDRESS OF PSW
16475     P3,      ASPLO(11)+0,             ! AND SAVE IT
16476     NEXT,    J/BUSFCN712A
16477 (4314) DCS(0.00.0.0.0.0) BM(0000..00.00..11.01..000..000...0.1.0..0..0...0.0000...0..0001.0...11.000...011.001.110)

16478 4316: !(FREE)
16479 BUSFCN712A:
16480     P1,      BA+ASPLO(R11),          ! BA=(177776)
16481     P2-T,    SR+BSPHI(C000001),     ! SET BIT<00>=(1) FOR JAMUPP EXPECTED
16482     P3,      DATI,                  ! DO A BUS "DATI", SHOULD GET INTERNAL ADDR ABORTED
16483     NEXT,    J/NEXT712A             ! GO DELAY
16484 (4316) DCS(0.00.0.0.0.0) BM(1010..01.11..10.01..000..000...0.0.1..1..0...1.0110...0..0000.0...11.000...100.100.111)

16485 !** AT THIS POINT JAMUPP SHOULD OCCUR **
16486 !** CLASSIC FLOW (4777) -> (4757) -> (7XXX) -> (4XXX), AND THEN WE'RE BACK HERE **
16487 !** RETURN TO ADDRESS LEFT IN CSP(00) **
16488
16489 !*** END UP HERE IF NO JAMUPP ***
16490 4447:
16491 NEXT712A:
16492     SETUP,   RETURN/TEST712A,        ! FORCE A SCOPE LOOP ON THIS TEST, BUT FIRST
16493     NEXT,    GOTO-PAGE(7),           ! MUST DELAY A FEW MICROWORDS FOR BUS
16494

```

16495 J/BUTD-IS-ZERO ! ERROR TO TAKE EFFECT (IGNORE "BUT" OUTCOME HERE)  
 (4447) DCS(0.00.0.0.0.0) BM(0101..00.10..01.11..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

16496  
 16497  
 16498 !\*\*\* END UP HERE IF JAMUPP \*\*\*  
 16499

16500  
 16501 ! - - - - -  
 16502

16503 !TEST-712-B CHECKS THAT THE RIGHT JAM (INTERNAL ADDRESS) IS INDICATED IN THE JAMREG:  
 16504 ! FOR THIS JAM, THE JAMREG SHOULD REMAIN CLEAR

BIT:	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FCN:	000	0	SSYN	YEL	RED	WCS	POW	MEM	SSYN	CACH	ILL	MGT	RED	000	WCS	UBRK
	ADR		TIME	ZON	ZON	PAR	DIS	PAR	TIME	ERR	ADR	ABT	ZON	ADR	PAR	TRAP
(001000)	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

16510  
 16511 5505:  
 16512 TEST712B:  
 16513 PO, LOAD-ENUA(ZTARGET402), !SETUP FOR D=ZERO COMPARE  
 16514 LOAD-ERROR(TEST712B), !ERROR DIRECTORY KEY  
 16515 DCS-CTR(C10.), !COMPARE AT TARGET  
 16516 NEXT, J/GOTEST712B  
 (5505) DCS(1.00.1.0.0.0) BM(0101..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.101.000)

16517  
 16518 5450: !(FREE)  
 16519 GOTEST712B:  
 16520 SETUP, RETURN/TEST712C, !GO TO SUBR WHICH:  
 16521 ! (001000) -> CSP(02)  
 16522 ! (JAMREG)-XOR-CSP(02) -> D, BUT(D-IS-ZERO)  
 (5450) DCS(0.00.0.0.0.0) BM(0110..00.10..10.10..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.101)

16523  
 16524  
 16525  
 16526 ! - - - - -  
 16527

16528 !TEST-712-C CHECKS THAT THE "OTHER-JAM-H" SIGNAL IS LOW, INDICATING AN INTERNAL ADDRESS JAM  
 16529 6524:  
 16530 TEST712C:

PO,	LOAD-ENUA(ZTARGET401),	!BIT<01> CLEAR
	LOAD-ERROR(TEST712C),	!ERROR DIRECTORY KEY
	DCS-CTR(C3.),	!COMPARE AT TARGET
NEXT,	J/GOBUT712C	

16531  
 16532  
 16533  
 16534 (6524) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...101.000.000)

16535  
 16536 6500: !(FREE)  
 16537 GOBUT712C:  
 16538 SETUP, RETURN/TEST712D, !RETURN TO START OF NEXT SUBTEST  
 16539 NEXT, GOTO-PAGE(7), !BUT TABLE  
 16540 J/BUTOTHERJAM !"OTHER JAM H" IN BIT<01>  
 (6500) DCS(0.00.0.0.0.0) BM(0110..00.10..10.10..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.101.010)

16541  
16542  
16543  
16544  
16545  
16546  
16547  
16548  
16549  
16550  
16551  
16552  
16553  
16554  
16555  
16556  
16557  
16558  
16559  
  
16560  
16561  
16562  
16563  
16564  
  
16565  
16566  
16567  
16568  
16569  
16570  
  
16571  
16572  
16573  
16574  
16575  
16576  
16577  
16578  
  
16579  
16580  
16581  
16582  
16583  
16584  
16585  
16586  
16587  
16588  
16589

! - - - - -  
! TEST-712-D CHECKS THAT THE RIGHT BUS FUNCTION DECODE / PBA<17:16> ARE INDICATED IN SERVICE REG:  
! BACK IN 16. BIT MODE, SINCE I-O PAGE, PBA<17:16> READ AS "11"

BIT:	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FCN:	DATI	BG	0	NPR	DATOB	DATO	PBA	PBA	HIB	LOB	TAG	CON	FLT	POW	CACH	YEL
		SERV	TIME				17	16	ERR	ERR	ERR	SERV	SERV	FAIL	ERR	ZON
(101740)	1	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0

6523:

TEST7120:

PO, LOAD-ENVA(ZTARGET402), ! SETUP FOR D=ZERO COMPARE  
LOAD-ERROR(TEST7120), ! ERROR DIRECTORY KEY  
DCS-CTR(C11.), ! COMPARE AT TARGET  
J/EXPEC7120

(6523) DCS[1.00.1.0.0.0] BM[0100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...101.000.001]

6501: !(FREE)

EXPEC7120:

P3, CSPO[02]+EMIT, EMIT/101740, ! DATI(1)H SET, PBA<17:16>="11"  
J/GOTEST7120 ! IN SERVICE REG

(6501) DCS[0.00.0.0.0.0] BM[1000..10.00..11.11..100..000...0.0.0..0..0...0.1101...1..0000.0...11.000...101.000.010]

6502: !(FREE)

GOTEST7120:

SETUP, RETURN/SCOPE712, ! GO TO SUBR WHICH:  
! CLR-JAM-ERRORS-[1], FOR NEXT TEST  
NEXT, CALL[CJESERVICET00] ! (SERVICE)-XOR-CSP(02) -> 0, BUT(D-IS-ZERO)

(6502) DCS[0.00.0.0.0.0] BM[0101..00.10..01.00..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.111]

5447: !(FREE)

SCOPE712:

NEXT, BUTD(SCOPE), ! NO ERROR: "TEST713A" (+1. WORDS)  
J/TEST713A ! ERROR: "LOADRET712A" (-12. WORDS)

(5447) DCS[0.00.0.1.0.0] BM[0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...100.110.111]

! - - - - -  
! TESTING 'DATI-NOINT', 'ILLEGAL INTERNAL ADDRESS' JAMUPP  
! - - - - -



```

16590
16591 !TEST-713-A DOES A BUS "DATI-NOINT" FUNCTION TO AN INTERNAL ADDRESS, TRYING TO FORCE
16592 ! AN "ILLEGAL INTERNAL ADDRESS" ABORT/JAMUPP
16593 5467:
16594 TEST713A:
16595     PO,          LOAD-ENUA(4777),          !JAMUPP ADDRESS
16596                LOAD-ERROR(TEST713A),     !ERROR DIRECTORY KEY
16597                DCS-CTR(C4.),             !COMPARE JUST AFTER BUS CYCLE UNORD, AT JAM
16598                BUMP-VERIFY,              !COUNT
16599     NEXT,        J/LOADRET713A

```

```

(5467) DCS(1.00.1.0.0.1) BM(1011..00.10..01.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.000...100.110.100)

```

```

16600
16601 5464:
16602 LOADRET713A:
16603     P3,          CSPD(00)+EMIT, RETURN/TEST713B, !RETURN AFTER SUCCESSFUL JAM TO NEXT TEST
16604     NEXT,        GOTO-PAGE(7),              !XFER
16605                J/BUSFCN713A

```

```

(5464) DCS(0.00.0.0.0.0) BM(0101..10.11..00.01..110..111...0.0.0..0..0...0.1111...1..0000.0...11.100...001.110.111)

```

```

16606
16607 7167: !(FREE)
16608 BUSFCN713A:
16609     P1,          BA+ASPLO(R11),             !BA=(177776), GENERATED IN PREVIOUS SET OF TESTS
16610     P2-T,        SR+BSPHI(C000001),        !SET BIT(00)=(1) FOR JAMUPP EXPECTED
16611     P3,          DATI-NOINT,                !DO A BUS "DATI-NOINT"
16612                ! SHOULD GET "ILLEGAL INTERNAL ADDR" ABORTED
16613     NEXT,        J/NEXT713A                !GO DELAY

```

```

(7167) DCS(0.00.0.0.0.0) BM(1010..01.11..10.01..000..000...0.0.1..1..0...1.0001...0..0000.0...11.000...011.110.101)

```

```

16614
16615 !** AT THIS POINT JAMUPP SHOULD OCCUR **
16616 !** CLASSIC FLOW (4777) -> (4757) -> (7XXX) -> (4XXX), AND THEN WE'RE BACK HERE **
16617 !** RETURN TO ADDRESS LEFT IN CSP(00) **

```

```

16618
16619 !*** END UP HERE IF NO JAMUPP ***
16620 7365:
16621 NEXT713A:
16622     SETUP,       RETURN/TEST713A,          !FORCE A SCOPE LOOP ON THIS TEST, BUT FIRST
16623     NEXT,        GOTO-PAGE(7),              ! MUST DELAY A FEW MICROWORDS FOR BUS
16624                J/BUTD-IS-ZERO            ! ERROR TO TAKE EFFECT (IGNORE "BUT" OUTCOME HERE)

```

```

(7365) DCS(0.00.0.0.0.0) BM(0101..00.10..01.10..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

```

```

16625
16626
16627 !*** END UP HERE IF JAMUPP ***

```

```

16628
16629
16630 ! - - - - -
16631
16632 !TEST-713-B CHECKS THAT THE RIGHT JAM (ILLEGAL INTERNAL ADDRESS) IS INDICATED IN THE JAMREG:

```

```

16633
16634 BIT:      15   14   13   12   11   10   09   08   07   06   05   04   03   02   01   00
16635 FCN:      ODD   0   SSYN YEL   RED   WCS   POW   MEM   SSYN CACH ILL   MGT   RED   ODD   WCS   UBRK
16636          ADR   0   TIME ZON   ZON   PAR   DIS   PAR   TIME  ERR  ADR   ABT   ZON   ADR   PAR   TRAP
16637 (001040)  0   0   0   0   0   0   1   0   0   0   1   0   0   0   0
16638

```

```

16639 5616:
16640 TEST7138:
16641     PO,          LOAD-ENUA(ZTARGET402),      !SETUP FOR D=ZERO COMPARE
16642     LOAD-ERROR(TEST7138),      !ERROR DIRECTORY KEY
16643     DCS-CTR(C10.),             !COMPARE AT TARGET
16644     BUMP-VERIFY,              !COUNT
16645     NEXT          J/EXPEC7138
(5616) DCS(1.00.1.0.0.1) BM(0101..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.101.010)

```

```

16646 5452: !(FREE)
16647 EXPEC7138:
16648     P3,          CSPD(02)+EMIT, EMIT/001040,      !"ILLEGAL INTERNAL ADDRESS(1)H" SET
16649     NEXT        J/GOTEST7138
(5452) DCS(0.00.0.0.0.0) BM(0000..10.00..10.00..100..000...0.0.0..0..0...0.1101...1..0000.0...11.000...100.101.011)

```

```

16651 5453: !(FREE)
16652 GOTEST7138:
16653     SETUP,      RETURN/TEST713C,                !GO TO SUBR WHICH:
16654     NEXT        CALL(JAMT00)                    ! (JAMREG)-XOR-CSP(02) -> 0, BUT(D-IS-ZERO)
(5453) DCS(0.00.0.0.0.0) BM(0110..00.10..10.10..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.110)

```

```

16656 ! - - - - -
16657
16658 !TEST-713-C CHECKS THAT THE "OTHER-JAM-H" SIGNAL IS HIGH, INDICATING A JAM PRESENT OTHER
16659 ! THAN ONLY A VALID "INTERNAL ADDRESS" JAM
16660
16661 6522:
16662 TEST713C:
16663     PO,          LOAD-ENUA(ZTARGET403),      !BIT<01> SET
16664     LOAD-ERROR(TEST713C),      !ERROR DIRECTORY KEY
16665     DCS-CTR(C3.),             !COMPARE AT TARGET
16666     NEXT        J/GOBUT713C
(6522) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...101.000.011)

```

```

16669 6503: !(FREE)
16670 GOBUT713C:
16671     SETUP,      RETURN/TEST713C,                !RETURN TO START OF NEXT SUBTEST
16672     NEXT        GOTO-PAGE(7),                    !BUT TABLE
16673     J/BUTOTHER)QA                            !"OTHER JAM H" IN BIT<01>
(6503) DCS(0.00.0.0.0.0) BM(0110..00.10..10.10..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.101.010)

```

```

16675 ! - - - - -
16676
16677 !TEST-713-D CHECKS THAT THE RIGHT BUS FUNCTION DECODE / PBA<17:16> ARE INDICATED IN SERVICE REG:
16678 ! BACK IN 16. BIT MODE, SINCE I-0 PAGE, PBA<17:16> READ AS "11"
16679
16680 BIT:      15   14   13   12   11   10   09   08   07   06   05   04   03   02   01   00
16681 FCN:      DATI  BG    0   NPR  DATOB  DATO  PBA  PBA  HIB  LOB  TAG  CON  FLT  POW  CACH  YEL
16682          SERV          TIME          17  16  ERR  ERR  ERR  SERV  SERV  FAIL  ERR  ZON
16683
16684
16685

```

KD11-K MICRO V00A-1 00:00:03 12-MAR-77

16686 ! (101740) 1 0 0 0 0 0 1 1 1 1 1 0 0 0 0

```

16687
16688 6521:
16689 TEST7130:
16690     PO,          LOAD-ENUA(ZTARGET402),      !SETUP FOR D=ZERO COMPARE
16691                LOAD-ERROR(TEST7130),        !ERROR DIRECTORY KEY
16692                DCS-CTR(C11.),              !COMPARE AT TARGET
16693                BUMP-VERIFY,                !COUNT
16694                J/EXPEC7130
(6521) DCS(1.00.1.0.0.1) BM(0100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...101.000.100)

```

```

16695
16696 6504: !(FREE)
16697 EXPEC7130:
16698     P3,          CSPD(02)+EMIT, EMIT/101740,  !DATI(1)H SET, PBA<17:16>="11"
16699     NEXT,        J/GOTEST7130              ! IN SERVICE REG
(6504) DCS(0.00.0.0.0.0) BM(1000..10.00..11.11..100..000...0.0.0..0..0...0.1101...1..0000.0...11.000...101.000.101)

```

```

16700
16701 6505: !(FREE)
16702 GOTEST7130:
16703     SETUP,       RETURN/SCOPE713,          !GO TO SUBR WHICH:
16704     NEXT,        CALL(CJESERVICETOD)      ! CLR-JAM-ERRORS-[I], FOR NEXT TEST
16705                ! (SERVICE)-XOR-CSP(02) -> 0, BUT(D-IS-ZERO)
(6505) DCS(0.00.0.0.0.0) BM(0101..00.10..01.01..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.111)

```

```

16706
16707
16708
16709
16710 5451: !(FREE)
16711 SCOPE713:
16712     NEXT,        BUTD(SCOPE),              !NO ERROR: "TEST720A" (+1. WORDS)
16713                J/TEST720A                ! ERROR: "LOADRET713A" (-11. WORDS)
(5451) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...100.110.101)

```

```

16714
16715
16716
16717
16718
16719
16720
16721 ! -----
16722 ! TESTING 'DATIB*-BYTE', '000-ADDRESS' JAMUPP, I/O-PAGE-PBA<17:16> DECODE
16723 ! -----
16724
16725
16726

```

```

16727 !TEST-720-A ATTEMPTS TO DO A BUS "DATIB*-BYTE" FUNCTION, TRYING TO FORCE AN "000 ADDRESS"
16728 ! ABORT/JAMUPP, ALSO CHECKING THAT ALL SERVICE / JAM STATUS BITS SET CORRECTLY
16729 5465:
16730 TEST720A:
16731     PO,          LOAD-ENUA(4777),          !JAMUPP ADDRESS
16732                LOAD-ERROR(TEST720A),      !ERROR DIRECTORY KEY
16733                DCS-CTR(C7.)              !COMPARE JUST AFTER BUS CYCLE UWORD, AT JAM
16734     NEXT,        J/LOADRET720A

```

```

(5465) DCS(1.00.1.0.0.0) BM(1000..00.10..01.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.000...100.110.010)
16735
16736 5462:
16737 LOADRET720A:
16738 P3, CSPD(00)+EMIT, RETURN/TEST720B, !RETURN AFTER SUCCESSFUL JAM TO NEXT TEST
16739 NEXT, GOTO-PAGE(7), !XFER
16740 J/LOADIR720A

(5462) DCS(0.00.0.0.0.0) BM(0110..10.10..10.10..000..111...0.0.0..0..0...0.1111...1..0000.0...11.100...001.111.010)
16741
16742 7172: !(FREE)
16743 LOADIR720A:
16744 P2-U, IR+EMIT, EMIT/000000, !"HALT" INSTRUCTION IS -BYTE
16745 NEXT, J/SETADR720A

(7172) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...1.010...0..0000.0...11.000...001.111.100)
16746
16747 7174: !(FREE)
16748 SETADR720A:
16749 P3, CSPD(16)+EMIT, EMIT/060001, !BIT<15:13>="011", -IOPAGE; ODD-ADDRESS
16750 NEXT, J/SETJAM720A

(7174) DCS(0.00.0.0.0.0) BM(0110..10.00..00.00..000..001...0.0.0..0..0...0.0001...1..0000.0...11.000...001.111.101)
16751
16752 7175: !(FREE)
16753 SETJAM720A:
16754 P2-T, SR+CSPD(D16), !SET BIT<00>=(1) FOR JAMUPP EXPECTED
16755 NEXT, J/BUSFCN720A

(7175) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...0.0.1..0..0...0.0001...0..0000.0...11.000...001.111.110)
16756
16757 7176: !(FREE)
16758 BUSFCN720A:
16759 P1, BA+SR, !SET BA<17:00>=(060001)
16760 P3, DATIB, !DO A BUS "DATIB*-BYTE", SHOULD GET ODD ADDRESS ABORTED
16761 NEXT, J/NEXT720A !GO DELAY

(7176) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..1..0...1.0011...0..0000.0...11.000...011.110.100)
16762
16763 !** AT THIS POINT JAMUPP SHOULD OCCUR **
16764 !** CLASSIC FLOW (4777) -> (4757) -> (7XXX) -> (4XXX), AND THEN WE'RE BACK HERE **
16765 !** RETURN TO ADDRESS LEFT IN CSP(00) **
16766
16767 !*** END UP HERE IF NO JAMUPP ***
16768 7364:
16769 NEXT720A:
16770 SETUP, RETURN/TEST720A, !FORCE A SCOPE LOOP ON THIS TEST, BUT FIRST
16771 NEXT, GOTO-PAGE(7), ! MUST DELAY A FEW MICROWORDS FOR BUS
16772 J/BUTD-IS-ZERO ! ERROR TO TAKE EFFECT (IGNORE "BUT" OUTCOME HERE)

(7364) DCS(0.00.0.0.0.0) BM(0101..00.10..01.10..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)
16773
16774 !*** END UP HERE IF JAMUPP ***
16775
16776 ! - - - - -
16777
16778 !
16779
16780 !TEST-720-B CHECKS THAT THE RIGHT JAM (ODD ADDRESS) IS INDICATED IN THE JAMREG:

```

16781	:																	
16782	:	BIT:	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
16783	:	FCN:	000	0	SSYN	YEL	RED	WCS	POW	MEM	SSYN	CACH	ILL	MGT	RED	000	WCS	UBRK
16784	:		ADR		TIME	ZON	ZON	PAR	DIS	PAR	TIME	ERR	ADR	ABT	ZON	ADR	PAR	TRAP
16785	:	(101004)	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0

```

16786
16787 6520:
16788 TEST720B:
16789 PO, LOAD-ENUA(ZTARGET402), !SETUP FOR D=ZERO COMPARE
16790 LOAD-ERROR(TEST720B), !ERROR DIRECTORY KEY
16791 DCS-CTR(C10.), !COMPARE AT TARGET
16792 BUMP-VERIFY, !COUNT
16793 NEXT J/GOTEST720B
(6520) DCS(1.00.1.0.0.1) BM(0101..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...101.000.110)

```

```

16794
16795 6506: !(FREE)
16796 GOTEST720B:
16797 SETUP, RETURN/TEST720C, !GO TO SUBR WHICH:
16798 NEXT CALL(000JAMT00) ! (JAMREG)-XOR-(101004) -> D, BUT(D=ZERO) (000-ADDRESS)
(6506) DCS(0.00.0.0.0.0) BM(0100..00.11..11.10..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.100)

```

```

16800
16801
16802 ! - - - - -
16803
16804 !TEST-720-C CHECKS THAT THE RIGHT BUS FUNCTION DECODE / PBA<17:16> ARE INDICATED IN SERVICE REG:
16805 NOTE: PBA<17:16> SHOULD NOT BE FORCED TO "11", IN 16. BIT MODE, WHEN BA<15:13>="011"
16806
16807 BIT: 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
16808 FCN: DATI BG 0 NPR DATOB DATO PBA PBA HIB LOB TAG CON FLT POW CACH YEL
16809 SERV TIME 17 16 ERR ERR ERR SERV SERV FAIL ERR ZON
16810 (100340) 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0

```

```

16811
16812 4760:
16813 TEST720C:
16814 PO, LOAD-ENUA(ZTARGET402), !SETUP FOR D=ZERO COMPARE
16815 LOAD-ERROR(TEST720C), !ERROR DIRECTORY KEY
16816 DCS-CTR(C11.), !COMPARE AT TARGET
16817 NEXT J/GOTEST720C
(4760) DCS(1.00.1.0.0.0) BM(0100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.001.111)

```

```

16818
16819 4317: !(FREE)
16820 GOTEST720C:
16821 SETUP, RETURN/TEST721A, !GO TO SUBR WHICH:
16822 NEXT CALL(DATISERVICET00) ! CLR-JAM-ERRORS, TO RESET FOR NEXT TEST
16823 (4317) DCS(0.00.0.0.0.0) BM(0100..00.10..11.11..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.101)

```

16824  
16825  
16826  
16827  
16828

```

16829
16830
16831 !-----
16832
16833 !TESTING 'DATOB*-BYTE', 'ODD-ADDRESS' JAMUPP, I/O-PAGE-PBA<17:16> DECODE
16834
16835 !-----
16836
16837 !TEST-721-A ATTEMPTS TO DO A BUS "DATOB*-BYTE" FUNCTION, TRYING TO FORCE AN "ODD ADDRESS"
16838 ! ABORT/JAMUPP, ALSO CHECKING THAT ALL SERVICE / JAM STATUS BITS SET CORRECTLY
16839 7571:
16840 TEST721A:
16841     PO,          LOAD-ENUA(4777),          !JAMUPP ADDRESS
16842     LOAD-ERROR(TEST721A),          !ERROR DIRECTORY KEY
16843     DCS-CTR(C6.)          !COMPARE JUST AFTER BUS CYCLE UWORD, AT JAM
16844     NEXT,        J/LOADRET721A
(4571) DCS[1.00.1.0.0.0] BM[1001..00.10..01.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.000...011.010.000]

16845
16846 4320: !(FREE)
16847 LOADRET721A:
16848     P3,          CSPD[00]+EMIT, RETURN/TEST721B,          !RETURN AFTER SUCCESSFUL JAM TO NEXT TEST
16849     NEXT,        GOTO-PAGE(7),          !XFER
16850     NEXT,        J/SETADR721A
(4320) DCS[0.00.0.0.0.0] BM[0100..10.10..11.11..011..111...0.0.0..0..0...0.1111...1..0000.0...11.100...001.111.011]

16851
16852 7173: !(FREE)
16853 SETADR721A:
16854     P3,          CSPD[16]+EMIT, EMIT/120001,          !BIT<15:13>="101", -IOPAGE; ODD-ADDRESS
16855     NEXT,        J/SETJAM721A
(7173) DCS[0.00.0.0.0.0] BM[1010..10.00..00.00..000..001...0.0.0..0..0...0.0001...1..0000.0...11.000...010.000.000]

16856
16857 7200: !(FREE)
16858 SETJAM721A:
16859     P2-1,        SR+CSPD(D16),          !SET BIT<00>=(1) FOR JAMUPP EXPECTED
16860     NEXT,        J/BUSFCN721A
(7200) DCS[0.00.0.0.0.0] BM[1010..10.00..00.00..000..000...0.0.1..0..0...0.0001...0..0000.0...11.000...010.000.001]

16861
16862 7201: !(FREE)
16863 BUSFCN721A:
16864
16865     P1,          BA+SR,          !NOTE: "IR"=(000000)=-BYTE FROM PREV TEST
16866     P3,          DATOB,          !SET BA<17:00>=(120001)
16867     NEXT,        J/NEXT721A          !DO A BUS "DATOB*-BYTE", SHOULD GET ODD ADDRESS ABORTED
!GO DELAY
(7201) DCS[0.00.0.0.0.0] BM[0000..00.00..00.00..000..000...0.0.0..1..0...1.0101...0..0000.0...11.000...011.110.010]

16868
16869 !** AT THIS POINT JAMUPP SHOULD OCCUR **
16870 !** CLASSIC FLOW (4777) -> (4757) -> (7XXX) -> (4XXX), AND THEN WE'RE BACK HERE **
16871 !** RETURN TO ADDRESS LEFT IN CSP(00) **
16872
16873 !*** END UP HERE IF NO JAMUPP ***
16874 7362:
16875 NEXT721A:
16876     SETUP, RETURN/TEST721A,          !FORCE A SCOPE LOOP ON THIS TEST, BUT FIRST

```

16877 NEXT, GOTO-PAGE(7), ! MUST DELAY A FEW MICROWORDS FOR BUS  
 16878 J/BUTD-IS-ZERO ! ERROR TO TAKE EFFECT (IGNORE "BUT" OUTCOME HERE)  
 (7362) DCS(0.00.0.0.0.0) BM(0100..00.10..11.11..001..111...0.0.0..0.0...0.0000...0..0000.0...11.100...011.100.001)

!\*\*\* END UP HERE IF JAMUPP \*\*\*

! - - - - -

!TEST-721-B CHECKS THAT THE RIGHT JAM (ODD ADDRESS) IS INDICATED IN THE JAMREG:

BIT:	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FCN:	ODD	0	SSYN	YEL	RED	WCS	POW	MEM	SSYN	CACH	ILL	MGT	RED	ODD	WCS	UBRK
	ADR		TIME	ZON	ZON	PAR	DIS	PAR	TIME	ERR	ADR	ABT	ZON	ADR	PAR	TRAP
(101004)	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0

4573:  
 TEST721B:  
 PO, LOAD-ENUA(ZTARGET402), !SETUP FOR D=ZERO COMPARE  
 LOAD-ERROR(TEST721B), !ERROR DIRECTORY KEY  
 DCS-CTR(C10.), !COMPARE AT TARGET  
 NEXT, J/GOTEST721B

(4573) DCS(1.00.1.0.0.0) BM(0101..00.11..11.00..000..010...0.0.0..0.0...0.0000...0..0000.0...11.000...011.010.001)

4321: !(FREE)  
 GOTEST721B:

16900 SETUP, RETURN/TEST721C, !GO TO SUBR WHICH:  
 16901 NEXT, CALL(000JAMTOD) ! (JAMREG)-XOR-(101004) -> D, BUT(D=ZERO) [000-ADDRESS]  
 (4321) DCS(0.00.0.0.0.0) BM(0100..00.10..11.11..111..111...0.0.0..0.0...0.0000...0..0000.0...11.100...010.101.100)

! - - - - -

!TEST-721-C CHECKS THAT THE RIGHT BUS FUNCTION DECODE / PBA<17:16> ARE INDICATED IN SERVICE REG:  
 NOTE: PBA<17:16> SHOULD NOT BE FORCED TO "11", IN 16. BIT MODE, WHEN BA<15:13>="101"

BIT:	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FCN:	DATI	BG	0	NPR	DATOB	DATO	PBA	PBA	HIB	LOB	TAG	CON	FLT	POW	CACH	YEL
	SERV		TIME				17	16	ERR	ERR	ERR	SERV	SERV	FAIL	ERR	ZON
(002340)	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0

4577:  
 TEST721C:  
 PO, LOAD-ENUA(ZTARGET402), !SETUP FOR D=ZERO COMPARE  
 LOAD-ERROR(TEST721C), !ERROR DIRECTORY KEY  
 DCS-CTR(C11.), !COMPARE AT TARGET  
 NEXT, J/GOTEST721C

(4577) DCS(1.00.1.0.0.0) BM(0100..00.11..11.00..000..010...0.0.0..0.0...0.0000...0..0000.0...11.000...011.010.010)

4322: !(FREE)  
 GOTEST721C:

16923  
 16924  
 16925

```

16926          SETUP, RETURN/SCOPE721,          !GO TO SUBR WHICH:
16927          ! CLR-JAM-ERRORS, TO RESET FOR NEXT TEST
16928          NEXT, CALL(DATOSERVICET00)         ! (SERVICE)-XOR-(002340) -> D, BUT(D=ZERO) (DATO(11H))
(4322) DCS(0.00.0.0.0.0) BM(0101..00.10..01.01..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.110)

16929
16930
16931
16932
16933          5454: !(FREE)
16934          SCOPE721:
16935          NEXT, BUTD(SCOPE),                  !NO ERROR: "TEST722A" (+1. WORDS)
16936          J/TEST722A                          ! ERROR: "LOADRET720A" (-24. WORDS)
(5454) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...100.110.011)

16937
16938
16939
16940
16941
16942          ! -----
16943          ! TESTING 'INVALIDATE', '000-ADDRESS' JAMUPP, I/O-PAGE-PBA<17:16> DECODE
16944          ! -----
16945          !
16946          !
16947          ! TEST-722-A ATTEMPTS TO DO A BUS "INVALIDATE" FUNCTION, TRYING TO FORCE AN "000 ADDRESS"
16948          ! ABORT/JAMUPP, ALSO CHECKING THAT ALL SERVICE / JAM STATUS BITS SET CORRECTLY
16949          !
16950          5463:
16951          TEST722A:
16952          PO, LOAD-ENUA(4777),                  ! JAMUPP ADDRESS
16953          LOAD-ERROR(TEST722A),                ! ERROR DIRECTORY KEY
16954          DCS-CTR(C6.),                        ! COMPARE JUST AFTER BUS CYCLE UWORD, AT JAM
16955          NEXT, J/LOADRET722A
(5463) DCS(1.00.1.0.0.0) BM(1001..00.10..01.11..111..111...0.0.0..0..0...0.0000...0..0000.0...11.000...100.101.110)

16956
16957          5456: !(FREE)
16958          LOADRET722A:
16959          P3, CSPD(00)+EMIT, RETURN/TEST722B, ! RETURN AFTER SUCCESSFUL JAM TO NEXT TEST
16960          NEXT, GOTO-PAGE(4),                 ! XFER
16961          NEXT, J/SETADR722A
(5456) DCS(0.00.0.0.0.0) BM(0100..10.10..01.10..100..100...0.0.0..0..0...0.1111...1..0000.0...11.100...100.111.000)

16962
16963          4470:
16964          SETADR722A:
16965          P3, CSPD(16)+EMIT, EMIT/140001,     ! BIT<15:13>="110", -IOPAGE; 000-ADDRESS
16966          NEXT, J/SETJAM722A
(4470) DCS(0.00.0.0.0.0) BM(1100..10.00..00.00..000..001...0.0.0..0..0...0.0001...1..0000.0...11.000...011.010.011)

16967
16968          4323: !(FREE)
16969          SETJAM722A:
16970          P2-T, SR+CSPD(016),                ! SET BIT<00>=(1) FOR JAMUPP EXPECTED
16971          NEXT, J/BUSFCN722A
(4323) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000.000...0.0.1..0..0...0.0001...0..0000.0...11.000...011.010.100)

```



```

16972
16973 4324: !(FREE)
16974 BUSFCN722A:
16975 P1, BA+SR, !SET BA(17:00)=(140001)
16976 P3, INVALIDATE, !DO A BUS "INVALIDATE", SHOULD GET ODD ADDRESS ABORTED
16977 NEXT, J/NEXT722A, !GO DELAY
(4324) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..1..0...1.0111...0..0000.0...11.000...101.010.100)

```

```

16978 !** AT THIS POINT JAMUPP SHOULD OCCUR **
16979 !** CLASSIC FLOW (4777) -> (4757) -> (7XXX) -> (4XXX), AND THEN WE'RE BACK HERE **
16980 !** RETURN TO ADDRESS LEFT IN CSP(00) **
16981
16982
16983 !*** END UP HERE IF NO JAMUPP ***

```

```

16984 4524:
16985 NEXT722A:
16986 SETUP, RETURN/TEST722A, !FORCE A SCOPE LOOP ON THIS TEST, BUT FIRST
16987 NEXT, GOTO-PAGE(7), ! MUST DELAY A FEW MICROWORDS FOR BUS
16988 J/BUTD-IS-ZERO, ! ERROR TO TAKE EFFECT (IGNORE "BUT" OUTCOME HERE)
(4524) DCS(0.00.0.0.0.0) BM(0101..00.10..01.10..011..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

```

```

16989 !*** END UP HERE IF JAMUPP ***
16990
16991
16992
16993
16994 ! - - - - -
16995

```

```

16996 !TEST-722-B CHECKS THAT THE RIGHT JAM (ODD ADDRESS) IS INDICATED IN THE JAMREG:

```

BIT:	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FCN:	000	0	SSYN	YEL	RED	WCS	POW	MEM	SSYN	CACH	ILL	MGT	RED	000	WCS	UBRK
	ADR		TIME	ZON	ZON	PAR	DIS	PAR	TIME	ERR	ADR	ABT	ZON	ADR	PAR	TRAP
(101004)	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0

```

17003 4464:
17004 TEST722B:
17005 PO, LOAD-ENJA(ZTARGET402), !SETUP FOR D=ZERO COMPARE
17006 LOAD-ENOR(TEST722B), !ERROR DIRECTORY KEY
17007 DCS-CTR(C11.), !COMPARE AT TARGET
17008 BUMP-VERIFY, !COUNT
17009 NEXT, J/GOTEST722B
(4464) DCS(1.00.1.0.0.1) BM(0100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.010.101)

```

```

17010 4325: !(FREE)
17011 GOTEST722B:
17012 SETUP, RETURN/TEST722C, !GO TO SUBR WHICH:
17013 NEXT, CALL(ODDJAMT00), ! (JAMREG)-XOR-(101004) -> D, BUT(D=ZERO) [ODD-ADDRESS]
17014 (4325) DCS(0.00.0.0.0.0) BM(0100..00.10..01.10..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.100)

```

```

17015 ! - - - - -
17016
17017
17018
17019

```

```

17020 !TEST-722-C CHECKS THAT THE RIGHT BUS FUNCTION DECODE / PBA<17:16> ARE INDICATED IN SERVICE REG:
17021 ! NOTE: PBA<17:16> SHOULD NOT BE FORCED TO "11", IN 16. BIT MODE, WHEN BA<15:13>="110"
17022
17023 BIT:      15    14    13    12    11    10    09    08    07    06    05    04    03    02    01    00
17024 FCN:     DATA  BG     0  NPR  DATOR DATOR PBA   PBA   HIB   LOB   TAG   CON   FLT   POW   CACH YEL
17025          SERV          TIME          17   16   ERR   ERR   ERR  SERV  SERV  FAIL  ERR  ZON
17026 (002340)  0     0     0     0     0     1     0     0     1     1     1     0     0     0     0     0
17027
17028 4465:
17029 TEST722C:
17030 PO,      LOAD-ENVA(ZTARGET402),      !SETUP FOR D=ZERO COMPARE
17031          LOAD-ERROR(TEST722C),      !ERROR DIRECTORY KEY
17032          DCS-CTR(C11.),             !COMPARE AT TARGET
17033          J/GOTEST722C
17034 (4465) DCS(1.00.1.0.0.0) BM(0100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.010.110)
17035
17036 4326: !(FREE)
17037 GOTEST722C:
17038          SETUP, RETURN/SCOPE722,      !GO TO SUBR WHICH:
17039          NEXT, CALL(DATOSERVICET00)    ! CLR-JAM-ERRORS, TO RESET FOR NEXT TEST
17040          ! (SERVICE)-XOR-(002340) -> 0, BUT(D=ZERO) (DATO[1]H)
17041 (4326) DCS(0.00.0.0.0.0) BM(0100..00.01..10.10..111..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.110)
17042
17043
17044 4327: !(FREE)
17045 SCOPE722:
17046          NEXT, BUTD(SCOPE),           !NO ERROR: "TEST730A" (+1. WORDS)
17047          J/TEST730A                   ! ERROR: "SETADR720A" (-11. WORDS)
17048 (4327) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...100.111.001)
17049
17050
17051
17052
17053 !.PAGE=====
17054
17055 .TOC * TEST730-731: BUS CYCLES TO/FROM MEMORY
17056
17057 *****
17058 !*
17059 !* TESTS: 730A - 731E                                UWORDS: 063 + 057
17060 !*
17061 !* FUNCTIONS:
17062 !*
17063 !* THESE TESTS CHECK THAT ACTUAL BUS CYCLES CAN BE CORRECTLY EXECUTED.
17064 !*
17065 !*
17066 !*****
17067
17068
17069

```

```

17070
17071
17072
17073 ! -----
17074 ! THIS FIRST SERIES OF TESTS DOES A DATO/DATIP/DATO/DATIB SEQUENCE, CHECKING THAT
17075 ! EACH FUNCTION OPERATES AS EXPECTED.
17076 ! -----
17077
17078 !
17079 !
17080 ! TEST-730-A DOES A DATO, AND THEN CHECKS THAT THE DBUF LATCH (DS) ALSO GETS LOADED WITH THE
17081 ! DATA, AND THAT IT IS ENABLED ON BUSDIN IN THE MICROWORD AFTER THE BUS CYCLE (IE, EMIT
17082 ! IS TEMPORARILY DISABLED)
17083 4471:
17084 TEST730A:
17085     PO,      LOAD-ENUA(ZTARGET402),      ! SETUP FOR D=ZERO COMPARE
17086     P1,      LOAD-ERROR(TEST730A),      ! ERROR DIRECTORY KEY
17087     P2-T,    DCS-CTR(C9.),              ! COMPARE AT TARGET
17088     NEXT,    J/LOADIR730A
(4471) DCS(1.00.1.0.0.0) BM(0110..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...111.101.100)

17089
17090 4754:
17091 LOADIR730A:
17092     P2-U,    IR+EMIT, EMIT/125200,      ! "CMP-BYTE" INSTR; INSTR5-E88(412) DECODE
17093     NEXT,    J/LOADDATA730A
(4754) DCS(0.00.0.0.0.0) BM(1010..00.10..10.10..000..000...0.0.0..0..0...1.1010...0..0000.0...11.000...011.011.000)

17094
17095 4330: !(FREE)
17096 LOADDATA730A:
17097     P3,      CSPD(16)+EMIT, EMIT/125252, ! PATTERN (125252) IN BASCON AREA
17098     NEXT,    J/BUSFCN730A
(4330) DCS(0.00.0.0.0.0) BM(1010..10.10..10.10..101..010...0.0.0..0..0...0.0001...1..0000.0...11.000...011.011.001)

17099
17100 4331: !(FREE)
17101 BUSFCN730A:
17102     PO,      BUMP-VERIFY,              ! COUNT
17103     P1,      BA+ASPHI(C000000),        ! USE MEMORY ADDR(000000)
17104     P2-T,    D+CSPB(B16), D(C)+0,      ! USE DATA (125252)
17105     P3,      DATO,                      ! FOR A BUS "DATO" CYCLE
17106     NEXT,    J/GETDBUF730A
(4331) DCS(0.00.0.0.0.1) BM(1010..11.01..11.01..100..000...0.1.0..1..0...1.0010...0..0000.0...11.000...011.011.010)

17107
17108 4332: !(FREE)
17109 GETDBUF730A:
17110     PO,      BUMP-VERIFY,              ! COUNT
17111     P3,      CSPD(17)+BUSDIN, EMIT/037777, ! DBUF SHOULD BE ENABLED; EMIT IS NOISE
17112     P3-T,    D+JUNK, SAVE-D(C),        ! MANGLE DATA IN D, DONT CARE WHAT RESULTS
17113     NEXT,    J/COMP730A
(4332) DCS(0.00.0.0.0.1) BM(0011..10.11..11.11..111..111...1.1.0..0..0...0.0000...1..0000.0...11.000...011.011.011)

17114
17115 4333: !(FREE)
17116 COMP730A:
17117     PO,      BUMP-VERIFY,              ! COUNT
17118     P2-T,    D+CSPD(17)-XOR-ASPHI(C125252), ! COMPARE RECEIVED:EXPECTED

```

```

17119      NEXT      J/ZAPDBUF730A
(4333) DCS(0.00.0.0.0.0.1) BM(0110..10.00..11.01..110..000...0.1.0..0..0...0 0000...0..0000.0...11.000...011.011.100)
17120
17121      4334:  !(FREE)
17122      ZAPDBUF730A:
17123      P3,      DBUF←D-[I],      !COPY ZEROED(?) D-REG INTO DBUF
17124      NEXT      J/GOBUT730A
(4334) DCS(0.00.0.0.0.0.0) BM(0100..00.00..00.00..000..100...0.0.0..0..0...1.1011...0..0000.0...11.000...011.011.101)
17125
17126      4335:  !(FREE)
17127      GOBUT730A:
17128      SETUP,   RETURN/TEST730B,      !RETURN TO START OF NEXT SUBTEST
17129      NEXT,    GOTO-PAGE(7),          !BUT TABLE
17130      J/BUTD-IS-ZERO                    !CHECK FOR EQUALITY
(4335) DCS(0.00.0.0.0.0.0) BM(0100..00.10..01.10..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)
17131
17132
17133
17134
17135
17136
17137
17138      ! - - - - -
17139
17140      !TEST-730-B DOES A DATIP, AND THEN CHECKS THAT:
17141      ! 1) NO JOB ADDRESS ERROR RESULTS
17142      ! 2) THE RIGHT DATA (OUTPUT ABOVE) IS RETRIEVED (NOTE DBUF LATCH WAS ZEROED
17143      !    TO ALTER ITS COPY OF THE DATA)
17144      ! 3) THE BUS HOLDING FUNCTION OF THE 'DATIP' SHOULD BE EMPLOYED; 'BBSY' SHOULD REMAIN ASSERTED
17145      !    WELL PAST THE 'NORMAL' MICRWARD AFTER THE BUS CYCLE. IN FACT, IT SHOULD REMAIN ASSERTED
17146      !    (HOLDING BUSDIN=UNLESS DATA-BUFFER, NOT EMIT, ETC) UNTIL CLEARED BY ANOTHER BUS CYCLE
17147      !    (NOT A DATIP), OR A BUTA(LAST) [DONE HERE].
17148
17149      4466:
17150      TEST730B:
17151      PO,      LOAD-EMIT(ZTARGET402),      !SETUP FOR D=ZERO COMPARE
17152      LOAD-ERROR(TEST730B),              !ERROR DIRECTORY KEY
17153      DCS-CTR(C7.),                      !COMPARE AT TARGET
17154      NEXT      J/EXPEC730B
(4466) DCS(1.00.1.0.0.0.0) BM(1000..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.011.110)
17155
17156      4336:  !(FREE)
17157      EXPEC730B:
17158      P3,      CSPD(02)←EMIT, EMIT/000340,      !SERVICE PORT BITS FOR A "DATIP"
17159      NEXT      J/BUSFCN730B
(4336) DCS(0.00.0.0.0.0.0) BM(0000..10.00..00.11..100..000...0.0.0..0..0...0.1101...1..0000.0...11.000...011.011.111)
17160
17161      4337:  !(FREE)
17162      BUSFCN730B:
17163      P1,      BA←ASPHI(000001),          !USE MEMORY ADDR(000001), 000 BYTE
17164      P2-T,    D←ZERO, D(C)←ALU15,      !ZAP D
17165      P3,      DATIP,                    !FOR A BUS "DATIP" CYCLE
17166      BUTA(CLR-FLAG-RES-UCON),          !RESET BUSDIN TO EMIT/ZAP DBUF←D UCON
17167      NEXT      J/GETIT730B

```

```

(4337) DCS(0.00.0.0.0.0) BM(0011..00.00..11.01..000..100...0.1.0..1..0...1.0100...0..0000.0...11.010...011.100.000)
17167
17168 4340: !(FREE)
17169 GETIT730B:
17170 PO, BUMP-VERIFY, !COUNT
17171 P3, CSPD(17)-BUSDIN, EMIT/052525, !UNIBUS DATA SHOULD BE ENABLED; EMIT IS NOISE
17172 NEXT, J/GOBUT730B
(4340) DCS(0.00.0.0.0.1) BM(0101..10.01..01.01..010..101...0.0.0..0..0...0.0000...1..0000.0...11.000...011.100.001)
17173
17174 4341: !(FREE)
17175 GOBUT730B:
17176 SETUP, RETURN/TEST730C, !EXEC SUBR WHICH:
17177 NEXT, CALL(CSP17XOR125252) ! CSP(17)-XOR-(125252) -> 0, BUT(D=ZERO)
(4341) DCS(0.00.0.0.0.0) BM(0100..00.10..01.10..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.000.011)
17178
17179
17180
17181
17182
17183 ! - - - - -
17184
17185 !TEST-730-C NOW CHECKS THAT THE 'DATIP' FUNCTION IS STILL HOLDING THE BUS BY
17186 ! VERIFYING THAT THE UNIBUS DATA BUFFER IS STILL ENABLED ON BUSDIN, DATA=(125252)
17187 4460:
17188 TEST730C:
17189 PO, LOAD-ENUA(ZTARGET402), !SETUP FOR D=ZERO COMPARE
17190 LOAD-ERROR(TEST730C), !ERROR DIRECTORY KEY
17191 DCS-CTR(CS.), !COMPARE AT TARGET
17192 BUMP-VERIFY, !COUNT
17193 NEXT, J/GOBUT730C
(4460) DCS(1.00.1.0.0.1) BM(1010..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.100.010)
17194
17195 4342: !(FREE)
17196 GOBUT730C:
17197 SETUP, RETURN/TEST730C1, !EXEC SUBR WHICH:
17198 NEXT, CALL(BUSDINXOR125252) ! 1) BUSDIN -> CSP(17), EMIT=(052525)
17199 ! 2) CSP(17)-XOR-(125252) -> 0, BUT(D=ZERO)
(4342) DCS(0.00.0.0.0.0) BM(0100..00.10..01.01..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...001.111.111)
17200
17201
17202
17203
17204
17205 ! - - - - -
17206
17207 !TEST-730-C1 NOW CHECKS THAT THE 'DATIP' FUNCTION WILL RELEASE THE BUS BY
17208 ! VERIFYING THAT THE UNIBUS DATA BUFFER IS NOT ENABLED ON BUSDIN, DATA=(125252),
17209 ! AFTER ISSUING BUTA(LAST), WHICH SHOULD CLEAR THE DATIP/BBSY FLOP
17210 4450:
17211 TEST730C1:
17212 PO, LOAD-ENUA(ZTARGET402), !SETUP FOR D=ZERO COMPARE
17213 LOAD-ERROR(TEST730C1), !ERROR DIRECTORY KEY

```

```

17214          DCS-CTR(C5.),          !COMPARE AT TARGET
17215          P3, BUTA(LAST),          !ACTIVE BUT EFFECT: CLEAR OUT DATIP
17216          NEXT, J/GOBUT730C1
(4450) DCS(1.00.1.0.0.0) BM(1010..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...10.000...100.100.011)

17217
17218          4443:
17219          GOBUT730C1:
17220          SETUP, RETURN/TEST7300,  !EXEC SUBR WHICH:
17221          NEXT, CALL(BUSDINXOR052525) ! 1) BUSDIN -> CSP(17), EMIT=(052525)
17222          ! 2) CSP(17)-XOR-(052525) -> D, BUT(D=ZERO)
(4443) DCS(0.00.0.0.0.0) BM(0100..00.10..01.10..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.000.100)

17223
17224
17225
17226
17227
17228
17229          !-----
17230          !TEST-730-D DOES A DATO FOLLOWED BY A DATIB*BYTE*000, AND THEN CHECKS THAT THE RIGHT DATA
17231          ! IS WRITTEN/RETRIEVED FROM MEMORY LOCATIONS (000000)/(000001)
17232          ! AND THAT THE EMIT-DISABLE/UNIBUS DATA BUFFER ENABLE IS HANDLED CORRECTLY
17233          4461:
17234          TEST7300:
17235          PO, LOAD-EMUA(ZTARGET402), !SETUP FOR D=ZERO COMPARE
17236          LOAD-ERROR(TEST7300), !ERROR DIRECTORY KEY
17237          DCS-CTR(C15.), !HOLD UP FOR NOW
17238          NEXT, J/LOADDATA7300
(4461) DCS(1.00.1.0.0.0) BM(0000..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.100.011)

17239          4343: !(FREE)
17240          LOADDATA7300:
17241          P3, CSPD(16)+EMIT, EMIT/052525, !PATTERN (052525) IN BASCON AREA
17242          NEXT, J/SETZER7300
(4343) DCS(0.00.0.0.0.0) BM(0101..10.01..01.01..010..101...0.0.0..0..0...0.0001...1..0000.0...11.000...011.100.100)

17243          4344: !(FREE)
17244          SETZER7300:
17245          P1, BA+ASPHI(C000000), !USE MEMORY ADDR(000000)
17246          P2-T, D+CSPB(B16), D(C)+0, !USE DATA (052525)
17247          P3, DATO, !FOR A BUS "DATO" CYCLE
17248          NEXT, J/MANGLED7300
(4344) DCS(0.00.0.0.0.0) BM(1010..11.01..11.01..100..000...0.1.0..1..0...1.0010...0..0000.0...11.000...011.100.101)

17250          4345: !(FREE)
17251          MANGLED7300:
17252          P3-T, D+ASPHI(C125252), SAVE-D(C), !MANGLE D CONTENTS, AFTER/AT P3-T
17253          NEXT, J/BUSFCN7300
(4345) DCS(0.00.0.0.0.0) BM(1111..00.00..11.01..110..111...1.1.0..0..0...0.0000...0..0000.0...11.000...011.100.110)

17255          4346: !(FREE)
17256          BUSFCN7300:
17257          PO, BUMP-VERIFY, !COUNT
17258          DCS-CTR(C12.), !COMPARE AT TARGET
17259          P1, BA+ASPHI(C000001), !ADDRESS 000 BYTE
17260

```

17261 P3, DATIB, !BYTE READ -> PLAIN DATI; NO ODD ADDRESS ERROR  
 17262 NEXT, J/GETIT7300  
 (4346) DCS(0.00.1.0.0.0) BM(0011..00.00..11.01..000..000...0.0.0..1..0...1.0011...0..0000.0...11.000...011.100.111)

17263 4347: !(FREE)  
 17264 GETIT7300:  
 17265 P3, CSPC(17)+BUSDIN, EMIT/00000), !UNIBUS DATA SHOULD BE ENABLED; EMIT IS NOISE  
 17266 NEXT, J/GOBUT7300  
 (4347) DCS(0.00.0.0.0.0) BM(0000..10.00..00.00..000..000...0.0.0..0..0...0.0000...1..0000.0...11.000...011.101.000)

17268 4350: !(FREE)  
 17269 GOBUT7300:  
 17270 SETUP, RETURN/TEST730E, !GO TO SUBR WHICH:  
 17271 NEXT, CALL[CLRSERVICE100] ! (SERVICE)-XOR-(000340) -> D, BUT(D=ZERO)  
 17272 (4350) DCS(0.00.0.0.0.0) BM(0100..00.10..01.10..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.100)

17273  
 17274  
 17275  
 17276  
 17277  
 17278  
 17279  
 17280 ! - - - - -  
 17281  
 17282

17283 !TEST-730-E NOW CHECKS THAT THE BUS FUNCTION ABOVE ACTUALLY RETRIEVED THE RIGHT DATA:  
 17284 ! THE (052525) WRITTEN TO MEMORY LOCATION (00000) IN TEST730E

17285 4462:  
 17286 TEST730E:  
 17287 PD, LOAD-ENUA(ZTARGET402), !SETUP FOR D=ZERO COMPARE  
 17288 LOAD-ERROR(TEST730E), !ERROR DIRECTORY KEY  
 17289 DCS-CTR(C4.), !COMPARE AT TARGET  
 17290 NEXT, J/GOBUT730E  
 (4462) DCS(1.00.1.0.0.0) BM(1011..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.101.001)

17291 4351: !(FREE)  
 17292 GOBUT730E:  
 17293 SETUP, RETURN/SCOPE730, !EXEC SUBR WHICH:  
 17294 NEXT, CALL[CSP17XOR052525] ! CSP(17)-XOR-(052525) -> D, BUT(D=ZERO)  
 17295 (4351) DCS(0.00.0.0.0.0) BM(0100..00.01..11.01..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.000.101)

17296 4352: !(FREE)  
 17297 SCOPE730:  
 17298 NEXT, BUTD(SCOPE), !NO ERROR: "TEST731A" (+1. WORDS)  
 17300 J/TEST731A ! ERROR: "LOADIR730A" (-24. WORDS)  
 17301 (4352) DCS(0.00.0.1.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...111.101.101)

17302  
 17303  
 17304  
 17305  
 17306

```

17307
17308
17309
17310 !-----
17311
17312 !THIS SECOND SERIES OF TESTS DOES A DATOB/DATI/DATI-CLKIR SEQUENCE, CHECKING THAT
17313 !EACH FUNCTION OPERATES AS EXPECTED.
17314 !-----
17315
17316
17317 !TEST-731-A DOES A DATOB*BYTE*000, AND THEN CHECKS THAT THE DBUF LATCH (DS) ALSO GETS LOADED WITH THE
17318 ! DATA, AND THAT IT IS ENABLED ON BUSDIN IN THE MICROWORD AFTER THE BUS CYCLE (IE, EMIT
17319 ! IS TEMPORARILY DISABLED)
17320 4755:
17321 TEST731A:
17322     PO,      LOAD-ENUA(ZTARGET402),      !SETUP FOR D=ZERO COMPARE
17323             LOAD-ERROR(TEST731A),      !ERROR DIRECTORY KEY
17324             DCS-CTR(C6.),              !COMPARE AT TARGET
17325     NEXT,    J/BUSFCN731A
(4755) DCS(1.00.1.0.0.0) BM(1001..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...111.110.100)

17326
17327 4764:
17328 BUSFCN731A:
17329     P1,      BA+ASPHI(C000001),          !USE MEMORY ADDR(000001), 000 BYTE
17330     P2-T,    D+ZERO, D(C)+ALU15,        !USE DATA (000), ONE BYTE (000) ONLY
17331     P3,      DATOB,                      !FOR A BUS "DATOB*BYTE*000" CYCLE
17332     NEXT,    J/GETDBUF731A
(4764) DCS(0.00.0.0.0.0) BM(0011..00.00..11.01..000..100...0.1.0..1..0...1.0101...0..0000.0...11.000...011.101.011)

17333
17334 4353: !(FREE)
17335 GETDBUF731A:
17336     PO,      BUMP-VERIFY,                !COUNT
17337     P3,      CSPD(17)+BUSDIN,            !DBUF SHOULD BE ENABLED; EMIT IS NOISE
17338     P3-T,    D+ASPHI(C177777), SAVE-D(C), !MANGLE DATA IN D, DONT CARE WHAT RESULTS
17339     NEXT,    J/COMP731A
(4353) DCS(0.00.0.0.0.1) BM(1111..10.00..11.01..101...111...1.1.0..0..0...0.0000...1..0000.0...11.000...011.101.100)

17340
17341 4354: !(FREE)
17342 COMP731A:
17343     P2-T,    D+CSPD(D17), SAVE-D(C),     !COMPARE RECEIVED:(000000)
17344     NEXT,    J/GOBUT731A
(4354) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000...111...0.1.0..0..0...0.0000...0..0000.0...11.000...011.101.101)

17345
17346 4355: !(FREE)
17347 GOBUT731A:
17348     SETUP,   RETURN/TEST731B,           !RETURN TO START OF NEXT SUBTEST
17349     NEXT,    GOTO-PAGE(7),              !BUT TABLE
17350     J/BUTD-IS-ZERO                       !CHECK FOR EQUALITY
(4355) DCS(0.00.0.0.0.0) BM(0100..00.10..01.01..111...111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001)

17351
17352
17353
17354

```



```

17355
17356
17357
17358 ! - - - - -
17359
17360 !TEST-731-B NOW CHECKS THAT NONE OF THE ABOVE BUS FUNCTIONS HAS ALTERED THE IR FROM WHEN
17361 ! IT WAS LOADED IN TEST730A, WITH THE VALUE (125200), INSTR5-EB8(412) DECODE
17362 4457:
17363 TEST731B:
17364     PO,          LOAD-ENUA(ZTARGET412),          !SETUP FOR INSTR5/EB8 DECODE
17365                LOAD-ERROR(TEST731B),          !ERROR DIRECTORY KEY
17366                DCS-CTR(CS.),                  !COMPARE AT TARGET
17367                NEXT, J/BUSFCN731B
(4457) DCS(1.00.1.0.0.0) BM(1010..00.11..11.00..001..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.101.110)

17368
17369 4356: !(FREE)
17370 BUSFCN731B:
17371     P1,          BA+ASPHI(C000000),          !USE MEMORY ADDR(U00000)
17372     P3,          DATI                        !FOR A BUS "DATI" CYCLE
17373     NEXT, J/GETIT731B
(4356) DCS(0.00.0.0.0.0) BM(0000..00.00..11.01..100..000...0.0.0..1..0...1.0110...0..0000.0...11.000...011.101.111)

17374
17375 4357: !(FREE)
17376 GETIT731B:
17377     PO,          BUMP-VERIFY,                !COUNT
17378     P3,          CSPD(17)+BUSDIN, EMIT/152500, !UNIBUS DATA SHOULD BE ENABLED; EMIT IS NOISE
17379     NEXT, J/GOBUT731B                       ! THIS DATA IS INSTR5-EB8(405) DECODE
(4357) DCS(0.00.0.0.0.1) BM(1101..10.01..01.01..000..000...0.0.0..0..0...0.0000...1..0000.0...11.000...011.110.000)

17380
17381 4360: !(FREE)
17382 GOBUT731B:
17383     SETUP,      RETURN/TEST731C,           !RETURN TO START OF NEXT SUBTEST
17384     NEXT,       GOTO-PAGE(7),             !BUT TABLE
17385     J/BUTINSTRS, !CHECK THAT IR STILL HAS DATA (125200)
(4360) DCS(0.00.0.0.0.0) BM(0100..00.10..01.01..110..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.001)

17386
17387
17388
17389
17390
17391
17392
17393 ! - - - - -
17394
17395
17396 !TEST-731-C NOW CHECKS THAT THE BUS FUNCTION ABOVE ACTUALLY RETRIEVED THE RIGHT DATA:
17397 ! THE (052525) WRITTEN TO MEMORY LOCATION (001)/(000) IN TEST730D
17398 ! AND THE (000) WRITTEN TO MEMORY LOCATION (001) IN TEST731A
17399 ! TOGETHER THESE FORM A (000125) IN MEMORY LOCATION (001)*(000): INSTR5-E78(432) DECODE
17400 4456:
17401 TEST731C:
17402     PO,          LOAD-ENUA(ZTARGET432),          !SETUP FOR INSTR5-E78(432) DECODE
17403                LOAD-ERROR(TEST731C),          !ERROR DIRECTORY KEY

```

```

17404          DCS-CTR(C7.),          !COMPARE AT TARGET
17405          BUMP-VERIFY,          !COUNT
17406          NEXT J/COMP731C
(4456) DCS(1.00.1.0.0.1) BM(1000..00.11..11.00..011..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.110.001)

17407          4361: !(FREE)
17408          COMP731C:
17409          P2-T, D+CSPD(D17), D(C1)+0,          !GET DATA READ FROM DAT1, ABOVE
17410          NEXT J/GOBUT731C
(4361) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...0.1.0..0..0...0.0000...0..0000.0...11.000...011.110.010)

17412          4362: !(FREE)
17413          GOBUT731C:
17414          SETUP, RETURN/TEST731D,          !RETURN TO START OF NEXT SUBTEST
17415          NEXT CALL(DINTOIR-5)          !SUBR FOR: D -> IR, BUT(INSTR5)
(4362) DCS(0.00.0.0.0.0) BM(0100..00.10..01.01..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.111.011)

17417
17418
17419
17420
17421
17422          ! - - - - -
17423
17424          !TEST-731-D DOES A DAT1-CLKIR, AND THEN CHECKS THAT THE RIGHT DATA
17425          ! IS WRITTEN/RETRIEVED FROM MEMORY LOCATIONS (000000)/(000001)
17426          ! AND THAT THE EMIT-DISABLE/UNIBUS DATA BUFFER ENABLE IS HANDLED CORRECTLY
17427          4455:
17428          TEST731D:
17429          PO, LOAD-ENVA(ZTARGET402),          !SETUP FOR D=ZERO COMPARE
17430          LOAD-ERROR(TEST731D),          !ERROR DIRECTORY KEY
17431          DCS-CTR(C14.),          !COMPARE AT TARGET
17432          NEXT J/LOADIR731D
(4455) DCS(1.00.1.0.0.0) BM(0001..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.110.011)

17433          4363: !(FREE)
17434          LOADIR731D:
17435          P2-U, IR+EMIT, EMIT/125200,          !PREV DATA IN IR FOR INSTR5-E88(412)
17436          NEXT J/BUSFCN731D
(4363) DCS(0.00.0.0.0.0) BM(1010..00.10..10.10..000..000...0.0.0..0..0...1.1010...0..0000.0...11.000...011.110.100)

17438          4364: !(FREE)
17439          BUSFCN731D:
17440          P1, BA+ASPHI(C000000),          !FROM LOCATION (000000)
17441          P3, DAT1-CLKIR,          !DO A DAT1, AND CLKIR
17442          NEXT J/MANGLED731D
(4364) DCS(0.00.0.0.0.0) BM(0000..00.00..11.01..100..000...0.0.0..1..0...1.0000...0..0000.0...11.000...011.110.101)

17444          4365: !(FREE)
17445          MANGLED731D:
17446          P2-U, !IR+DATA,          !IR SHOULD GET DATA HERE
17447          EMIT/152500,          !NOISE ON EMIT: INSTR5-E88(405)
17448          NEXT J/GOBUT731D
(4365) DCS(0.00.0.0.0.0) BM(1101..00.01..01.01..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...011.110.110)

```

```

17450
17451 4366: !(FREE)
17452 GOBUT731D:
17453     SETUP, RETURN/TEST731E,           !GO TO SUBR WHICH:
17454     NEXT,  CALL(CLRSERVICE)OD)      ! (SERVICE)-XOR-(000340) -> 0 BUT(D=ZERO)
(4366) DCS(0.00.0.0.0.0) BM(0100..00.10..01.01..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.100)

```

```

17455
17456
17457
17458
17459
17460
17461
17462 ! - - - - -
17463
17464

```

```

17465 !TEST-731-E NOW CHECKS THAT THE BUS FUNCTION ABOVE ACTUALLY RETRIEVED THE RIGHT DATA:
17466 4454:
17467 TEST731E:
17468     PD,      LOAD-ENVA(ZTARGET432),      !SETUP FOR INSTR5-E78(432) DECODE
17469           LOAD-ERROR(TEST731E),        !ERROR DIRECTORY KEY
17470           DCS-CTR(C3.),                 !COMPARE AT TARGET
17471     NEXT,    J/GOBUT731E
(4454) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..011..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.110.111)

```

```

17472
17473 4367: !(FREE)
17474 GOBUT731E:
17475     SETUP, RETURN/SCOPE731,           !RETURN TO SCOPE LOOP TEST WORD
17476     NEXT,  GOTO-PAGE(7),             !BUT TABLE
17477           J/BUTINSTRS                 !CHECK FOR RIGHT DATA
(4367) DCS(0.00.0.0.0.0) BM(0100..00.01..11.11..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.001)

```

```

17478
17479
17480 4370: !(FREE)
17481 SCOPE731:
17482     P2-U,   IR+EMIT, EMIT/125200,      !RESET IR FOR SCOPE LOOP
17483     NEXT,   BUTD(SCOPE)                 !NO ERROR: "LOADIR740A" (+5. WORDS)
17484           J/LOADIR740A                 ! ERROR: "BUSFCN731A" (-18. WORDS)
(4370) DCS(0.00.0.1.0.0) BM(1010..00.10..10.10..000..000...0.0.0..0..0...1.1010...0..0000.0...11.000...111.110.101)

```

```

17485
17486 ! - - - - -
17487
17488 ! THESE SUBROUTINES ARE USED IN THE PREVIOUS TESTS:
17489
17490

```

```

17491 7177: !(FREE)
17492 BOX12:
17493     P3,     CSPD(17)+BUSDIN, EMIT/052525, !ENTRY FOR "BUSDINXOR125252"
17494     NEXT,   J/C17X12
(7177) DCS(0.00.0.0.0.0) BM(0101..10.01..01.01..010..101...0.0.0..0..0...0.0000...1..0000.0...11.000...010.000.011)

```

```

17495
17496 7203: !(FREE)

```

```

17497 C17X12:
17498 P2-T, D+CSPD(17)-XOR-ASPHI(C125252), !ENTRY FOR "CSP17XOR125252"
17499 SAVE-D(C),
17500 NEXT J/BUTD-IS-ZERO !FINISH OUT BY TESTING RESULT OF COMPARISON
(7203) DCS(0.00.0.0.0.0) BM(0110..10.00..11.01..110..111...0.1.0..0...0.0000...0..0000.0...11.000...011.100.001)

17501
17502
17503 7204: !(FREE)
17504 BOX05:
17505 P3, CSPD(17)+BUSDIN, EMIT/052525, !ENTRY FOR "BUSDINXOR052525"
17506 NEXT J/C17X05
(7204) DCS(0.00.0.0.0.0) BM(0101..10.01..01.01..010..101...0.0.0..0..0...0.0000...1..0000.0...11.000...010.000.101)

17507
17508 7205: !(FREE)
17509 C17X05:
17510 P2-T, D+CSPD(17)-XOR-ASPHI(C052525), !ENTRY FOR "CSP17XOR052525"
17511 SAVE-D(C),
17512 NEXT J/BUTD-IS-ZERO !FINISH OUT BY TESTING RESULT OF COMPARISON
(7205) DCS(0.00.0.0.0.0) BM(0110..10.00..11.01..111..111...0.1.0..0..0...0.0000...0..0000.0...11.000...011.100.001)

```

```

17513 ! - - - - -
17514 !
17515 !
17516 !
17517 !
17518 !
17519 !

```

! .PAGE=====

```

17521
17522
17523 .TOC * TEST740: BUS CYCLE MODIFICATION - PREFETCH ALTERATION, OVERLAP YANK
17524

```

```

17525 !*****
17526 !*
17527 !* TESTS: 740A - 740D UWORDS: 020 + 043
17528 !*
17529 !* FUNCTIONS:
17530 !*
17531 !* THESE FOUR TESTS CHECK THAT LOGIC CONCERNED WITH:
17532 !*
17533 !* 1) BCD/PREFETCH = BCD(1) * NOT(PREFETCH#BUT(INSTR1))
17534 !* IE, ALTER BUS CODE (**1) -> (**0), IN A PREFETCH SITUATION
17535 !*
17536 !* AND 2) BEGIN-DATA-XFER = BEGIN(BUS/UCON) * NOT(PULSE-SUPPRESS(0)) * NOT(INH-SPL0)
17537 !* IE, 'YANK' (STOP) A BUS CYCLE IN A NON-OVERLAP SITUATION
17538 !*
17539 !*****

```

```

17540
17541
17542
17543
17544
17545 ! - - - - -
17546 !
17547 !

```

```

17548 !TEST-740-A CHECKS THAT BC(0)-H DOES NOT GET ALTERED FROM (1) -> (0) WHEN:
17549 ! BUTA(INSTR1)-H IS ASSERTED, BUT PREFETCH-H IS NEGATED
17550 4765:
17551 LOADIR740A:
17552 P2-U, IR+EMIT, EMIT/056000, ! (OVERLAP,-PREFETCH), (-BYTE,DOP,-SMO)
17553 NEXT J/TEST740A
(4765) DCS(0.00.0.0.0.0) BM(0101..00.11..00.00..000..000...0.0.0..0..0...1.1010...0..0000.0...11.000...111.100.100)

17554 4744:
17555 TEST740A:
17556 PO, LOAD-ENUA(ZTARGET402), ! SETUP FOR D=ZERO COMPARE
17557 LOAD-ERROR(TEST740A), ! ERROR DIRECTORY KEY
17558 DCS-CTR(C12.), ! COMPARE AT TARGET
17559 NEXT J/BUSFCN740A
17560 (4744) DCS(1.00.1.0.0.0) BM(0011..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.111.001)

17561 4371: !(FREE)
17562 BUSFCN740A:
17563 PO, BUMP-VERIFY, ! COUNT
17564 P1, BA+ASPHI(C000000), ! USE MEMORY ADDR (000000)
17565 P2-T, SR+ZERO, ! SR(00)=0, FOR UNEXPECTED JAMUPPS
17566 P3, DATIB, ! CODE=(3)/DATIB, POSSIBLY ALTERED TO (2)/DATO
17567 BUTA(INSTR-1), ! ACTIVE EFFECT ONLY - BRANCH MASKED
17568 NEXT J/GOBUT740A ! (056000) TARGETS TO (316), MASKED UNDER (776)
17569 (4371) DCS(0.00.0.0.0.1) BM(0011..00.00..11.01..100..000...0.0.1..1..0...1.0011...0..0000.0...00.110...111.111.110)

17570 4776:
17571 GOBUT740A:
17572 SETUP, RETURN/LOADIR740B, ! GO TO SUBR WHICH:
17573 ! CLR-JAM-ERRORS FOR INSURANCE
17574 NEXT CALL(DATISERVICET00) ! (SERVICE)-XOR-(100340) -> 0, BUT(D=ZERO)
17575 (4776) DCS(0.00.0.0.0.0) BM(0100..00.01..11.11..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.101)

17576
17577
17578
17579
17580
17581 ! - - - - -
17582
17583 !TEST-740-B CHECKS THAT BC(0)-H DOES GET ALTERED FROM (1) -> (0) WHEN:
17584 ! BUTA(INSTR1)-H IS ASSERTED, AND PREFETCH-H IS ASSERTED
17585 4372: !(FREE)
17586 LOADIR740B:
17587 P2-U, IR+EMIT, EMIT/020606, ! (OVERLAP,PREFETCH), (-BYTE,DOP,SMO)
17588 NEXT J/TEST740B
(4372) DCS(0.00.0.0.0.0) BM(0010..00.00..01.10..000..110...0.0.0..0..0...1.1010...0..0000.0...11.000...100.101.011)

17589 4453:
17590 TEST740B:
17591 PO, LOAD-ENUA(ZTARGET402), ! SETUP FOR D=ZERO COMPARE
17592 LOAD-ERROR(TEST740B), ! ERROR DIRECTORY KEY
17593 DCS-CTR(C12.), ! COMPARE AT TARGET
17594 NEXT, J/BUSFCN740B
17595

```

```

(4453) DCS(1.00.1.0.0.0) BM(0011..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.111.011)
17596
17597 4373: !(FREE)
17598 BUSFCN740B:
17599 P1, BA+ASPHI(C000000), !USE MEMORY ADDR (000000)
17600 P3, DATIB, !CODE=(3)/DATIB, POSSIBLY ALTERED TO (2)/DATO
17601 NEXT BUTA(INSTR-1), !ACTIVE EFFECT ONLY - BRANCH MASKED
17602 J/G0BUT740B ! (020606) TARGETS TO (042), MASKED UNDER (766)
(4373) DCS(0.00.0.0.0.0) BM(0000..00.00..11.01..100..000...0.0.0..1..0...1.0011...0..0000.0...00.110...111.110.110)
17603
17604 4766:
17605 G0BUT740B:
17606 SETUP, RETURN/TEST740C, !GO TO SUBR WHICH:
17607 ! CLR-JAM-ERRORS, FOR INSURANCE
17608 NEXT CALL(DATOSERVICET00) ! (SERVICE)-XOR-(002340) -> 0, BUT(D=ZERO)
(4766) DCS(0.00.0.0.0.0) BM(0100..00.10..01.01..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.110)
17609
17610
17611
17612
17613
17614 ! - - - - -
17615
17616 !TEST-740-C CHECKS THAT BC(0)-H DOES NOT GET ALTERED FROM (1) -> (0) WHEN:
17617 ! BUTA(INSTR1)-H IS NEGATED, BUT PREFETCH-H IS ASSERTED
17618 4452:
17619 TEST740C:
17620 PO, LOAD-ENJA(ZTARGET402), !SETUP FOR D=ZERO COMPARE
17621 LOAD-ERROR(TEST740C), !ERROR DIRECTORY KEY
17622 DCS-CTR(C12.), !COMPARE AT TARGET
17623 BUMP-VERIFY, !COUNT
17624 NEXT J/BUSFCN740C !IR AS ABOVE: (OVERLAP,PREFETCH), (-BYTE,DOP,SMD)
(4452) DCS(1.00.1.0.0.1) BM(0011..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.111.100)
17625
17626 4374: !(FREE)
17627 BUSFCN740C:
17628 P1, BA+ASPHI(C000000), !USE MEMORY ADDR (000000)
17629 P3, DATIB, !CODE=(3)/DATIB, POSSIBLY ALTERED TO (2)/DATO
17630 NEXT J/G0BUT740C ! (020606) TARGETS TO (042), MASKED UNDER (766)
(4374) DCS(0.00.0.0.0.0) BM(0000..00.00..11.01..100..000...0.0.0..1..0...1.0011...0..0000.0...11.000...011.111.101)
17631
17632 4375: !(FREE)
17633 G0BUT740C:
17634 SETUP, RETURN/LOADIR7400, !GO TO SUBR WHICH:
17635 ! CLR-JAM-ERRORS, FOR INSURANCE
17636 NEXT CALL(DATISERVICET00) ! (SERVICE)-XOR-(100340) -> 0, BUT(D=ZERO)
(4375) DCS(0.00.0.0.0.0) BM(0100..00.10..00.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.101)
17637
17638
17639
17640
17641

```

```

17642 ! - - - - -
17643
17644 ! TEST-740-D CHECKS THAT WHEN BUTA(INSTR1)-H IS ASSERTED, AND OVERLAP-L IS
17645 ! NEGATED, THEN THE BUS CYCLE IS NOT EVEN ALLOWED TO BEGIN
17646 4400: !(FREE)
17647 LOADIR7400:
17648 PO, BUMP-VERIFY !COUNT
17649 P2-U, IR+EMIT, EMIT/076000, !(-OVERLAP,-PREFETCH), (-BYTE,-DOP,-SOP)
17650 NEXT, J/TEST7400
(4400) DCS(0.00.0.0.0.0) BM(0111..00.11..00.00..000..000...0.0.0..0..0...1.1010...0..0000.0...11.000...100.101.001)

17651
17652 4451:
17653 TEST7400:
17654 PO, LOAD-ENUA(ZTARGET402), !SETUP FOR D=ZERO COMPARE
17655 LOAD-ERROR(TEST7400), !ERROR DIRECTORY KEY
17656 DCS-CTR(C11.), !COMPARE AT TARGET
17657 NEXT, J/BUSFCN7400
(4451) DCS(1.00.1.0.0.0) BM(0100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.000.001)

17658
17659 4401: !(FREE)
17660 BUSFCN7400:
17661 P1, BA+ASPHI(C000001), !USE MEMORY ADDR (000001), TRY TO FORCE ODD ADDR
17662 P3, DATO !TRY TO ALTER SERVICE FROM (100340)/DATI TO (002340)/DATO
17663 BUTA(INSTR-1), !ACTIVE EFFECT ONLY - BRANCH MASKED
17664 NEXT, J/GOBUT7400 ! (076000) TARGETS TO (047), MASKED UNDER (767)
(4401) DCS(0.00.0.0.0.0) BM(0000..00.00..11.01..000..000...0.0.0..1..0...1.0010...0..0000.0...00.110...111.110.111)

17665
17666 4767:
17667 GOBUT7400:
17668 SETUP, RETURN/SCOPE740, !GO TO SUBR WHICH:
17669 NEXT, CALL(CJESERVICETOD) ! [CSP(02) LOADED IN LAST TEST; SAME VALUE= (100340)]
17670 ! (SERVICE)-XOR-CSP(02) -> D, BUT(D=ZERO)
(4767) DCS(0.00.0.0.0.0) BM(0100..00.10..00.00..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.100.111)

17671
17672
17673
17674 4402: !(FREE)
17675 SCOPE740:
17676 P2-U, IR+EMIT, EMIT/056000, !RELOAD IR FOR TEST740A
17677 NEXT, BUTD(SCOPE), !NO ERROR: "TEST761A" (+1. WORDS)
17678 J/TEST761A ! ERROR: "LOADIR740A" (-16. WORDS)
(4402) DCS(0.00.0.1.0.0) BM(0101..00.11..00.00..000..000...0.0.0..0..0...1.1010...0..0000.0...11.000...111.100.101)

17679
17680
17681
17682
17683
17684 !.PAGE=====
17685
17686
17687 .TOC * TEST761-763: TESTING UNIBUS INTERRUPT SERVICE WITH DL11-W LINE CLOCK
17688
17689 !*****

```

17690  
17691  
17692  
17693  
17694  
17695  
17696  
17697  
17698  
17699  
17700  
17701  
17702  
17703  
17704  
17705  
17706  
17707  
17708  
17709  
17710  
17711  
17712  
17713  
17714  
17715  
17716  
17717  
17718  
17719  
17720  
17721  
17722  
17723  
17724  
17725  
17726  
17727  
17728  
17729  
17730  
17731  
17732  
17733  
17734  
17735  
17736  
17737  
17738  
17739  
17740  
17741

```

!*
!* TESTS: 761A - 7630                                UWORDS: 060 + 045
!*
!* FUNCTIONS: TESTS 761A - 7630 CAUSE AN INTERRUPT ON THE UNIBUS AT
!*              LEVEL BR6, USING THE DL11-W LINE CLOCK AS A GENERATOR,
!*              THEN TEST TO SEE THAT ALL THE APPROPRIATE BELLS AND
!*              WHISTLES AND PARAPHERNALEAXJAISQURE ALSO RESPOND.
!*
*****
!-----
! THIS FIRST SET OF THREE TESTS CLEAR OUT ALL THE I-O UCON REGISTERS:
! CLR-JAM-ERRORS, CLR-YELLOW-ZONE, CLEAR-CONSOLE-SERVICE, CLR-NPR-TIMEOUT, CLR-PWR-FAIL, AND
! ALSO THIS TIME DO AN "INIT" ON THE UNIBUS, VIA THE BUS-INIT-UCON FUNCTION.
! ALSO SET ALL THE FLAGS<8:0>H=ZERO, AND THE FULL PS<15:00>H=ZERO,
! (IE, PROCESSOR PRIO=0, T-BIT=0).
! THEN CHECK TO SEE THAT ALL THE RELEVANT BUS CONTROL INTERRUPT LOGIC IS RESET.
!-----
! TEST 761A CHECKS THAT SERVICE-H=NOT(INTR-HIGH-H*FLAG7(0)H*BG-SERVICE(0)H) IS LOW,
! WHEN INTR-HIGH-H=HIGH, FLAG7(0)H=HIGH, AND BG-SERVICE(0)H=HIGH
! IE, AFTER UNIBUS-INIT, AND CLEAR SERVICE CONDITIONS, W/ PSW PRIO=000,
! THERE SHOULD BE NOTHING PENDING
4745:
TEST761A:
    PO,          LOAD-ENUA(ZTARGET402),          !BIT<00> CLEAR
                LOAD-ERROR(TEST761A),          !ERROR DIRECTORY KEY
                DCS-CTR(C12.),                 !COMPARE AT TARGET
    P3,          BUTA(CLR-FLAG-RES-UCON),        !SET SR/LOAD, BUSDIN/EMIT
    NEXT,        J/ZERODSR761A
(4745) DCS[1.00.1.0.0.0] BM[0011..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...1!.010...111.111.010]

4772:
ZERODSR761A:
    P2-T,        D+ZERO, SR+ZERO, D(C)+ALU15,    !ZERO FOR FLAGS/PS/JAMUPP FLAG
    NEXT,        J/ZEROIT761A
(4772) DCS[0.00.0.0.0.0] BM[0011..00.00..00.00..000..100...0.1.1..0..0...0.0000...0..0000.0...11.000...100.000.011]

4403: !(FREE)
ZEROIT761A:
    PO,          BUMP-VERIFY,                   !COUNT
                BUSDIN+EMIT-[I],              !KEEP IT ON

```



17742 P3, FLAG(8-0)=0(15-8)-(1), !ZERO THE FLAGS  
 17743 PS=0-(1), !ZERO ALL OF THE PS  
 17744 NEXT, J/CLEAR761A

(4403) DCS(0.00.0.0.0.1) BM(1000..00.00..00.01..010..011...0.0.0..0..0...1.1011...0..0000.0...11.000...100.000.100)

17745  
 17746 4404: !(FREE)  
 17747 CLEAR761A:  
 17748 SETUP, RETURN/GOBUT761A, !GO TO SUBR WHICH DOES THE CLEARS AND BUS-INIT  
 17749 NEXT, CALL(CLEAR-I-O-A)

(4404) DCS(0.00.0.0.0.0) BM(0100..00.10..00.00..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.010.111)

17750  
 17751 4405: !(FREE)  
 17752 GOBUT761A:  
 17753 SETUP, RETURN/TEST761B, !RETURN TO START OF NEXT SUBTEST  
 17754 NEXT, GOTO-PAGE(7), !BUT TABLE  
 17755 J/BUTSERVICE !SERVICE-H IN BIT<00>

(4405) DCS(0.00.0.0.0.0) BM(0100..00.11..11.11..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.101)

17756  
 17757  
 17758  
 17759  
 17760  
 17761 ! - - - - -

17762 !TEST 761B CHECKS THAT VECTOR-LOAD(1)H=-(UNIBUS-INTR-L) IS LOW  
 17763 4771:  
 17764 TEST761B:  
 17765 PO, LOAD-ENUA(ZTARGET401), !BIT<01> CLEAR  
 17766 LOAD-ERROR(TEST761B), !ERROR DIRECTORY KEY  
 17767 DCS-CTR(C3.), !COMPARE AT TARGET  
 17768 BUMP-VERIFY, !COUNT  
 17769 NEXT, J/GOBUT761B

(4771) DCS(1.00.1.0.0.1) BM(1100..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...100.000.110)

17771  
 17772 4406: !(FREE)  
 17773 GOBUT761B:  
 17774 SETUP, RETURN/TEST761C, !RETURN TO START OF NEXT SUBTEST  
 17775 NEXT, GOTO-PAGE(7), !BUT TABLE  
 17776 J/BUTVECTLOAD !VECTOR-LOAD(1)H IN BIT<01>

(4406) DCS(0.00.0.0.0.0) BM(0100..00.11..11.11..100..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.110)

17777  
 17778  
 17779  
 17780  
 17781 ! - - - - -

17782 !TEST 761C CHECKS THAT BC-SERVICE(0)H=BR>PS-L IS HIGH WHEN NO DEVICES REQUEST INTR ON UNIBUS  
 17783 4774:  
 17784 TEST761C:  
 17785 PO, LOAD-ENUA(ZTARGET407), !BIT<02> SET (ACTIVE LOW)  
 17787

```

17788 LOAD-ERROR(TEST761C), !ERROR DIRECTORY KEY
17789 DCS-CTR(C3.), !COMPARE AT TARGET
17790 NEXT J/GOBUT761C
(4774) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.000...100.000.111)

```

```

17791 4407: !(FREE)
17792 GOBUT761C:
17793 SETUP, RETURN/SCOPE761, !RETURN TO SCOPE LOOP TEST WORD
17794 NEXT, GOTO-PAGE(7), !BUT TABLE
17795 J/BUTBGSERV1 !BG-SERVICE(01H IN BIT<02>)
17796 (4407) DCS(0.00.0.0.0.0) BM(0100..00.10..00.01..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.110)

```

```

17797 4410: !(FREE)
17798 SCOPE761:
17799 PO, BUMP-VERIFY !COUNT
17800 P3, CSPD(05)+EMIT, EMIT/177546, ! (177546) IS UNIBUS ADDR OF CSR FOR DL11-W
17801 NEXT, BUTD(SCOPE), !NO ERROR: "TEST762A" (+1. WORDS)
17802 J/TEST762A !ERROR: "ZERODSR761A" (-8. WORDS)
17803 (4410) DCS(0.00.0.1.0.1) BM(1111..10.11..11.01..100..110...0.0.0..0..0...0.1010...1..0000.0...11.000...111.111.011)

```

```

17806 !-----
17807 !THE FOLLOWING FIVE TESTS NOW CAUSE AN INTERRUPT ON THE UNIBUS, AND THEN
17808 !CHECK THAT THE BUS CONTROL LOGIC RESPONDS CORRECTLY.
17809 !FIRST THE LINE CLOCK INTR EN...E BIT<06> IS SET, AND THE PROCESSOR PRIORITY IS SET TO (6).
17810 !SINCE THE DL11-W IS A BR6 DEVICE, THE INTERRUPT SHOULD NOT COME THRU UNTIL THE PRIORITY IS
17811 !LOWERED TO LEVEL (5) OR LOWER. NOTE THAT THE MICROCODE MUST GO IN TO A WAIT LOOP FOR A
17812 !MINIMUM OF 16.7 MILLISEC @ 60 HZ, OR 20 MILLISEC @ 50 HZ TO GUARANTEE THAT AN INTERRUPT WILL
17813 !BE PENDING. THE DELAY, BASED UPON UP3-UP3 = 170 NS, WILL BE 22.3 MILLISEC; AN ADEQUATE MARGIN.
17814
17815
17816
17817
17818
17819
17820
17821
17822
17823
17824 !-----
17825

```

```

17826 !TEST 762A ENABLES THE INTERRUPT, THEN WAITS FOR THE DELAY PERIOD.
17827 4773:
17828 TEST762A:
17829 PO, LOAD-ENUA(ZAPD762A), !COMPARE JUST AFTER BUS CYCLE INVOKED
17830 LOAD-ERROR(TEST762A), !ERROR DIRECTORY KEY
17831 DCS-CTR(C5.), !COMPARE JUST AFTER "DATO" INITIATED, BELOW
17832 NEXT J/DW11L762A
17833 (4773) DCS(1.00.1.0.0.0) BM(1010..00.10..00.11..001..101...0.0.0..0..0...0.0000...0..0000.0...11.000...100.001.001)

```

```

17834 4411: !(FREE)
17835 DW11L762A:
17836

```

```

17837 P2-T, SR+CSPD(005), !GET SR=(177546), ADDR OF DL11-W CSR
17838 NEXT, GOTO-PAGE(7); !XFER
17839 J/MASK762A

```

(4411) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..111...0.0.1..0..0...0.1010...0..0000.0...11.100...010.000.010)

```

17840 7202: !(FREE)
17841 MASK762A:
17842 P3, CSPD(04)+EMIT, EMIT/177777, !READ ALL BITS IN REGISTERS READ
17843 NEXT, J/PRI06762A
17844

```

(7202) DCS(0.00.0.0.0.0) BM(1111..10.11..11.11..111..111...0.0.0..0..0...0.1011...1..0000.0...11.000...010.000.111)

```

17845 7207: !(FREE)
17846 PRI06762A:
17847 P3, CSPD(15)+EMIT, !EMITCON FOR: PS<7:5>="110"=(6), T-BIT=0
17848 EMIT/000300, AND BIT<06> SET FOR DL11-W INTR ENABLE
17849 NEXT, J/GETIT762A
17850

```

(7207) DCS(0.00.0.0.0.0) BM(0000..10.00..00.11..000..000...0.0.0..0..0...0.0010...1..0000.0...11.000...010.001.000)

```

17851 7210: !(FREE)
17852 GETIT762A:
17853 P1, BA+SR, !BA (- (177546), BITS<17:16> FORCED TO "11" ON IO-PAGE ADDRESS
17854 P2-T, D+CSPB(B15), ! (000300) INTO D ** NOTE: BIT<1:0> = 00 FOR BA LOAD **
17855 P3, DATO, !WRITE IT OUT
17856 NEXT, GOTO-PAGE(4), !XFER FOR DCS XTN BITS
17857 J/ZAPD762A
17858

```

(7210) DCS(0.00.0.0.0.0) BM(1010..11.10..00.00..000..100...0.1.0..1..0...1.0010...0..0000.0...11.100...011.001.101)

```

17859 4315: !(FREE)
17860 ZAPD762A:
17861 P3-T, D+ZERO, D(C)+ALU15, !ZERO D, D(C) FOR LOOP; MUST DO AFTER P3-T
17862 NEXT, J/NEXT0762A !ENTER AT 2ND WORD OF LOOP; LET D SETTLE
17863

```

(4315) DCS(0.00.0.0.0.0) BM(0011..00.00..00.00..000..100...1.1.0..0..0...0.0000...0..0000.0...11.000...100.110.111)

```

17864 !*** THE FOLLOWING TWO WORDS NOW GO INTO A COUNT LOOP TO WAIT FOR THE LINE CLOCK TO INTERRUPT
17865 !*** THE WAIT WILL BE A MAXIMUM DELAY, DEPENDING UPON THE PROCESSOR UP3-UP3 CYCLE TIME:
17866 !***
17867 !***
17868 !***
17869 !***
17870 !***
17871 !***
17872 !***
17873 !***
17874 !***
17875 !***
17876 !***
17877 !***
17878 !***
17879 !***
17880 !***
17881 !***
17882 !***
17883 !* ENTER HERE FOR ANOTHER LOOP *

```

PROCESSOR CYCLE TIME (NANOSEC)	TIME DELAY (MILLISEC)
150	19.7
160	21.0
* 170 *	* 22.3 *
180	23.6
190	24.9

<NOMINAL VALUE>

NOTE THAT THE ABOVE LOOP TIME IS A MAXIMUM VALUE; WE WILL EXIT EARLY IF THE INTERRUPT COMES THROUGH BEFORE WE OVERFLOW THE COUNTER. IF NO INTERRUPT COMES THROUGH BY THE TIME THE COUNT HAS OVERFLOWN, IT WILL BE CONSIDERED AN ERROR.

KD11-K MICRO V00A-1 00:00:03 12-MAR-77

```

17884 4551:
17885 BUMP0762A:
17886 P3-T, D=D-PLUS-1, D(C)+COUT15, !BUMP D, SAVE CARRYOUT
17887 NEXT, BUTB(BG-SERVICE-L), !NEGATED: "NEXT0762A"
17888 J/SETPR6-762A !ASSERTED: "SETPR6-762A"
(4551) DCS(0.00.0.0.0.0) BM(1001..01.11..01.01..000..110...1.1.0..0..0...0.0000...0..0000.0...01.100...100.110.011)

```

```

17889 4467:
17890 NEXT0762A:
17891 PO, DCS-CTR(C15.), !STALL; NOTE: NO BUMP-VERIFIES IN THIS LOOP
17892 NEXT, BUTR(D(C)-8), !SET: "TEST762A1" D OVERFLOWN, ERROR
17893 J/BUMP0762A !CLEAR: "BUMP0762A" GO FOR NEXT LOOP
17894 (4467) DCS(0.00.1.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...10.011...101.101.001)

```

```

17895 !* COME HERE IF D OVERFLOWN *
17896 4553:
17897 TEST762A1:
17898 PO, LOAD-ERROR(TEST762A1), !ERROR DIRECTORY KEY
17899 DCS-CTR(C0.), !SIGNAL ERROR NOW
17900 NEXT, J/TEST762A !FORCE A SCOPE LOOP
17901 (4553) DCS(1.00.1.0.0.0) BM(1111..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...111.111.011)

```

```

17903 !*** COME HERE IF EXIT LOOP OK: D NOT OVERFLOWED, INTERRUPT PENDING ***
17904 4463:
17905 SETPR6-762A:
17906 P2-T, D=CSPD(D15), D(C)+0, !GET D<7:5>=(6), D<4>=(0)
17907 NEXT, J/SETPRI762A
17908 (4463) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..000...0.1.0..0..0...0.0010...0..0000.0...11.000...100.001.011)

```

```

17910 4413: !(FREE)
17911 SETPRI762A:
17912 PO, BUSDIN+EMIT-[I], !KEEP IT ON
17913 P2, PS(3-0)+D(3-0)-[I], !FOR USE IN TEST763A, PS(CC)="0000" HERE
17914 P3, PS(7-4)+D(7-4)-[I], !PRIO=6, T-BIT=0
17915 NEXT, J/TEST762B !NOTE: THE BR6 PENDING INTR SHOULD NOW HIDE UNDER PROCESSOR PRIO(6)
17916 (4413) DCS(0.00.0.0.0.0) BM(1000..00.00..00.01..010..000...0.0.0..0..0...1.1011...0..0000.0...11.000...111.101.001)

```

```

17917 ! - - - - -
17918 !
17919 !
17920 !TEST 762B CHECKS THAT BG-SERVICE(0)H=BR>PS-L IS HIGH WHEN THE PROCESSOR PRIORITY(=6) IS AS HIGH
17921 !AS THE ONLY DEVICE WISHING TO REQUEST AN INTERRUPT (AT BR6).
17922 4751:
17923 TEST762B:
17924 PO, LOAD-ENUA(ZTARGET407), !BIT<02> SET (ACTIVE LOW)
17925 LOAD-ERROR(TEST762B), !ERROR DIRECTORY KEY
17926 DCS-CTR(C3.), !COMPARE AT TARGET
17927 BUMP-VERIFY,
17928 NEXT, J/GOBUT762B
17929 (4751) DCS(1.00.1.0.0.1) BM(1100..00.11..11.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.000...111.110.010)

```

```

17930 4762:
17931

```

```

17932 GOBUT762B:
17933     SETUP, RETURN/TEST762C,           !RETURN TO START OF NEXT SUBTEST
17934     NEXT,  GOTO-PAGE(7),             !BUT TABLE
17935     J/BUTBGSERVL                     !BG-SERVICE(0)H IN BIT<02>
(4762) DCS(0.00.0.0.0.0) BM(0100..00.11..11.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.110)

```

```

17936
17937
17938
17939
17940
17941 ! - - - - -
17942

```

```

17943 !TEST 762C CHECKS THAT BG-SERVICE(0)H=BR>PS-L IS LOW WHEN A BR6 DEVICE (DL11-W) IS REQUESTING AN INTR,
17944 !AND THE PROCESSOR PRIORITY(=5) IS < THE BR LEVEL(=6). NO OTHER DEVICES PRESENT.
17945 4740:
17946 TEST762C:

```

```

17947     PO, LOAD-ENUA(ZTARGET403),           !BIT<02> CLEAR (ACTIVE LOW)
17948     LOAD-ERROR(TEST762C),             !ERROR DIRECTORY KEY
17949     DCS-CTR(C7.),                     !COMPARE AT TARGET
17950     NEXT, J/FILL762C
(4740) DCS(1.00.1.0.0.0) BM(1000..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...100.001.100)

```

```

17951 4414: !(FREE)
17952 FILL762C:
17953     P2-T, D+ASPHI(C125252), D(C)+ALU00, !BIT<7:5>="101"=(5), BIT<4>=0
17954     NEXT, J/DELAY762C
(4414) DCS(0.00.0.0.0.0) BM(1111..00.00..11.01..110..010...0.1.0..0..0...0.0000...0..0000.0...11.000...100.001.101)

```

```

17956 4415: !(FREE)
17957 DELAY762C:
17958     SETUP, RETURN/SETPRI762C,         !EXEC 3. UWORDS AFTER SETTING PRIO, FOR DELAY
17959     P2, PS(3-0)+D(3-0),               !FOR USE IN TEST763A, PS(CC)="1010"
17960     P3, PS(7-4)+D(7-4),               !SET PRIO=(5), AT P3-T OF THIS UWORD
17961     NEXT, GOTO-PAGE(7),                !GO DO A JUMP, AND A BUTA(RETURN)
17962     J/BUTD-IS-ZERO                     !DON'T REALLY CARE ABOUT THE RESULT OF THIS
(4415) DCS(0.00.0.0.0.0) BM(0111..00.01..00.00..110..111...0.0.0..0..0...1.1010...0..0000.0...11.100...011.100.001)

```

```

17964 7206: !(FREE)
17965 SETPRI762C:
17966
17967     SETUP, RETURN/TEST762D,           !BG-SERVICE(0)H SHOULD BE ASSERTED BY NOW
17968     NEXT, GOTO-PAGE(7),               !RETURN TO START OF NEXT SUBTEST
17969     J/BUTBGSERVL                       !BUT TABLE
17970     J/BUTBGSERVL                       !BG-SERVICE(0)H IN BIT<02>
(7206) DCS(0.00.0.0.0.0) BM(0100..00.11..11.00..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.110)

```

```

17971
17972
17973
17974
17975
17976 ! - - - - -
17977

```

```

17978 !TEST 762D CHECKS THAT BG-SERVICE(1)H+FLTPT-SERVICE-H IS HIGH, WHEN BG-SERVICE(1)H IS SET

```

KD11-K MICRO V00A-1 00:00:03 12-MAR-77

```

17979 4741:
17980 TEST762D:
17981 PO, LOAD-ENUA(ZTARGET417), !BIT<03> SET
17982 LOAD-ERROR(TEST762D), !ERROR DIRECTORY KEY
17983 DCS-CTR(C3.), !COMPARE AT TARGET
17984 NEXT J/GOBUT762D
(4741) DCS(1.00.1.0.0.0) BM(1100..00.11..11.00..001..111...0.0.0..0..0...0.0000...0..0000.0...11.000...100.001.110)

```

```

17985 4416: !(FREE)
17986 GOBUT762D:
17987 SETUP, RETURN/TEST762E, !RETURN TO START OF NEXT SUBTEST
17988 NEXT, GOTO-PAGE(7), !BUT TABLE
17989 J/BUTBGFPSEV !BG-SERVICE(1)H+FLTPT-SERVICE-H IN BIT<03>
17990 (4416) DCS(0.00.0.0.0.0) BM(0100..00.11..10.11..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.000.111)

```

17991  
17992  
17993  
17994  
17995

! - - - - -

```

17998 !TEST 762E NOW READS THE "SERVICE" PORT OF THE STATUS MUX TO SEE:
17999 !SERVICE<15:00>H = "0 100 011 111 100 000"

```

```

18001 IMPORTANT BITS ARE:
18002 B15 = DATI(1)H = 0 B14 = BG-SERVICE(1)H = 1 B11 = DATOB(1)H = 0
18003 B10 = DATO(1)H = 1 B09 = PBA<17>H = 1 B08 = PBA<16>H = 1

```

```

18005 4730:
18006 TEST762E:
18007 PO, LOAD-ENUA(ZTARGET402), !SETUP FOR D=ZERO TEST
18008 LOAD-ERROR(TEST762E), !ERROR DIRECTORY KEY
18009 DCS-CTR(C10.), !COMPARE AT TARGET
18010 NEXT J/EXPEC762E
(4730) DCS(1.00.1.0.0.0) BM(0101..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.001.111)

```

```

18011 4417: !(FREE)
18012 EXPEC762E:
18013 P3, CSPD(02)+EMIT, !EXPECTED VALUE OUT OF SERVICE PORT:
18014 EMIT/043740, !"0100 0111 1110 0000"
18015 NEXT J/GOGET762E
18016 (4417) DCS(0.00.0.0.0.0) BM(0100..10.01..11.11..100..000...0.0.0..0..0...0.1101...1..0000.0...11.000...100.010.000)

```

```

18017 4420: !(FREE)
18018 GOGET762E:
18019 PO, BUMP-VERIFY, !
18020 SETUP, RETURN/TEST762F, !GO TO SUBR WHICH:
18021 NEXT, CALL(SERVICET00) !CSP(02).XOR.SERVICE -> D, BUT(D=ZERO)
18022 (4420) DCS(0.00.0.0.0.1) BM(0100..00.11..10.01..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.101.000)

```

18023  
18024  
18025

```

18026 ! -----
18027
18028 !TEST 762F CHECKS THAT SERVICE-H=NOT(INTR-HIGH-H*FLAG7(0)H*BG-SERVICE(0)H) IS HIGH,
18029 ! WHEN INTR-HIGH-H=HIGH, FLAG7(0)H=HIGH, AND BG-SERVICE(0)H=LOW
18030 4711:
18031 TEST762F:
18032     PO,          LOAD-ENUA(ZTARGET403),          !BIT<00> SET
18033                LOAD-ERROR(TEST762F),          !ERROR DIRECTORY KEY
18034                DCS-CTR(C3.),                  !COMPARE AT TARGET
18035     NEXT,        J/GOBUT762F
(4711) DCS[1.00.1.0.0.0] BM[1100..00.11..11.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.000...100.010.001]

18036 4421: !(FREE)
18037 GOBUT762F:
18038     SETUP,      RETURN/TEST763A,              !RETURN TO START OF NEXT SUBTEST
18039     NEXT,       GOTO-PAGE(7),                 !BUT TABLE
18040                J/BUTSERVICE                 !SERVICE-H IN BIT<00>
18041 (4421) DCS[0.00.0.0.0.0] BM[0100..00.11..11.10..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.101]

18042
18043
18044
18045
18046
18047 ! -----
18048
18049 !THIS NEXT SEQUENCE OF TWO TESTS RESPONDS TO THE BUS INTERRUPT REQUEST BY:
18050 ! (1) ASSERTING "ALLOW-BG[1]H", THUS ALLOWING THE BUS GRANT TO THE DL11-W, SO THAT
18051 ! (2) "VECTOR-LOAD[1]H" WILL BE ASSERTED, INDICATING THAT THE DEVICE HAS PUT ITS
18052 !     VECTOR ON UNIBUS DATA<8:0>L, AND THEN
18053 ! (3) ACTUALLY READING THE VECTOR FOR THE DL11-W (100)(8), AND VALIDATING ITS CORRECTNESS.
18054
18055
18056
18057 ! -----
18058
18059 !TEST 763A CHECKS THAT AFTER "ALLOW-BG[1]H" IS GIVEN TO THE INTERRUPTING DEVICE,
18060 ! THEN "VECTOR-LOAD[1]H" IS ASSERTED
18061 4761:
18062 TEST763A:
18063     PO,          LOAD-ENUA(VECTLOAD763A),      !COMPARE AT "VECTOR-LOAD[1]H" IN BIT<01> SET
18064                LOAD-ERROR(TEST763A),          !ERROR DIRECTORY KEY
18065                DCS-CTR(C5.),                  !COMPARE AT TARGET
18066     NEXT,        J/EXPEC763A
(4761) DCS[1.00.1.0.0.0] BM[1010..00.10..01.11..011..011...0.0.0..0..0...0.0000...0..0000.0...11.000...100.010.010]

18067 4422: !(FREE)
18068 EXPEC763A:
18069     PO,          BUMP-VERIFY,                   !COUNT
18070     P3,          CSPD[01]+EMIT, EMIT/100,      !DL11-W VECTOR IS (100)8
18071     NEXT,        J/ALLOW763A
(4422) DCS[0.00.0.0.0.1] BM[0000..10.00..00.01..000..000...0.0.0..0..0...0.1110...1..0000.0...11.000...111.011.001]

18073 4731:
18074                                     !*** ** *01

```

```

18075 ALLOW763A:
18076 PD, BUSDIN+PS-[1], !NOISE BITS ON BUSDIN TO IMPEDE READING VECTOR
18077 !AT THIS POINT PS=(000252)
18078 P2, ALLOW-BG[1]H-[1], !GIVE BUS GRANT TO PENDING INTERRUPT
18079 NEXT, J/READYECT763A
(4731) DCS[0.00.0.0.0.0] BM[0100..00.01..10.01..000..000...0.0.0..0..0...1.1011...0..0000.0...11.000...100.010.011]

18080 4423: !(FREE)
18081 READYECT763A:
18082 P3, CSPD[07]+BUSDIN, !MUST READ BUSDIN=VECTOR RITE HERE, OR
18083 !WE WILL LOSE IT
18084 EMIT/052525, !NOISE ON EMIT
18085 NEXT, J/TESTVECT763A
(4423) DCS[0.00.0.0.0.0] BM[0101..10.01..01.01..010..101...0.0.0..0..0...0.1000...1..0000.0...11.000...100.010.100]

18087 4424: !(FREE)
18088 TESTVECT763A:
18089 P2-T, SR+CSPD(D01), !SR=(000100), EXPECTED VECTOR
18090 NEXT, BUTR(VECTOR-LOAD), !IF SET, "VECTLOAD763A", THIS IS EXPECTED
18091 !IF CLR, "ALLOW763A", VECTOR-LOAD[1]H NOT SET, ERROR
18092 (4424) DCS[0.00.0.0.0.0] BM[1010..10.00..00.00..000..000...0.0.1..0..0...0.1110...0..0000.0...10.001...111.011.001]

18093 ! - - - - -
18094 ! - - - - -
18095 ! - - - - -
18096 ! - - - - -
18097 ! - - - - -
18098 !IF VECTOR-LOAD[1]H="1", COME TO HERE.
18099 !TEST 763B NOW CHECKS THAT THE CORRECT VECTOR FOR THE DL11-W = 100(8) WAS READ
18100 4733: !*** ** *11
18101 VECTLOAD763A:
18102 P2-T, D+SR-XOR-CSPD(D07), !COMPARE EXPECTED VECTOR:RECEIVED VECTOR
18103 NEXT, J/TEST763B
(4733) DCS[0.00.0.0.0.0] BM[0110..10.00..00.00..000..000...0.1.0..0..0...0.1000...0..0000.0...11.000...111.010.000]

18106 4720:
18107 TEST763B:
18108 PD, LOAD-ENVA(ZTARGET402), !SETUP FOR D=ZERO TEST
18109 LOAD-ERROR(TEST763B), !ERROR D XECTORY KEY
18110 DCS-CTR(C3.), !COMPARE AT TARGET
18111 NEXT, J/GOBUT763B
(4720) DCS[1.00.1.0.0.0] BM[1100..00.11..11.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...100.010.101]

18113 4425: !(FREE)
18114 GOBUT763B:
18115 SETUP, RETURN/TEST763C, !RETURN TO START OF NEXT SUBTEST
18116 NEXT, GOTO-PAGE(7), !BUT TABLE
18117 J/BUTD-IS-ZERO !CHECK EQUALITY
18118 (4425) DCS[0.00.0.0.0.0] BM[0100..00.11..10.10..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.001]

18119
18120
18121

```



```

18122
18123
18124 ! - - - - -
18125
18126 !TEST 763C CHECKS THAT VECTOR-LOAD[1]H=-(UNIBUS-INTR-L) IS AGAIN LOW
18127 4721:
18128 TEST763C:
18129     PO,          LOAD-ENUA(ZTARGET401),      !BIT<01> CLEAR
18130     LOAD-ERROR(TEST763C),      !ERROR DIRECTORY KEY
18131     DCS-CTR(C3.),              !COMPARE AT TARGET
18132     BUMP-VERIFY,              !COUNT
18133     NEXT          J/GOBUT763C
(4721) DCS[1.00.1.0.0.0] BM[1100..00.11..11.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...100.010.110]

18134
18135 4426: !(FREE)
18136 GOBUT763C:
18137     SETUP,      RETURN/TEST763D,      !RETURN TO START OF NEXT SUBTEST
18138     NEXT,        GOTO-PAGE(7),         !BUT TABLE
18139     J/BUTVECTLOAD      !VECTOR-LOAD[1]H IN BIT<01>
(4426) DCS[0.00.0.0.0.0] BM[0100..00.11..10.01..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.100.110]

18140
18141
18142
18143
18144 ! - - - - -
18145
18146 !TEST 763D CHECKS THAT BG-SERVICE(0)H=BR>PS-L IS HIGH WHEN UNIBUS INTR REQUEST HAS BEEN SATISFIED
18147 4710:
18148 TEST763D:
18149     PO,          LOAD-ENUA(ZTARGET407),      !BIT<02> SET (ACTIVE LOW)
18150     LOAD-ERROR(TEST763D),      !ERROR DIRECTORY KEY
18151     DCS-CTR(C3.),              !COMPARE AT TARGET
18152     NEXT          J/GOBUT763D
(4710) DCS[1.00.1.0.0.0] BM[1100..00.11..11.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.000...100.010.111]

18153
18154 4427: !(FREE)
18155 GOBUT763D:
18156     SETUP,      RETURN/SCOPE763,      !RETURN TO SCOPE LOOP TEST WORD
18157     NEXT,        GOTO-PAGE(7),         !BUT TABLE
18158     J/BUTBGSERV      !BG-SERVICE[0]H IN BIT<02>
(4427) DCS[0.00.0.0.0.0] BM[0100..00.10..00.11..000..111...0.0.0..0..0...0.0000...0..0000.0...11.100...011.001.110]

18159
18160
18161
18162 4430: !(FREE)
18163 SCOPE763:
18164     PO,          BUSDIN+EMIT-[1],      !RESET PROC UCON FOR EMIT
18165     P2,          PS=0-[1],             !ZAP THE PS TO ZERO
18166     NEXT,        BUTD(SCOPE),          !NO ERROR: "KILL764A" (+1. WORDS)
18167     J/KILL764A      !ERROR: "EXPEC763A" (-11. WORDS)
(4430) DCS[0.00.0.1.0.0] BM[1000..00.00..00.01..010..010...0.0.0..0..0...1.1011...0..0000.0...11.000...111.110.011]

18168

```

18169  
18170  
18171  
18172  
18173  
18174  
18175  
18176  
18177  
18178  
18179  
18180  
  
18181  
18182  
18183  
18184  
18185  
18186  
18187  
18188  
18189  
18190  
18191  
18192  
18193  
18194  
18195  
18196  
18197  
18198  
18199  
18200  
18201  
18202  
18203  
18204  
18205  
18206  
18207  
18208  
18209  
18210  
18211  
18212  
18213  
18214  
18215  
18216  
18217  
18218  
18219  
18220  
18221

! -----  
!\*\*\* ALL DONE WITH LINE CLOCK, DISABLE IT FROM FURTHER INTERRUPTS BEFORE LEAVING FOR EOP \*\*\*

4763;  
KILL764A:  
P2-T, D+ZERO, D(C)+ALUDD, !NOW MUST GO KILL THE LINE CLOCK  
P3, DATO !BA=(177546), DL11-W CSR  
BUTA(CLR-FLAG-RES-UCON), !ZAP UCON, RES-CONTROL FOR EOP ROUTINE (MUST BE DONE)  
NEXT J/EOP001

(4763) DCS(0.00.0.0.0.0) BM(0011..00.00..00.00..000..010...0.1.0..0..0...1.0010...0..0000.0...11.010...100.011.0011)

!.PAGE=====

.TOC \* END OF PASS CODE

\*\*\*\*\*

FINAL ENTRY HERE TO TEST FOR MULTIPLE PASSES, VERIFY MODE, ET AL

\*\*\* INDICATIONS ON CONSOLE DISPLAY \*\*\*

	0.0.0.0.0.0.	DCS IS RUNNING; REQUIRES APPROX 8. SECONDS/64. PASSES
		"RUN" "PROC" "USER" "CONSOLE" "BATTERY"
		(ON) (BLINK) (BLINK) (OFF) (OFF)
OR	0 0 0 0 0 0	ERROR DETECTED / PASS=1 / SCOPE LOOPING
	2 1 2 1 2 1	
OR	0.0.0.0.0.0.	ERROR DETECTED / 1<PASS<64. / SCOPE LOOPING
	2.1.2.1.2.1.	
	1.2.3.3.2.1.	SUCCESSFUL 64. PASSES COMPLETED

AT END OF PASS, THE GENERAL REGISTERS (BASE MACHINE R0-R7)  
CONTAIN THE FOLLOWING INFORMATION:

R0 = (NU)  
R1 = (NU)  
R2 = (NU)  
R3 = (NU)  
R4 = (NU)  
R5 = REVISION NUMBER OF DCS MICROCODE, WITH

```

18222      BIT15 SET TO INDICATE END OF PASS
18223      R6 = (NU)
18224      R7 = (123321), END OF PASS INDICATION CONSTANT
18225      !*****
18226      !*****
18227      !*****
18228      !*****
18229      4431: !(FREE)
18230      EOP001:
18231      PD, DCS-CTR(C15.), !HOLD UP ERROR COMPARE
18232      NEXT, GOTO-PAGE(6), !XFER
18233      J/EOP002
(4431) DCS(0.00.1.0.0.0) BM(0000..00.00..00.00..000..110...0.0.0..0..0...0.0000...0..0000.0...11.100...100.101.010)

18234
18235
18236      6452: !(FREE)
18237      EOP002:
18238      P3, CSPD(01)+EMIT, EMIT/123321, !SETUP SUCCESSFUL END OF PASS CONSTANT
18239      NEXT, J/EOP003
(6452) DCS(0.00.0.0.0.0) BM(1010..10.01..10.11..010..001...0.0.0..0..0...0.1110...1..0000.0...11.000...101.001.000)

18240
18241
18242      6510: !(FREE)
18243      EOP003:
18244      P2-T, D+CSPD(D01), D(C)+0, !GET EOP CONSTANT INTO
18245      P3, PC+D, !GPR PC, ON BOTH SP SIDES
18246      NEXT, J/EOP004 !FOR DISPLAY FROM "CONSOLE HALT" ROUTINE
(6510) DCS(0.00.0.0.0.0) BM(1010..10.00..10.01..111..000...0.1.0..0..0...0.1110...0..0011.0...11.000...101.001.001)

18247
18248
18249      6511: !(FREE)
18250      EOP004:
18251      SETUP, RETURN/EOP005, !GO TO DISPLAY-D-IN-LIGHTS ROUTINE
18252      NEXT, CALL(DISPLAY) !WHICH SHOULD PUT (212121) INTO DISPLAY
(6511) DCS(0.00.0.0.0.0) BM(0111..00.01..00.01..001..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.010.001)
!FOR INTERMEDIATE EOP INDICATION

18253
18254
18255      7211: !(FREE)
18256      EOP005:
18257      PD, SIGNAL-EOP !TRY TO SIGNAL EOP
18258      P2-T, SR+ALL-ONES !SET SR TO ALL ONES, FOR FULL BASE MACHINE INIT
18259      NEXT, GOTO-PAGE(6) !XFER TO 6
18260      BUTD(VERIFY-MODE), !IF TRUE, "VFY001" (NEXT PAGE FOLLOWING)
18261      J/EOP006 !IF FALSE, "EOP006" (NEXT MICROWORD)
(7211) DCS(0.00.0.0.1.0) BM(1111..00.00..11.01..101..110...0.0.1..0..0...0.0000...0..0000.0...11.100...111.111.110)

18262
18263
18264      6776:
18265      EOP006:
18266      PD, DCS-CTR(C11.), !HOLD UP ERROR COMPARE
18267      P3-U, SET-CONSOLE-DP-LEDS, !UCON SET DP LEDS, INDICATING >=1 SUCCESSFUL PASSES

```

18268 NEXT, BUTD(EOP-OVERFLOW), !IF TRUE, "EOP007" FOLLOWING, DONE 64. PASSES  
 18269 (6776) DCS(0.00.1.0.0.0) J/EOP007 BM(0100..00.00..00.00..110..001...0.0.0..0..0...1.1011...0..0000.0...11.000...101.001.010) !IF FG SE, "TEST001" GO FOR NEXT PASS AT (4000)  
 18270 !BEGIN COUNTDOWN  
 18271 ! 3 ... 2 ... 1 .

18272  
 18273  
 18274 6512: !(FREE)  
 18275 EOP007:  
 18276 SETUP, RETURN/EOP010, !GO TO SUBR THAT PUTS REV-NUMBER, WITH  
 18277 NEXT, GOTO-PAGE(7), ! B(15)=(1), INTO B.M. GPR "R5"  
 18278 J/INSERTEOPREVNO  
 (6512) DCS(0.00.0.0.0.0) BM(0111..00.01..00.01..010..111...0.0.0..0..0...0.0000...0..0000.0...11.100...010.001.100)

18279  
 18280  
 18281 7212: !(FREE)  
 18282 EOP010:  
 18283 SETUP, RETURN/CON99, !RETURN TO "FORCE CONSOLE-MODE HALT" ROUTINE IN BASE MACHINE  
 18284 NEXT, GOTO-PAGE(3), !GOTO "INIT---" ROUTINE FOR FULL BASE MACHINE  
 18285 J/INIT01 ! MICROCODE INITIALIZATION, BUTA(RETURN) AT END TO "CON99"  
 (7212) DCS(0.00.0.0.0.0) BM(0001..00.00..01.00..000..011...0.0.0..0..0...0.0000...0..0000.0...11.100...100.001.010)

18291 !.PAGE=====

18292  
 18293  
 18294 .TOC \* VERIFY MODE CODE  
 18295  
 18296 !  
 18297 ! VERIFY MODE ENTERS HERE:  
 18298 !  
 18299 !

18300 6774:  
 18301 VFY001:  
 18302 PD, DCS-CTR(C4.), !LOAD COUNTER FOR COMPARE IN 4. MICROWORDS  
 18303 NEXT, GOTO-PAGE(4), !ZFER  
 18304 J/VFY002  
 (6774) DCS(0.00.1.0.0.0) BM(1011..00.00..00.00..000..100...0.0.0..0..0...0.0000...0..0000.0...11.100...100.001.010)

18305  
 18306  
 18307 4412: !(FREE)  
 18308 VFY002:  
 18309 P3, CLEAR-CONSOLE-LED, !MAKE "CONSOLE" LED BLINK, JUST FOR FUN  
 18310 NEXT, J/VFY003  
 (4412) DCS(0.00.0.0.0.0) BM(0100..00.00..00.00..010..001...0.0.0..0..0...1.1011...0..0000.0...11.000...010.101.101)

18311  
 18312 4255:

```

18313 VFY003:
18314 PO, LOAD-ENUA(ZTARGET523), !ERROR CODE = [4255], ENUA = [7523]
18315 LOAD-ERROR(VFY003), !ERROR DIRECTORY KEY
18316 BUMP-VERIFY, !GIVE A VERIFY PULSE
18317 NEXT, J/VFY004
(4255) DCS[1.00.0.0.0.1] BM[0000..00.11..11.01..010..011...0.0.0..0..0...0.0000...0..0000.0...11.000...100.011.011]

```

```

18318
18319
18320 4433: !(FREE)
18321 VFY004:
18322 SETUP, RETURN/VFY005, !RETURN TO INLINE
18323 NEXT, GOTO-PAGE(7), !BUT'S ARE ON PAGE 7
18324 J/ZTARGET522, !TNUA = [7522], NOT EQUAL TO ENUA
(4433) DCS[0.00.0.0.0.0] BM[0111..00.01..11.11..101..111...0.0.0..0..0...0.0000...0..0000.0...11.100...101.010.010]

```

```

18325
18326
18327 ! NEXT MICROWORD COMES FROM ZTARGET522, AT WHICH THE ENABLED COMPARE
18328 ! TAKES PLACE. ENUA WAS SETUP NOT EQUAL TO TNUA, SO ERROR SHOULD BE SIGNALLED
18329
18330
18331

```

```

18332 7375:
18333 VFY005:
18334 P3, SET-CONSOLE-LED, !THE OTHER HALF OF MAKING IT BLINK
18335 NEXT, BUTD(SCOPE), !NO ERROR: "VFY005" [SELF LOOP, SHOULDN'T HAPPEN]
18336 J/VFY005, ! ERROR: "VFY006" [SHOULD HAPPEN]
(7375) DCS[0.00.0.1.0.0] BM[0100..00.00..00.00..100..001...0.0.0..0..0...1.1011...0..0000.0...11.000...011.111.101]

```

```

18337
18338
18339 7374:
18340 VFY006:
18341 PO, SIGNAL-EOP, !GIVE AN EOP PULSE, AFTER ERROR SIGNALLED
18342 NEXT, GOTO-PAGE(6), !LOOP BACK
18343 J/VFY001, ! ON CONTINUOUS VERIFY
(7374) DCS[0.00.0.0.1.0] BM[0000..00.00..00.00..000..110...0.0.0..0..0...0.0000...0..0000.0...11.100...111.111.100]

```

```

18344
18345
18346
18347
18348
18349 !.PAGE=====
18350

```

```

18351
18352 .TOC * DCS MICROCODE REVISION NUMBER
18353
18354 ! THE FOLLOWING ROUTINE WILL PUT THE CURRENT DCS MICROCODE REVISION NUMBER
18355 ! INTO B.M. GPR "R5", FROM THE EMIT FIELD OF THE MICROWORD.
18356
18357 ! THE ENTRY "INSERTREVNO" IS USED, EXCEPT AT END OF PASS; IE B<15>=(0)
18358 ! THE ENTRY "INSERTEOPREVNO" IS USED ONLY AT END OF PASS; IE, B<15>=(1)
18359
18360

```

```

18361 7214: !(FREE)
18362 INSERTOPREVNO:
18363 PO, BUSDIN+EMIT-(!), !SELECT EMIT
18364 NEXT, J/INSERTO2
(7214) DCS(0.00.0.0.0.0) BM(0000..00.00..00.01..000..000...0.0.0..0..0...1.1001...0..0000.0...11.000...010.001.101)

```

```

18365 7215: !(FREE)
18366 INSERTO2:
18367 P3, CSPD(17)+EMIT, EMIT/REV-NUMBER-AND-B15, !DCS REVISION NUMBER, B15 SET
18368 NEXT, J/INSERTO4
18369 (7215) DCS(0.00.0.0.0.0) BM(1000..10.00..00.01..000..001...0.0.0..0..0...0.0000...1..0000.0...11.000...010.010.000)

```

```

18370 7216: !(FREE)
18371 INSERTREVNO:
18372 PO, BUSDIN+EMIT-(!), !SELECT EMIT
18373 NEXT, J/INSERTO3
18374 (7216) DCS(0.00.0.0.0.0) BM(0000..00.00..00.01..000..000...0.0.0..0..0...1.1001...0..0000.0...11.000...010.001.111)
18375

```

```

18376 7217: !(FREE)
18377 INSERTO3:
18378 P3, CSPD(17)+EMIT, EMIT/REV-NUMBER, !DCS REVISION NUMBER, B15 CLEAR
18379 NEXT, J/INSERTO4
18380 (7217) DCS(0.00.0.0.0.0) BM(0000..10.00..00.01..000..001...0.0.0..0..0...0.0000...1..0000.0...11.000...010.010.000)

```

```

18381 7220: !(FREE)
18382 INSERTO4:
18383 P2-T, D+CSPD(D17), D(C)+0, !GET IT
18384 P3, RS+0, !AND STUFF IT
18385 NEXT, J/RESETUONP !AND RETURN
18386 (7220) DCS(0.00.0.0.0.0) BM(1010..10.00..10.01..110..000...0.1.0..0..0...0.0000...0..0011.0...11.000...010.111.001)
18387

```

```

18388
18389
18390
18391 !.PAGE=====
18392
18393
18394 .TOC * COMMON SUBROUTINES
18395
18396 .TOC * CONSOLE DISPLAY SUBROUTINE
18397
18398
18399 !
18400 ! DISPLAYS NUMBER REPRESENTED BY D<05:00>#D<05:00>#D<05:00>
18401 ! AS SIX OCTAL DIGITS IN CONSOLE 7 SEGMENT DISPLAY
18402 !

```

```

18403 7221: !(FREE)
18404 DISPLAY:
18405 P3-U, CLEAR-CONSOLE-COUNTER, !POINT TO DIGITS ....XX
18406 NEXT, J/DISPOO2
(7221) DCS(0.00.0.0.0.0) BM(0100..00.00..00.00..010..000...0.0.0..0..0...1.1011...0..0000.0...11.000...010.010.010)

```

```

18407
18408

```

```

18409 7222: !(FREE)
18410 DISPO02:
18411 P3-U, STROBE-CONSOLE-DISPLAY, !WRITE OUT DIGITS ....XX
18412 NEXT J/DISPO03
(7222) DCS(0.00.0.0.0.0) BM(0100..00.00..00.00..000..001...0.0.0..0..0...1.1011...0..0000.0...11.000...010.010.011)

```

```

18413
18414
18415 7223: !(FREE)
18416 DISPO03:
18417 P3-U, INCREMENT-CONSOLE-COUNTER, !POINT TO DIGITS ..XX..
18418 NEXT J/DISPO04
(7223) DCS(0.00.0.0.0.0) BM(0100..00.00..00.00..100..000...0.0.0..0..0...1.1011...0..0000.0...11.000...010.010.100)

```

```

18419
18420
18421 7224: !(FREE)
18422 DISPO04:
18423 P3-U, STROBE-CONSOLE-DISPLAY, !WRITE OUT DIGITS ..XX..
18424 NEXT J/DISPO05
(7224) DCS(0.00.0.0.0.0) BM(0100..00.00..00.00..000..001...0.0.0..0..0...1.1011...0..0000.0...11.000...010.010.101)

```

```

18425
18426
18427 7225: !(FREE)
18428 DISPO05:
18429 P3-U, INCREMENT-CONSOLE-COUNTER, !POINT TO DIGITS XX....
18430 NEXT J/DISPO06
(7225) DCS(0.00.0.0.0.0) BM(0100..00.00..00.00..100..000...0.0.0..0..0...1.1011...0..0000.0...11.000...010.010.110)

```

```

18431
18432
18433 7226: !(FREE)
18434 DISPO06:
18435 P3-U, STROBE-CONSOLE-DISPLAY, !WRITE OUT DIGITS XX....
18436 NEXT J/RESETUONP !GO RESET PROC UCON/EMIT, DO A (RETURN)
(7226) DCS(0.00.0.0.0.0) BM(0100..00.00..00.00..000..001...0.0.0..0..0...1.1011...0..0000.0...11.000...010.111.001)

```

```

18437
18438
18439
18440
18441
18442 !.PAGE=====

```

```

18443
18444
18445 .TOC * CLEAR I-O / BUS CONTROL / SERVICE AREA STATUS LATCHES SUBR

```

```

18446
18447
18448
18449 : THIS SUBR CLEARS OUT, VIA I-O UCON COMMANDS, THOSE STATUS LATCHES
18450 : CONCERNED WITH SERVICE CONDITIONS, UNIBUS ERROR CONDITIONS, ETC.
18451 :

```

```

18452
18453 7227: !(FREE)
18454 CLEAR-I-O-A:
18455 P3, BUS-INIT-UCON-(I), !DO A 10 MILLISEC UNIBUS INIT
18456 NEXT J/CLEAR-I-O-B
(7227) DCS(0.00.0.0.0.0) BM(0100..00.01..11.00..000..000...0.0.0..0..0...1.1011...0..0000.0...11.000...010.011.000)

```

```

18457
18458 7230: !(FREE)
18459 CLEAR-I-0-B:
18460 P3, CLR-JAM-ERRORS-(I), !RESET CACHE ERROR STATUS
18461 NEXT, J/CLEAR1002
(7230) DCS(0.00.0.0.0.0) BM(0100..00.00..10.00..000..000...0.0.0..0..0...1.1011...0..0000.0...11.000...010.011.001)

18462
18463 7231: !(FREE)
18464 CLEAR1002:
18465 P3, CLEAR-CONSOLE-SERVICE, !CLEAR OUT CONSOLE SRVC(1)H
18466 P3, CLR-NPR-TIMEOUT-(I), !RESET NPR/SACK TIME OUT STATUS
18467 NEXT, J/CLEAR1004
(7231) DCS(0.00.0.0.0.0) BM(0100..00.00..11.00..110..000...0.0.0..0..0...1.1011...0..0000.0...11.000...010.011.010)

18468
18469 7232: !(FREE)
18470 CLEAR1004:
18471 P3, CLEAR-CONSOLE-LED, !SETUP FOR 16. BIT ADDRESS MODE ON UNIBUS
18472 P3, CLR-PWR-FAIL-(I), !RESET POWER FAIL STATUS
18473 NEXT, J/CLEAR1005
(7232) DCS(0.00.0.0.0.0) BM(0100..00.01..00.00..010..001...0.0.0..0..0...1.1011...0..0000.0...11.000...010.011.011)

18474
18475 7233: !(FREE)
18476 CLEAR1005:
18477 P3, CLR-YELLOW-ZONE-(I), !RESET YELLOW ZONE STATUS
18478 NEXT, J/RESEUONCP
(7233) DCS(0.00.0.0.0.0) BM(0100..00.01..01.00..000..000...0.0.0..0..0...1.1011...0..0000.0...11.000...010.111.001)

```

```

18479 !"RESEUONCP" IS AT END OF "DINTOIR" SUBROUTINE
18480 !THIS WORD (1) ENABLES BUSDIN+EMIT, (2) ENABLES PROC EN-CLK-IR,
18481 !AND EXITS WITH A BUTA(RETURN)
18482
18483
18484
18485
18486
18487

```

```

18488 !.PAGE=====
18489
18490

```

```

18491 .TOC * SUBR FOR PUTTING SELECTED PORTIONS OF D(15-00) INTO IR
18492
18493 !
18494 THESE SUBR(S) ARE USED IN TESTING THE ALU/LOGIC FUNCTIONS.
18495
18496 NOTE: ENTRY POINTS "D(15-12)", "D(11-06)", "D(05-00)" EXPECT THAT:
18497 (1) 'IR+DBUF', 'DBUF+D' FUNCTIONS ARE ALREADY ENABLED
18498 (2) THE FOLLOWING CONSTANTS ARE IN CSP(17:14):
18499 CSP(17) = (007700); CSP(16) = (170000);
18500 CSP(15) = (000077); CSP(14) = (000100);
18501
18502 FURTHERMORE:
18503 (1) ENTRY POINT "DZERO" SETS UP THE UCONS:
18504 'IR+DBUF', 'DBUF+D'
18505 (2) ENTRY POINT "D(15-12)" COPIES THE ORIGINAL PATTERN, LEFT IN
18506 D, INTO ASPHI(17) FOR SAFEKEEPING, AND REUSE LATER

```



KD11-K MICRO V00A-1 00:00:03 12-MAR-77

```

18507 !
18508
18509
18510 7234: !(FREE)
18511 D[15-12]:
18512 P2-U, IR+DBUF, !JUST HAPPENS, DONT CARE NOW
18513 P3, DBUF+D, !PATTERN IN D -> DBUF
18514 A#BSPHI[17]+D, !SAVE IN ASPHI, TOO
18515 NEXT, J/D1512A
(7234) DCS[0.00.0.0.0.0] BM[0000..00.11..00.01..011..000...0.0.0..0..0...1.1010...0..1011.0...11.000...010.011.101]

18516 7235: !(FREE)
18517 D1512A:
18518 P2-U, IR+DBUF, !SEND DBUF -> IR
18519 P3, DBUF+D, !JUST HAPPENS, DONT CARE NOW
18520 NEXT, J/BUTIR15-12 !IR<15:12>H IN BIT<03:00> "BUT"
18521 (7235) DCS[0.00.0.0.0.0] BM[0000..00.00..00.00..000..000...0.0.0..0..0...1.1010...0..0000.0...11.000...011.000.000]

18522 7236: !(FREE)
18523 D[11-06]:
18524 P2-T, SR+ASPHI[17]-AND-007700, !MASK ALL BITS EXCEPT D<11:06> TO ZEROES
18525 NEXT, J/D1106A
(7236) DCS[0.00.0.0.0.0] BM[1011..11.00..11.01..011..000...0.0.1..0..0...0.0000...0..0000.0...11.000...010.011.111]

18527 7237: !(FREE)
18528 D1106A:
18529 P2-T, D+SR-IOR-170000, SAVE-D(C), !FORCE BITS D<15:12> TO ONES
18530 !THIS MANUEVER SHOULD FORCE INSTRS DIAGNOSTIC (E78)
18531 !ROM ADDRESSES (752), (725) WHEN PATTERN GOES INTO IR
18532 P2-U, IR+DBUF, !JUST HAPPENS, DONT CARE NOW
18533 P3, DBUF+D, !PUT PATTERN JUST SET IN D INTO DBUF
18534 NEXT, J/D10IRB
(7237) DCS[0.00.0.0.0.0] BM[1110..11.01..00.00..000..111...0.1.0..0..0...1.1010...0..0000.0...11.000...010.100.011]

18536 7240: !(FREE)
18537 D[05-00]:
18538 P2-T, SR+ASPHI[17]-AND-000077, !MASK ALL BITS EXCEPT D<05:00> TO ZEROES
18539 NEXT, J/D0500A
(7240) DCS[0.00.0.0.0.0] BM[1011..11.10..11.01..011..000...0.0.1..0..0...0.0000...0..0000.0...11.000...010.100.001]

18541 7241: !(FREE)
18542 D0500A:
18543 P2-T, D+SR-IOR-000100, SAVE-D(C), !FORCE BIT D<06> TO A ONE
18544 !THIS MANUEVER SHOULD FORCE INSTRS DIAGNOSTIC (E88)
18545 !ROM ADDRESSES (152), (125) WHEN PATTERN GOES INTO IR
18546 P2-U, IR+DBUF, !JUST HAPPENS, DONT CARE NOW
18547 P3, DBUF+D, !PUT PATTERN JUST SET IN D INTO DBUF
18548 NEXT, J/D10IRB
(7241) DCS[0.00.0.0.0.0] BM[1110..11.11..00.00..000..111...0.1.0..0..0...1.1010...0..0000.0...11.000...010.100.011]

18550 7242: !(FREE)
18551 DZERO:
18552 P2-U, IR+DBUF-[I], !JUST HAPPENS, DONT CARE
18553 P3, DBUF+D-[I], !COPY PATTERN IN D TO DBUF
18554

```

```

18555      NEXT      J/DTOIRB
(7242) DCS(0.00.0.0.0.0) BM(0100..00.00..00.01..000..100!..0.0.0..0..0...1.1011...0..0000.0...11.000...010.100.011)
18556
18557      7243:  !(FREE)
18558      DTOIRB:
18559      P2-U,      IR+DBUF,      !NOW COPY PATTERN IN DBUF TO IR
18560      P3,         DBUF+D,      !JUST HAPPENS, DONT CARE
18561      NEXT      J/BUTINSTRS    !GO DO INSTRS "BUT" ON PATTERN IN IR
(7243) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...1.1010...0..0000.0...11.000...011.000.001)
18562
18563
18564
18565
18566
18567      !.PAGE=====
18568
18569
18570      .TOC *      UCON SUBROUTINES (FLAGS, PS, FPS, CUA, SERVICE, JAM, PBA)
18571
18572      !
18573      THESE SUBROUTINES MANIPULATE THE PROCESSOR UCONS, AND VARIOUS OTHER BUSDIN DRIVERS
18574
18575      ASPLO(17) = A TEMPORARY LOCATION, DESTROYED
18576
18577      CSP(02) = VALUE EXPECTED TO BE READ
18578      CSP(03) = ACTUAL VALUE READ
18579      CSP(04) = MASK VALUE
18580
18581      AT RETURN, D = (BUSDINSELECT.AND.MASKVALUE).XOR.EXPECTEDVALUE
18582      !
18583
18584      7244:  !(FREE)
18585      CLRSERVICE00:
18586      P3,         CSPD(02)+EMIT, EMIT/000340,      !SERVICE PORT OF STATUS:  ALL BITS RESET
18587      NEXT      J/CJESERVICE00
(7244) DCS(0.00.0.0.0.0) BM(0000..10.00..00.11..100..000...0.0.0..0..0...0.1101...1..0000.0...11.000...010.100.111)
18588
18589      7245:  !(FREE)
18590      DATISERVICE00:
18591      P3,         CSPD(02)+EMIT, EMIT/100340,      !SERVICE PORT OF STATUS:  ONLY DATI(1)H SET
18592      NEXT      J/CJESERVICE00
(7245) DCS(0.00.0.0.0.0) BM(1000..10.00..00.11..100..000...0.0.0..0..0...0.1101...1..0000.0...11.000...010.100.111)
18593
18594      7246:  !(FREE)
18595      DATOSERVICE00:
18596      P3,         CSPD(02)+EMIT, EMIT/002340,      !SERVICE PORT OF STATUS:  ONLY DATO(1)H SET
18597      NEXT      J/CJESERVICE00
(7246) DCS(0.00.0.0.0.0) BM(0000..10.01..00.11..100..000...0.0.0..0..0...0.1101...1..0000.0...11.000...010.100.111)
18598
18599      7247:  !(FREE)
18600      CJESERVICE00:
18601      P3,         CLR-JAM-ERRORS-(I),      !SPECIAL ENTRY POINT TO CLEAR OUT JAM
18602      NEXT      J/SERVICE00           ! PORT OF STATUS MUX, BEFORE READ SERVICE
(7247) DCS(0.00.0.0.0.0) BM(0100..00.00..10.00..000..000...0.0.0..0..0...1.1011...0..0000.0...11.000...010.101.000)

```

```

18603
18604 7250: !(FREE)
18605 SERVICE00:
18606 PO, BUSDIN+SERVICE-[I], !SERVICE (PORT 1) OF STATUS MUX
18607 NEXT J/GETPROCDAT
(7250) DCS(0.00.0.0.0.0) BM(0100..01.00..00.00..000..000...0.0.0..0..0...1.1001...0..0000.0...11.000...011.111.011)
! - - - - -
18608
18609
18610
18611 7251: !(FREE)
18612 PBA00:
18613 PO, BUSDIN+PBA-[I], !PBA (PORT 3) OF STATUS MUX
18614 NEXT J/GETPROCDAT
(7251) DCS(0.00.0.0.0.0) BM(1100..01.00..00.00..000..000...0.0.0..0..0...1.1001...0..0000.0...11.000...011.111.011)
! - - - - -
18615
18616
18617
18618 7252: !(FREE)
18619 FLAGFPST00:
18620 PO, BUSDIN+FLAGS#FPS-[I], !PUT FLAGS, FPS ON BUSDIN
18621 NEXT J/GETPROCDAT
(7252) DCS(0.00.0.0.0.0) BM(0000..00.00..11.01..000..000...0.0.0..0..0...1.1001...0..0000.0...11.000...011.111.011)
! - - - - -
18622
18623
18624
18625 7253: !(FREE)
18626 PST00:
18627 PO, BUSDIN+PS-[I], !PUT PS ON BUSDIN
18628 NEXT J/GETPROCDAT
(7253) DCS(0.00.0.0.0.0) BM(0000..00.00..10.01..000..000...0.0.0..0..0...1.1001...0..0000.0...11.000...011.111.011)
! - - - - -
18629
18630
18631
18632 7254: !(FREE)
18633 ODDJAMT00:
18634 P3, CSPD(02)+EMIT, EMIT/101004, !JAM PORT OF STATUS: ONLY ODD-ADDR[1]H SET
18635 NEXT J/JAMT00
(7254) DCS(0.00.0.0.0.0) BM(1000..10.00..10.00..000..100...0.0.0..0..0...0.1101...1..0000.0...11.000...010.101.110)
! - - - - -
18636
18637 7255: !(FREE)
18638 CLRJAMT00:
18639 P3, CSPD(02)+EMIT, EMIT/001000, !JAM PORT OF STATUS: ALL BITS RESET
18640 NEXT J/JAMT00
(7255) DCS(0.00.0.0.0.0) BM(0000..10.00..10.00..000..000...0.0.0..0..0...0.1101...1..0000.0...11.000...010.101.110)
! - - - - -
18641
18642 7256: !(FREE)
18643 JAMT00:
18644 PO, BUSDIN+JAM-[I], !PUT JAM REG (STATUS MUX PORT 2)
18645 NEXT J/GETPROCDAT !ONTO BUSDIN
(7256) DCS(0.00.0.0.0.0) BM(1100..00.00..00.00..000..000...0.0.0..0..0...1.1001...0..0000.0...11.000...011.111.011)
! - - - - -
18646
18647
18648
18649 7257: !(FREE)

```

```

18650 CUA 00:
18651 PO, BUSDIN+CUA-[I], !PUT CUA REG (HBMUX PORT 2)
18652 NEXT J/GETPROCDAT !ONTO BUSDIN
(7257) DCS(0.00.0.0.0.0) BM(0000..00.00..01.01..000..000...0.0.0..0..0...1.1001...0..0000.0...11.000...011.111.011)
18653 ! - - - - -
18654 !
18655 7373:
18656 GETPROCDAT:
18657 P3, CSPD(03)+BUSDIN, !GET PREVIOUSLY ENABLED PROC DATA
18658 NEXT J/GETMSKPROCDAT
(7373) DCS(0.00.0.0.0.0) BM(0000..10.00..00.00..000..000...0.0.0..0..0...0.1100...1..0000.0...11.000...010.110.000)
18660 7260: !(FREE)
18661 GETMSKPROCDAT:
18662 P2-T, D+CSPD(004), D(C)+0, !GET MASK VALUE
18663 P3, ASPL0(17)+0, !INTO A-SIDE
18664 NEXT J/MSKPROCDAT
(7260) DCS(0.00.0.0.0.0) BM(1010..10.00..00.01..011..000...0.1.0..0..0...0.1011...0..0001.0...11.000...010.110.001)
18666 7261: !(FREE)
18667 MSKPROCDAT:
18668 P2-T, D+ASPL0(17)-AND-CSPD(003), SAVE-D(C), !MASK OUT UNWANTED BITS
18669 P3, ASPL0(17)+0, !AND WRITE BACK
18670 NEXT J/CMPPROCDAT
(7261) DCS(0.00.0.0.0.0) BM(1011..10.00..10.01..011..111...0.1.0..0..0...0.1100...0..0001.0...11.000...010.110.010)
18672 7262: !(FREE)
18673 CMPPROCDAT:
18674 P2-T, D+ASPL0(17)-XOR-CSPD(002), SAVE-D(C), !COMPARE OBTAINED, EXPECTED BITWISE
18675 NEXT J/RESETPROCDAT
(7262) DCS(0.00.0.0.0.0) BM(0110..10.00..10.01..011..111...0.1.0..0..0...0.1101...0..0000.0...11.000...010.110.011)
18677 7263: !(FREE)
18678 RESETPROCDAT:
18679 PO, BUSDIN+EMIT-[I], !RESET PROC UCON
18680 EN-CLK-IR(15-00),
18681 NEXT J/BUTD-IS-ZERO !AND GO TEST D<15:00>
(7263) DCS(0.00.0.0.0.0) BM(0000..00.00..00.01..000..100...0.0.0..0..0...1.1001...0..0000.0...11.000...011.100.001)

```

```

18683 !.PAGE=====
18684
18685
18686
18687
18688
18689
18690
18691 .TOC * SUBR FOR LOADING FPS<3:0> [VIA BUTA(DIAGNOSE)]
18692
18693
18694
18695
18696
18697

```

THIS SUBROUTINE LOADS CSP(16)<3:0> -> FPS<3:0> VIA THE BUTA(DIAGNOSE) FUNCTION.

REQUIREMENTS FOR ENTRY:

```

18698 | (1) LOADING SR MUST BE SETUP
18699 | (2) CSP(16) SETUP WITH BITS TO LOAD
18700 | (3) CSP(00) CONTAINS RETURN MICROADDRESS IN BITS<14:03>
18701 |
18702 |
18703 | 7264: !(FREE)
18704 | LOADFPSCC:
18705 | P2-T, SR+ALL-ONES, !SET BIT<00> FOR JAMUPP EXPECTED
18706 | P3, BUTA(DIAGNOSE), !START THE XFR TO BM SEQUENCE
18707 | NEXT, J/LOADFPSCC02
(7264) DCS[0.00.0.0.0.0] BM[1111..00.00..11.01..101..000...0.0.1..0..0...0.0000...0..0000.0...11.011...010.110.101]

18708 |
18709 | 7265: !(FREE)
18710 | LOADFPSCC02:
18711 | NEXT, XFR-TO-BM[LOADNZW4] !LOAD PAGE, POINT UPF AT BM CODE
(7265) DCS[0.00.0.0.0.0] BM[0000..00.00..00.00..000..100...0.0.0..0..0...0.0000...0..0000.0...11.100...011.011.000]

18712 | !LOADNZW4: !(THIS WORD ACTUALLY COMES OUT OF BM ROMS)
18713 | P2-T, D+CSPB(16), !PUT "MD" = CSP(16) INTO 0
18714 | NEXT, J/LOADNZW5
18715 |
18716 | !LOADNZW5:
18717 | P2, FPS[CC]+0[3-0], !LOAD FPS<3:0>
18718 | P3, BUTA(DIAGNOSE), !BEGIN XFR SEQUENCE BACK TO DCS ROMS
18719 | NEXT, J/NZERO2
18720 |
18721 | !NZERO2:
18722 | NEXT, GOTO-PAGE(4),
18723 | J/XXXXXX
18724 |
18725 | ! (CONTROL NOW RETURNS TO DCS AT "JAMUPP001" WORD)
18726 | 4777:
18727 | JAMUPP001:
18728 | *** SEE FLOWS ON SUBSEQUENT PAGE ***
18729 |
18730 |
18731 |
18732 |
18733 |
18734 |
18735 | !.PAGE=====
18736 |
18737 |
18738 | .TOC * SUBR TO COPY D-REGISTER TO DBUF TO IR
18739 |
18740 |
18741 | SUBROUTINE TO COPY D-REGISTER TO DBUF TO IR
18742 | AND LEAVE PROCESSOR UCON "CLK IR" AND "BUSDIN+EMIT" ACTIVE
18743 |
18744 |
18745 |
18746 | 7266: !(FREE)
18747 | SRINTOIR:
18748 | P2-T, D+SR, SAVE-D[0], !COPY SR TO D

```

```

18749      NEXT, J/DINTOIR
(7266) DCS(0.00.0.0.0.0) BM(1111..00.00..00.00..000..111...0.1.0..0..0...0.0000...0..0000.0...11.000...010.110.111)
18750
18751      7267: !(FREE)
18752      DINTOIR:
18753          P2-U, IR+DBUF-[I],
18754          P3-U, DBUF+D-[I],
18755      NEXT, J/DBUFINTOIR
(7267) DCS(0.00.0.0.0.0) BM(0100..00.00..00.01..000..100...0.0.0..0..0...1.1011...0..0000.0...11.000...010.111.000)
18756
18757      7270: !(FREE)
18758      DBUFINTOIR:
18759          EMITC, EMIT/125200,
18760          P2-U, IR+DBUF,
18761          P3-U, DBUF+D,
18762      NEXT, J/RESETUCONP
(7270) DCS(0.00.0.0.0.0) BM(1010..00.10..10.10..000..000...0.0.0..0..0...1.1010...0..0000.0...11.000...010.111.001)
18763
18764      7271: !(FREE)
18765      RESETUCONP:
18766          SELECT, UCON-PROC,
18767          ENABLE, EN-CLK-IR[15-00],
18768          PO, BUSDIN+EMIT[15-00],
18769          SET-UCON-CONTROL,
18770      NEXT, BUTA(RETURN),
18771          J/BUTERROR7
(7271) DCS(0.00.0.0.0.0) BM(0000..00.00..00.01..000..100...0.0.0..0..0...1.1001...0..0000.0...11.111...011.111.110)
18772
18773      ! - - - - -
18774
18775
18776      7272: !(FREE)
18777      SRINTOIRS:
18778          P2-T, D+SR, SAVE-D[1],
18779      NEXT, J/DINTOIRS
(7272) DCS(0.00.0.0.0.0) BM(1111..00.00..00.00..000..111...0.1.0..0..0...0.0000...0..0000.0...11.000...010.111.011)
18780
18781      7273: !(FREE)
18782      DINTOIRS:
18783          P2-U, IR+DBUF-[I],
18784          P3-U, DBUF+D-[I],
18785      NEXT, J/DBUFINTOIRS
(7273) DCS(0.00.0.0.0.0) BM(0100..00.00..00.01..000..100...0.0.0..0..0...1.1011...0..0000.0...11.000...010.111.100)
18786
18787      7274: !(FREE)
18788      DBUFINTOIRS:
18789          EMITC, EMIT/125200,
18790          P2-U, IR+DBUF,
18791          P3-U, DBUF+D,
18792      NEXT, J/RESETUCONPS
(7274) DCS(0.00.0.0.0.0) BM(1010..00.10..10.10..000..000...0.0.0..0..0...1.1010...0..0000.0...11.000...010.111.101)
18793
18794      7275: !(FREE)
18795      RESETUCONPS:

```

! JUST HAPPENS - DON'T CARE  
! TRANSFER D -> DBUF

! JUNK ON EMIT: INSTRS/EBB/405 PATTERN  
! TRANSFER DBUF -> BUSDIN -> IR  
! JUST HAPPENS - DON'T CARE

! RESET TO PROCESSOR UCON:  
! ENABLE CLOCKING IR  
! PUT EMIT ONTO BUSDIN  
! WRITE CONTROLS  
! RETURN TO CALLER  
! FATAL ERROR IF GO HERE

! COPY SR TO D

! JUST HAPPENS - DON'T CARE  
! TRANSFER D -> DBUF

! JUNK ON EMIT: INSTRS/EBB/405 PATTERN  
! TRANSFER DBUF -> BUSDIN -> IR  
! JUST HAPPENS - DON'T CARE

```

18796 SELECT, UCON-PROC ;RESET TO PROCESSOR UCON:
18797 ENABLE, EN-CLK-IR(15-00) ; ENABLE CLOCKING IR
18798 BUSDIN+EMIT(15-00), ; PUT EMIT ONTO BUSDIN
18799 PO, SET-UCON-CONTROL, ;WRITE CONTROLS
18800 NEXT, J/BUTINSTRS ;AND EXIT TO DO INSTR-5 BUT INTO TARGET AREA
(7275) DCS(0.00.0.0.0.0) BM(0000..00.00..00.01..000..100...0.0.0..0..0...1.1001...0..0000.0...11.000...011.000.001)

```

18801  
18802  
18803  
18804  
18805  
18806

!.PAGE=====

18807  
18808  
18809  
18810  
18811  
18812  
18813  
18814  
18815  
18816  
18817  
18818  
18819  
18820

.TOC \* JAM UPP SERVICE SUBROUTINE

IF A JAMUPP TO 0777/4777 OCCURS, ACTION DEPENDS UPON SR<00> CONTENTS:

SR<00>=1 -> RESET SR TO (000000), DO A BUTA(RETURN) TO ADDRESS IN CSP(00)

SR<00>=0 -> SIGNAL ERROR, LOG STATUS, RETURN VIA "CUA" CONTENTS  
ERROR-CODE ::= INDICATES LAST TEST EXECUTED

18821  
18822  
18823  
18824  
18825  
18826  
18827

4777:

JAMUPP001:

```

P3-T, SR+0 ;SAVE CURRENT D CONTENTS IN SR
NEXT, BUTR(SR00), ;TEST SR<00> FLAG (PREV SR00 CONTENTS)
;IF SR<00>=1, EXPECTED JAM, GOTO "JAMUPP002B"
;IF SR<00>=0, NO JAM WAS EXPECTED, FORCE ERROR AT "JAMUPP003"
J/JAMUPP003

```

(4777) DCS(0.00.0.0.0.0) BM(1111..00.00..01.01..000..000...1.0.1..0..0...0.0000...0..0000.0...00.000...111.101.110)

18828  
18829  
18830  
18831  
18832

4757:

JAMUPP002B:

```

P2-T, D+CSPD(000) SAVE-D(C), ;GET RETURN ADDRESS
NEXT, J/JAMUPP002C

```

(4757) DCS(0.00.0.0.0.0) BM(1010..10.00..00.00..000..111...0.1.0..0..0...0.1111...0..0000.0...11.000...100.011.100)

18833  
18834  
18835  
18836  
18837  
18838

4434: !(FREE)

JAMUPP002C:

```

PO, RETURN+D(14-03), PAGE(7), ;PUT RETURN ADDRESS INTO REG
P2-T, D+SR, SAVE-D(C), ;RESTORE OLD D
NEXT, J/JAMUPP002D

```

(4434) DCS(0.00.0.0.0.0) BM(1111..00.00..00.00..000..111...0.1.0..0..0...0.0000...0..0000.0...11.101...010.001.011)

18839  
18840  
18841  
18842  
18843  
18844

72'3: !(FREE)

JAMUPP002D:

```

P2-T, SR+ZERO ;ZERO OUT SR FOR RETURN
NEXT, BUTA(RETURN), ;NOW RETURN
J/BUTERROR7 ;ERROR IF GO HERE

```

```

18845 (7213) DCS(0.00.0.0.0.0) BM(0011..00.00..00.00..000..000...0.0.1..0..0...0.0000...0..0000.0...11.111...011.111.110)
18846 4756:
18847 JAMUPP003:
18848 SELECT, BUSDIN+CUA-[I], !ENABLE READ JAMUPP MICROADDR
18849 NEXT, BUTA(ERROR), !WAS ERROR PREVIOUSLY SET?
18850 J/JAMUPP005 !YES/JAMUPP004, NO/JAMUPP005
(4756) DCS(0.00.0.1.0.0) BM(0000..00.00..01.01..000..000...0.0.0..0..0...1.1001...0..0000.0...11.000...111.100.111)
18851 !* ENTER HERE IF "ERROR" WAS PREVIOUSLY SET (INHIBIT LOG)
18852 4746:
18853 JAMUPP004:
18854 P3, CSPD(00)+LOG-CUA, !GET RETURN LOC <- SAVED CUA
18855 P3-T, BUTA(CUA-TRACK), !START CUA GOING AGAIN
18856 NEXT, J/JAMUPP002B !INHIBIT LOG IF PREV ERROR
(4746) DCS(0.00.0.0.0.0) BM(0000..10.00..00.00..000..000...1.0.0..0..0...0.1111...1..0000.0...11.001...111.101.111)
18858 !* ENTER HERE IF "ERROR" NOT YET SET (LOG FIRST MODE)
18859 4747:
18860 JAMUPP005:
18861 P0, DCS-CTR(CO.), !FORCE ERROR WITH PREV ERROR-CODE/ENUA, TNUA=(4747)
18862 P3, CSPD(00)+LOG-CUA, !RETURN LOC <- RETURN ADDR FROM CUA
18863 NEXT, J/JAMUPP006 !AND GO LOG REGISTERS THIS TIME
(4747) DCS(0.00.1.0.0.0) BM(1111..10.00..00.00..000..000...0.0.0..0..0...0.1111...1..0000.0...11.000...100.011.101)
18865 4435: !(FREE)
18866 JAMUPP006:
18867 SELECT, BUSDIN+SERVICE-[I], !SERVICE REGISTER
18868 NEXT, J/JAMUPP007
(4435) DCS(0.00.0.0.0.0) BM(0100..01.00..00.00..000..000...0.0.0..0..0...1.1001...0..0000.0...11.000...100.011.110)
18870 4436: !(FREE)
18871 JAMUPP007:
18872 P3, CSPD(01)+LOG-SERVICE, !LOG SERVICE INFO REGISTER
18873 NEXT, J/JAMUPP010
(4436) DCS(0.00.0.0.0.0) BM(0000..10.00..00.00..000..000...0.0.0..0..0...0.1110...1..0000.0...11.000...100.011.111)
18875 4437: !(FREE)
18876 JAMUPP010:
18877 SELECT, BUSDIN+JAM-[I], !JAM REGISTER
18878 NEXT, J/JAMUPP011
(4437) DCS(0.00.0.0.0.0) BM(1100..00.00..00.00..000..000...0.0.0..0..0...1.1001...0..0000.0...11.000...100.100.000)
18880 4440: !(FREE)
18881 JAMUPP011:
18882 P3, CSPD(02)+LOG-JAM, !LOG JAMUPP CAUSE INFO
18883 BUTA(CUA-TRACK), !RESET CUA TO TRACK, IN CASE ANOTHER JAM COMES
18884 NEXT, J/JAMUPP002B !GO TO TOP OF THIS PAGE, AND RETURN INLINE
(4440) DCS(0.00.0.0.0.0) BM(0000..10.00..00.00..000..000...0.0.0..0..0...0.1101...1..0000.0...11.001...111.101.111)
18886
18887
18888
18889
18890

```



18891 !.PAGE=====

18892  
18893  
18894  
18895  
18896  
18897  
18898  
18899  
18900  
18901  
18902  
18903  
18904  
18905  
18906  
18907  
18908  
18909  
18910  
18911  
18912  
18913  
18914  
18915  
18916  
18917  
18918  
  
18919  
18920  
18921  
18922  
18923  
18924  
18925  
18926  
  
18927  
18928  
18929  
18930  
18931  
18932  
18933  
18934  
  
18935  
18936  
18937  
18938  
18939  
18940  
18941

.TOC \* MICROBRANCH (BUT) TAKEOFF WORDS

```

*****
!
!                                UMWORDS: 000 + 056                                !
!
!
! FUNCTION:  THESE WORDS ARE THE INACTIVE-"BUT" (BRANCHING TYPE)
!             "TAKEOFF" OR SUBROUTINE MICROWORDS. ANY TEST WHICH
!             REQUIRES A SPECIFIC "BUT" CONDITION TO BE TESTED WILL USE
!             ONE OF THESE MICROWORDS AS A TAKEOFF POINT INTO THE "BUT
!             TARGET TABLE" (DESCRIBED NEXT), WHERE AN ENUA:TNUA
!             COMPARISON WILL HAVE BEEN PREVIOUSLY ENABLED (VIA SET-
!             TING THE DIAGNOSTIC COUNTER TO THE APPROPRIATE VALUE.
!
*****

```

!\*\*\* BUT 00 \*\*\*

```

!FULL WIDTH IS BUT(SR<3:0>)
7276: !(FREE)
BUTSR3-0:
      NEXT, BUT(SR3-0)                                !TO (400)-(417), W4
      J/ZTARGET400                                   !NO MASK
(7276) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000..0.0.0..0.0..0.0000..0..0000.0...00.000...100.000.000)

```

!\*\*\* BUT 01 \*\*\*

```

!FULL WIDTH IS BUT(IR<15:12>)
7300: !(FREE)
BUTIR15-12:
      NEXT, BUT(IR15-12),                             !TO (400)-(417), W4
      J/ZTARGET400                                   !NO MASK
(7300) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000..0.0.0..0.0..0.0000..0..0000.0...00.001...100.000.000)

```

!\*\*\* BUT 02 \*\*\*

```

!FULL WIDTH IS BUT(INSTR 5)
7301: !(FREE)
BUTINSTR5:
      NEXT, BUT(INSTR5),                             !TO (400)-(437), W5
      J/ZTARGET400                                   !NO MASK
(7301) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000..0.0.0..0.0..0.0000..0..0000.0...00.010...100.000.000)

```

!\*\*\* BUT 03 \*\*\*

```

!FULL WIDTH IS BUT(IR11#FLTPT<3:0>)
7302: !(FREE)
BUTIR11#FLTPT3-0:
      NEXT, BUT(IR11#FLTPT3-0),                     !TO (400)-(437), W5

```

18942 J/ZTARGET400 !NO MASK  
 (7302) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...00.011...100.000.000)

18943  
 18944  
 18945 !\*\*\* BUT 04 \*\*\*  
 18946 !FULL WIDTH IS BUT(IR<9:6>)  
 18947 7304: !(FREE)  
 18948 BUTIR9-6:  
 18949 NEXT, BUT(IR9-6), !TO (400)-(417), W4  
 18950 J/ZTARGET400 !NO MASK  
 (7304) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...00.100...100.000.000)

18951  
 18952  
 18953 !\*\*\* BUT 05 \*\*\*  
 18954 !FULL WIDTH IS BUT(MOV-DR7#IR<5:3>)  
 18955 7305: !(FREE)  
 18956 BUTMOVDR7IRS-3:  
 18957 NEXT, BUT(MOV-DR7#IRS-3), !TO (400)-(417), W4  
 18958 J/ZTARGET400 !NO MASK  
 (7305) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...00.101...100.000.000)

18959  
 18960  
 18961 !\*\*\* BUT 06 \*\*\*  
 18962 !FULL WIDTH IS BUT(INSTR 1) \*\*\* N.B.: THIS BUT IS ALSO ACTIVE \*\*\*  
 18963 7306: !(FREE)  
 18964 BUTINSTR1:  
 18965 NEXT, BUTA(INSTR1), !TO (400)-(777), W8  
 18966 J/ZTARGET400 !NO MASK  
 (7306) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...00.110...100.000.000)

18967  
 18968  
 18969 !\*\*\* BUT 07 \*\*\*  
 18970 !FULL WIDTH IS BUT(BG-SERV-H+FP-SERV-H#D(C)#FPRET<1:0>)  
 18971 7307: !(FREE)  
 18972 BUTBGFPSERV:  
 18973 NEXT, BUT(BGSERV-FPSERV#D(C)#FPRET), !TO (407),(417), W1  
 18974 J/ZTARGET407 !MASK OUT D(C), FPRET<1:0>  
 (7307) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...00.111...100.000.111)

18975  
 18976 7310: !(FREE)  
 18977 BUTD(C)C:  
 18978 NEXT, BUT(BGSERV-FPSERV#D(C)#FPRET), !TO (413),(417), W1  
 18979 J/ZTARGET413 !MASK OUT BG-SERV-H+FP-SERV-H, FPRET<1:0>  
 (7310) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...00.111...100.001.011)

18980  
 18981  
 18982 !\*\*\* BUT10 \*\*\*  
 18983 !FULL WIDTH IS BUT(COUT07#DOUT07#FPS05)  
 18984 7311: !(FREE)  
 18985 BUTCOUT7DOUT7:  
 18986 NEXT, BUT(COUT07#DOUT07#FPS05), !TO (401),(403),(405),(407), W2  
 18987 J/ZTARGET401 !MASK OUT FPS05  
 (7311) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...01.000...100.000.001)

18988

18989 7312: !(FREE)  
 18990 BUTFPSOS:  
 18991 NEXT, BUT(COUT07#DOUT07#FPSOS), !TO (406),(407), W1  
 18992 J/ZTARGET406 !MASK OUT COUT7, DOUT7  
 (7312) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...01.000...100.000.110)

18993  
 18994  
 18995 !\*\*\* BUT 11 \*\*\*  
 18996 !FULL WIDTH IS BUT(DMO#SMO#BYTE)  
 18997 7313: !(FREE)  
 18998 BUTDMO#BYTE:  
 18999 NEXT, BUT(DMO#SMO#BYTE), !TO (400)-(407), W3  
 19000 J/ZTARGET400 !NO MASK  
 (7313) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...01.001...100.000.000)

19001  
 19002  
 19003 !\*\*\* BUT 12 \*\*\*  
 19004 !FULL WIDTH IS BUT(GD<3:2>)  
 19005 7314: !(FREE)  
 19006 BUTGD3-2:  
 19007 NEXT, BUT(GD3-2) !TO (400)-(403), W2  
 19008 J/ZTARGET400 !NO MASK  
 (7314) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...01.010...100.000.000)

19009  
 19010  
 19011 !\*\*\* BUT 13 \*\*\*  
 19012 !FULL WIDTH IS BUT(SR<1:0>#COUNT-IS-377) \*\*\* N.B.: THIS BUT IS ALSO ACTIVE \*\*\*  
 19013 7315: !(FREE)  
 19014 BUTSR1-0:  
 19015 NEXT, BUTA(SR1-0#COUNT-IS-377), !TO (401),(403),(405),(407), W2  
 19016 J/ZTARGET401 !MASK OUT COUNT-IS-377  
 (7315) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...01.011...100.000.001)

19017  
 19018  
 19019 !\*\*\* BUT 14 \*\*\*  
 19020 !FULL WIDTH IS BUT(BG-SERVCE-L#MFSS#MULTIPLE)  
 19021 7316: !(FREE)  
 19022 BUTBGSERVL:  
 19023 NEXT, BUT(BG-SERVCE-L#MFSS#MULTIPLE), !TO (403),(407), W1  
 19024 J/ZTARGET403 !MASK OUT MFSS, MULTIPLE  
 (7316) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...01.100...100.000.011)

19025  
 19026 7317: !(FREE)  
 19027 BUTHMASKPS(T):  
 19028 SETUP, TEST(MASKED-PS(T)), !SELECT MULTIPLE BUT  
 19029 NEXT, J/BUTHMXT000 !NEXT  
 (7317) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.000...011.010.000)

19030  
 19031 7320: !(FREE)  
 19032 BUTHMXTO00:  
 19033 SETUP, TEST(MASKED-PS(T)), !SELECT MULTIPLE BUT  
 19034 NEXT, BUT(BG-SERVCE-L#MFSS#MULTIPLE), !TO (406),(407), W1  
 19035 J/ZTARGET406 !MASK OUT BG-SERVCE-L, MFSS  
 (7320) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...01.100...100.000.110)

```

19036
19037 7321: !(FREE)
19038 BUTM000:
19039     SETUP, TEST(D00),           !SELECT MULTIPLE BUT
19040     NEXT,  J/BUTMXT001         !NEXT
(7321) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..001...0.0.0..0..0...0.0000...0..0000.0...11.000...011.010.010)

19041
19042 7322: !(FREE)
19043 BUTMXT001:
19044     SETUP, TEST(D00),           !SELECT MULTIPLE BUT
19045     NEXT,  BUT(BG-SERVICE-L#MFSS#MULTIPLE), !TO (406),(407) W1
19046     J/ZTARGET406              !MASK OUT BG-SERVICE-L MFSS
(7322) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..001...0.0.0..0..0...0.0000...0..0000.0...01.100...100.000.110)

19047
19048 7323: !(FREE)
19049 BUTMPS(N):
19050     SETUP, TEST(PS(N)),         !SELECT MULTIPLE BUT
19051     NEXT,  J/BUTMXT002         !NEXT
(7323) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..010...0.0.0..0..0...0.0000...0..0000.0...11.000...011.010.100)

19052
19053 7324: !(FREE)
19054 BUTMXT002:
19055     SETUP, TEST(PS(N)),         !SELECT MULTIPLE BUT
19056     NEXT,  BUT(BG-SERVICE-L#MFSS#MULTIPLE), !TO (406),(407) W1
19057     J/ZTARGET406              !MASK OUT BG-SERVICE-L MFSS
(7324) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..010...0.0.0..0..0...0.0000...0..0000.0...01.100...100.000.110)

19058
19059 7325: !(FREE)
19060 BUTMFLAG7:
19061     SETUP, TEST(FLAG7),        !SELECT MULTIPLE BUT
19062     NEXT,  J/BUTMXT003         !NEXT
(7325) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..011...0.0.0..0..0...0.0000...0..0000.7...11.000...011.010.110)

19063
19064 7326: !(FREE)
19065 BUTMXT003:
19066     SETUP, TEST(FLAG7),        !SELECT MULTIPLE BUT
19067     NEXT,  BUT(BG-SERVICE-L#MFSS#MULTIPLE), !TO (406),(407) W1
19068     J/ZTARGET406              !MASK OUT BG-SERVICE-L MFSS
(7326) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..011...0.0.0..0..0...0.0000...0..0000.0...01.100...100.000.110)

19069
19070 7327: !(FREE)
19071 BUTMEXFLAG1:
19072     SETUP, TEST(EXFLAG1),      !SELECT MULTIPLE BUT
19073     NEXT,  J/BUTMXT004         !NEXT
(7327) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..100...0.0.0..0..0...0.0000...0..0000.0...11.000...011.011.000)

19074
19075 7330: !(FREE)
19076 BUTMXT004:
19077     SETUP, TEST(EXFLAG1),      !SELECT MULTIPLE BUT
19078     NEXT,  BUT(BG-SERVICE-L#MFSS#MULTIPLE), !TO (406),(407) W1
19079     J/ZTARGET406              !MASK OUT BG-SERVICE-L MFSS
(7330) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..100...0.0.0..0..0...0.0000...0..0000.0...01.100...100.000.110)

19080
19081 7331: !(FREE)

```

19082 BUTHFLTPTS:  
 19083 SETUP, TEST(FLTPTS), !SELECT MULTIPLE BUT  
 19084 NEXT, J/BUTMNXTOOS !NEXT  
 (7331) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..101...0.0.0..0..0...0.0000...0..0000.0...11.000...011.011.010)

19085 7332: !(FREE)  
 19086 BUTMNXTOOS:  
 19087 SETUP, TEST(FLTPTS), !SELECT MULTIPLE BUT  
 19088 NEXT, BUT(BG-SERVICE-L#MFSS#MULTIPLE), !TO (406),(407), W1  
 19089 J/ZTARGET406 !MASK OUT BG-SERVICE-L, MFSS  
 19090 (7332) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..101...0.0.0..0..0...0.0000...0..0000.0...01.100...100.000.110)

19091 7333: !(FREE)  
 19092 BUTMEXFLAG2:  
 19093 SETUP, TEST(EXFLAG2), !SELECT MULTIPLE BUT  
 19094 NEXT, J/BUTMNXTOOS !NEXT  
 (7333) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..110...0.0.0..0..0...0.0000...0..0000.0...11.000...011.011.100)

19096 7334: !(FREE)  
 19097 BUTMNXTOOS:  
 19098 SETUP, TEST(EXFLAG2), !SELECT MULTIPLE BUT  
 19099 NEXT, BUT(BG-SERVICE-L#MFSS#MULTIPLE), !TO (406),(407), W1  
 19100 J/ZTARGET406 !MASK OUT BG-SERVICE-L, MFSS  
 19101 (7334) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..110...0.0.0..0..0...0.0000...0..0000.0...01.100...100.000.110)

19102 7336: !(FREE)  
 19103 BUTMINITJAM:  
 19104 SETUP, TEST(INIT-JAM), !SELECT MULTIPLE BUT  
 19105 NEXT, J/BUTMNXTOOS !NEXT  
 (7336) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..111...0.0.0..0..0...0.0000...0..0000.0...11.000...011.100.000)

19108 7340: !(FREE)  
 19109 BUTMNXTOOS:  
 19110 SETUP, TEST(INIT-JAM), !SELECT MULTIPLE BUT  
 19111 NEXT, BUT(BG-SERVICE-L#MFSS#MULTIPLE), !TO (406),(407), W1  
 19112 J/ZTARGET406 !MASK OUT BG-SERVICE-L, MFSS  
 19113 (7340) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..111...0.0.0..0..0...0.0000...0..0000.0...01.100...100.000.110)

19114 !\*\*\* BUT 15 \*\*\*  
 19115 !FULL WIDTH IS BUT(D<14:00>=0#015)  
 19116 7341: !(FREE)  
 19117 BUTD-15-ZERO:  
 19118 NEXT, BUT(D14-00E00#015), !TO (400)-(403), W2  
 19119 J/ZTARGET400 !NO MASK  
 19120 (7341) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...01.101...100.000.000)

19121 !\*\*\* BUT 16 \*\*\*  
 19122 !FULL WIDTH IS BUT(IR11#PS15)  
 19123 7342: !(FREE)  
 19124 BUTIR11B:  
 19125 NEXT, BUT(IR11#PS15), !TO (401),(403), W1  
 19126 J/ZTARGET401 !MASK OUT PS15  
 19127  
 19128

```

19129 (7342) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...01.110...100.000.001)
19130 7343: !(FREE)
19131 BUTPS15:
19132 NEXT, BUT(IR11#PS15), !TO (402),(403), W1
19133 J/ZTARGET402 !MASK OUT IR11
(7343) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...01.110...100.000.010)
19134
19135 !*** BUT 17 ***
19136 !FULL WIDTH IS BUT(COUNT-IS-377#D(C)) *** N.B.: THIS BUT IS ALSO ACTIVE ***
19137 7344: !(FREE)
19138 BUTD(C)A:
19139 NEXT, BUTA(COUNT-IS-377#D(C)), !TO (402),(403), W1
19140 J/ZTARGET402 !MASK OUT COUNT-IS-377
19141 (7344) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...01.111...100.000.010)
19142
19143 !*** BUT 20 ***
19144 !FULL WIDTH IS BUT(PREFETCH-L#SERVICE) *** N.B.: THIS BUT IS ALSO ACTIVE ***
19145 7345: !(FREE)
19146 BUTSERVICE:
19147 NEXT, BUTA(PREFETCH-L#SERVICE), !TO (402),(403), W1
19148 J/ZTARGET402 !MASK OUT PREFETCH-L
19149 (7345) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...10.000...100.000.010)
19150
19151 !*** BUT 21 ***
19152 !FULL WIDTH IS BUT(VECTOR-LOAD#DR6/7L)
19153 7346: !(FREE)
19154 BUTVECTLOAD:
19155 NEXT, BUT(VECTOR-LOAD#DR6-7L), !TO (401),(403), W1
19156 J/ZTARGET401 !MASK OUT DR6/7-L
19157 (7346) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...10.001...100.000.001)
19158
19159 7347: !(FREE)
19160 BUTDR6-7L:
19161 NEXT, BUT(VECTOR-LOAD#DR6-7L), !TO (402),(403), W1
19162 J/ZTARGET402 !MASK OUT VECTOR-LOAD
(7347) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...10.001...100.000.010)
19163
19164 !*** BUT 22 ***
19165 !THIS IS AN ACTIVE BUT - NO BRANCH MODIFICATION
19166 !(THIS BUT IS NOT USED WITH THE TARGET TABLE)
19167
19168
19169 !*** BUT 23 ***
19170 !FULL WIDTH IS BUT(D(C)#BA00)
19171 7350: !(FREE)
19172 BUTD(C)B:
19173 NEXT, BUT(D(C)#BA00), !TO (401),(403), W1
19174 J/ZTARGET401 !MASK OUT BA00
19175 (7350) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...10.011...100.000.001)

```

19176  
 19177 7351: !(FREE)  
 19178 BUTBA00:  
 19179 NEXT, BUT(D(C)BA00), !TO (402),(403), W1  
 19180 J/ZTARGET402 !MASK OUT D(C)  
 (7351) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..0000..0000...0.0.0..0..0...0.0000...0..0000.0...10.011...100.000.010)

19181  
 19182  
 19183 !\*\*\* BUT 24 \*\*\*  
 19184 !FULL WIDTH IS BUT(OTHER-JAM#FP-PROC)  
 19185 7352: !(FREE)  
 19186 BUTOTHERJAM:  
 19187 NEXT, BUT(OTHER-JAM#FP-PROC), !TO (401),(403), W1  
 19188 J/ZTARGET401 !MASK OUT FLTPT-PROC-H  
 (7352) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..0000..0000...0.0.0..0..0...0.0000...0..0000.0...10.100...100.000.001)

19189  
 19190 7354: !(FREE)  
 19191 BUTFPPROC:  
 19192 NEXT, BUT(OTHER-JAM#FP-PROC), !TO (402),(403), W1  
 19193 J/ZTARGET402 !MASK OUT OTHER-JAM  
 (7354) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..0000..0000...0.0.0..0..0...0.0000...0..0000.0...10.100...100.000.010)

19194  
 19195  
 19196 !\*\*\* BUT 25 \*\*\*  
 19197 !FULL WIDTH IS BUT(COUNT-IS-377) \*\*\* N.B.: THIS BUT IS ALSO ACTIVE \*\*\*  
 19198 !(THIS BUT IS NOT USED WITH THE TARGET TABLE)  
 19199

19200  
 19201 !\*\*\* BUT 26 \*\*\*  
 19202 !FULL WIDTH IS BUT(INTR-HIGH#INSTR-BRANCH-L)  
 19203 7355: !(FREE)  
 19204 BUTINTRHIGH:  
 19205 NEXT, BUT(INTR-HIGH#INSTR-BRANCH-L), !TO (401),(403), W1  
 19206 J/ZTARGET401 !MASK OUT INSTR-BRANCH-L  
 (7355) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..0000..0000...0.0.0..0..0...0.0000...0..0000.0...10.110...100.000.001)

19207  
 19208 7356: !(FREE)  
 19209 BUTINSTRBRANCH:  
 19210 NEXT, BUT(INTR-HIGH#INSTR-BRANCH-L), !TO (402),(403), W1  
 19211 J/ZTARGET402 !MASK OUT INTR-HIGH-H  
 (7356) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..0000..0000...0.0.0..0..0...0.0000...0..0000.0...10.110...100.000.010)

19212  
 19213  
 19214 !\*\*\* BUT 27 \*\*\*  
 19215 !FULL WIDTH IS BUT(PREFETCH-JAM#FP-FD)  
 19216 7360: !(FREE)  
 19217 BUTPREFETCHJAM:  
 19218 NEXT, BUT(PREFETCH-JAM#FP-FD), !TO (401),(403), W1  
 19219 J/ZTARGET401 !MASK OUT FLTPT-FD-H  
 (7360) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..0000..0000...0.0.0..0..0...0.0000...0..0000.0...10.111...100.000.001)

19220  
 19221 7371: !(FREE)  
 19222 BUTFPPFD:  
 19223 NEXT, BUT(PREFETCH-JAM#FP-FD), !TO (402),(403), W1

19224 J/ZTARGET402 !MASK OUT PREFETCH-JAM-H  
 (7371) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...10.111...100.000.010)

19225  
19226  
19227  
19228  
19229  
19230  
19231  
19232  
19233  
19234  
19235  
19236  
19237  
19238  
19239  
19240  
19241  
19242  
19243  
19244  
19245  
19246  
19247  
19248  
19249  
19250  
19251  
19252  
19253  
19254  
19255  
19256  
19257  
19258  
19259  
19260  
19261  
19262  
19263  
19264  
19265  
19266  
19267  
19268

!.PAGE=====

.TOC \* MICROBRANCH [BUT] TARGET WORDS

```

*****
*
*                               UNWORDS: 000 + 400
*
* FUNCTION:   TARGET BUT TABLE
*             ALL THE ABOVE BUTS TARGET INTO THIS TABLE OF MICROWORDS,
*             ALL OF WHICH DO A BUTA(RETURN).  IN THIS MANNER, ANY OF
*             THE ABOVE BRANCHES MAY BE EXECUTED, AND CONTROL WILL ALWAYS
*             RETURN TO WHERE THE "BUT TES" SUBROUTINE WAS CALLED.
*
*****

```

!\*\*\* THE TARGET BUT TABLE \*\*\*

```

7400:
ZTARGET400:  NEXT,  BUTA(RETURN), J/BUTERROR7  !COMPARE MICROADDRESS, THEN RETURN
(7400) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)

7401:
ZTARGET401:  NEXT,  BUTA(RETURN), J/BUTERROR7  !COMPARE MICROADDRESS, THEN RETURN
(7401) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)

7402:
ZTARGET402:  NEXT,  BUTA(RETURN), J/BUTERROR7  !COMPARE MICROADDRESS, THEN RETURN
(7402) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)

7403:
ZTARGET403:  NEXT,  BUTA(RETURN), J/BUTERROR7  !COMPARE MICROADDRESS, THEN RETURN
(7403) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)

7404:
ZTARGET404:  NEXT,  BUTA(RETURN), J/BUTERROR7  !COMPARE MICROADDRESS, THEN RETURN
(7404) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)

7405:
ZTARGET405:  NEXT,  BUTA(RETURN), J/BUTERROR7  !COMPARE MICROADDRESS, THEN RETURN
(7405) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)

7406:
ZTARGET406:  NEXT,  BUTA(RETURN), J/BUTERROR7  !COMPARE MICROADDRESS, THEN RETURN
(7406) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)

7407:
ZTARGET407:  NEXT,  BUTA(RETURN), J/BUTERROR7  !COMPARE MICROADDRESS, THEN RETURN
(7407) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)

7410:
ZTARGET410:  NEXT,  BUTA(RETURN), J/BUTERROR7  !COMPARE MICROADDRESS, THEN RETURN

```



```

(7410) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19269 7411:
19270 ZTARGET411: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7411) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19271 7412:
19272 ZTARGET412: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7412) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19273 7413:
19274 ZTARGET413: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7413) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19275 7414:
19276 ZTARGET414: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7414) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19277 7415:
19278 ZTARGET415: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7415) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19279 7416:
19280 ZTARGET416: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7416) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19281 7417:
19282 ZTARGET417: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7417) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19283 7420:
19284 ZTARGET420: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7420) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19285 7421:
19286 ZTARGET421: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7421) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19287 7422:
19288 ZTARGET422: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7422) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19289 7423:
19290 ZTARGET423: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7423) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19291 7424:
19292 ZTARGET424: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7424) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19293 7425:
19294 ZTARGET425: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7425) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19295 7426:
19296 ZTARGET426: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7426) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19297 7427:
19298 ZTARGET427: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7427) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19299 7430:
19300 ZTARGET430: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7430) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19301 7431:
19302 ZTARGET431: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7431) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19303 7432:
19304 ZTARGET432: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN

```

```

(7432) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19305 7433:
19306 ZTARGET433: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7433) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19307 7434:
19308 ZTARGET434: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7434) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19309 7435:
19310 ZTARGET435: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7435) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19311 7436:
19312 ZTARGET436: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7436) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19313 7437:
19314 ZTARGET437: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7437) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19315 7440:
19316 ZTARGET440: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7440) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19317 7441:
19318 ZTARGET441: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7441) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19319 7442:
19320 ZTARGET442: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7442) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19321 7443:
19322 ZTARGET443: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7443) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19323 7444:
19324 ZTARGET444: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7444) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19325 7445:
19326 ZTARGET445: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7445) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19327 7446:
19328 ZTARGET446: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7446) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19329 7447:
19330 ZTARGET447: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7447) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19331 7450:
19332 ZTARGET450: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7450) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19333 7451:
19334 ZTARGET451: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7451) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19335 7452:
19336 ZTARGET452: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7452) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19337 7453:
19338 ZTARGET453: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7453) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19339 7454:
19340 ZTARGET454: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN

```

```

(7454) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19341 7455:
19342 ZTARGET455: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7455) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19343 7456:
19344 ZTARGET456: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7456) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19345 7457:
19346 ZTARGET457: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7457) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19347 7460:
19348 ZTARGET460: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7460) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19349 7461:
19350 ZTARGET461: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7461) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19351 7462:
19352 ZTARGET462: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7462) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19353 7463:
19354 ZTARGET463: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7463) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19355 7464:
19356 ZTARGET464: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7464) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19357 7465:
19358 ZTARGET465: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7465) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19359 7466:
19360 ZTARGET466: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7466) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19361 7467:
19362 ZTARGET467: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7467) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19363 7470:
19364 ZTARGET470: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7470) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19365 7471:
19366 ZTARGET471: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7471) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19367 7472:
19368 ZTARGET472: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7472) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19369 7473:
19370 ZTARGET473: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7473) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19371 7474:
19372 ZTARGET474: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7474) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19373 7475:
19374 ZTARGET475: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7475) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19375 7476:
19376 ZTARGET476: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN

```

```

(7476) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19377 7477:
19378 ZTARGET477: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7477) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19379 7500:
19380 ZTARGET500: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7500) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19381 7501:
19382 ZTARGET501: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7501) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19383 7502:
19384 ZTARGET502: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7502) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19385 7503:
19386 ZTARGET503: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7503) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19387 7504:
19388 ZTARGET504: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7504) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19389 7505:
19390 ZTARGET505: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7505) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19391 7506:
19392 ZTARGET506: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7506) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19393 7507:
19394 ZTARGET507: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7507) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19395 7510:
19396 ZTARGET510: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7510) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19397 7511:
19398 ZTARGET511: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7511) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19399 7512:
19400 ZTARGET512: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7512) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19401 7513:
19402 ZTARGET513: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7513) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19403 7514:
19404 ZTARGET514: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7514) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19405 7515:
19406 ZTARGET515: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7515) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19407 7516:
19408 ZTARGET516: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7516) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19409 7517:
19410 ZTARGET517: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7517) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19411 7520:
19412 ZTARGET520: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN

```

```

(7520) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19413 7521:
19414 ZTARGET521: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7521) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19415 7522:
19416 ZTARGET522: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7522) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19417 7523:
19418 ZTARGET523: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7523) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19419 7524:
19420 ZTARGET524: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7524) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19421 7525:
19422 ZTARGET525: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7525) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19423 7526:
19424 ZTARGET526: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7526) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19425 7527:
19426 ZTARGET527: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7527) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19427 7530:
19428 ZTARGET530: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7530) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19429 7531:
19430 ZTARGET531: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7531) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19431 7532:
19432 ZTARGET532: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7532) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19433 7533:
19434 ZTARGET533: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7533) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19435 7534:
19436 ZTARGET534: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7534) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19437 7535:
19438 ZTARGET535: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7535) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19439 7536:
19440 ZTARGET536: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7536) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19441 7537:
19442 ZTARGET537: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7537) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19443 7540:
19444 ZTARGET540: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7540) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19445 7541:
19446 ZTARGET541: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7541) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19447 7542:
19448 ZTARGET542: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN

```

```

(7542) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19449 7543:
19450 ZTARGET543: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7543) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19451 7544:
19452 ZTARGET544: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7544) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19453 7545:
19454 ZTARGET545: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7545) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19455 7546:
19456 ZTARGET546: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7546) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19457 7547:
19458 ZTARGET547: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7547) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19459 7550:
19460 ZTARGET550: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7550) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19461 7551:
19462 ZTARGET551: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7551) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19463 7552:
19464 ZTARGET552: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7552) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19465 7553:
19466 ZTARGET553: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7553) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19467 7554:
19468 ZTARGET554: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7554) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19469 7555:
19470 ZTARGET555: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7555) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19471 7556:
19472 ZTARGET556: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7556) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19473 7557:
19474 ZTARGET557: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7557) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19475 7560:
19476 ZTARGET560: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7560) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19477 7561:
19478 ZTARGET561: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7561) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19479 7562:
19480 ZTARGET562: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7562) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19481 7563:
19482 ZTARGET563: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7563) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19483 7564:
19484 ZTARGET564: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN

```

```

(7564) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19485 7565:
19486 ZTARGET565: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7565) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19487 7566:
19488 ZTARGET566: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7566) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19489 7567:
19490 ZTARGET567: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7567) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19491 7570:
19492 ZTARGET570: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7570) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19493 7571:
19494 ZTARGET571: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7571) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19495 7572:
19496 ZTARGET572: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7572) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19497 7573:
19498 ZTARGET573: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7573) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19499 7574:
19500 ZTARGET574: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7574) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19501 7575:
19502 ZTARGET575: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7575) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19503 7576:
19504 ZTARGET576: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7576) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19505 7577:
19506 ZTARGET577: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7577) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19507 7600:
19508 ZTARGET600: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7600) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19509 7601:
19510 ZTARGET601: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7601) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19511 7602:
19512 ZTARGET602: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7602) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19513 7603:
19514 ZTARGET603: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7603) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19515 7604:
19516 ZTARGET604: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7604) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19517 7605:
19518 ZTARGET605: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7605) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19519 7606:
19520 ZTARGET606: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN

```

```

(7606) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19521 7607:
19522 ZTARGET607: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7607) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19523 7610:
19524 ZTARGET610: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7610) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19525 7611:
19526 ZTARGET611: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7611) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19527 7612:
19528 ZTARGET612: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7612) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19529 7613:
19530 ZTARGET613: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7613) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19531 7614:
19532 ZTARGET614: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7614) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19533 7615:
19534 ZTARGET615: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7615) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19535 7616:
19536 ZTARGET616: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7616) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19537 7617:
19538 ZTARGET617: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7617) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19539 7620:
19540 ZTARGET620: NEXT, BUTA(RETURN), J, - JRROR7 !COMPARE MICROADDRESS, THEN RETURN
(7620) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19541 7621:
19542 ZTARGET621: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7621) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19543 7622:
19544 ZTARGET622: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7622) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19545 7623:
19546 ZTARGET623: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7623) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19547 7624:
19548 ZTARGET624: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7624) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19549 7625:
19550 ZTARGET625: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7625) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19551 7626:
19552 ZTARGET626: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7626) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19553 7627:
19554 ZTARGET627: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7627) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19555 7630:
19556 ZTARGET630: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN

```



```

(7630) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19557 7631:
19558 ZTARGET631: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7631) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19559 7632:
19560 ZTARGET632: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7632) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19561 7633:
19562 ZTARGET633: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7633) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19563 7634:
19564 ZTARGET634: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7634) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19565 7635:
19566 ZTARGET635: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7635) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19567 7636:
19568 ZTARGET636: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7636) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19569 7637:
19570 ZTARGET637: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7637) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19571 7640:
19572 ZTARGET640: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7640) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19573 7641:
19574 ZTARGET641: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7641) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19575 7642:
19576 ZTARGET642: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7642) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19577 7643:
19578 ZTARGET643: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7643) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19579 7644:
19580 ZTARGET644: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7644) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19581 7645:
19582 ZTARGET645: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7645) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19583 7646:
19584 ZTARGET646: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7646) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19585 7647:
19586 ZTARGET647: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7647) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19587 7650:
19588 ZTARGET650: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7650) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19589 7651:
19590 ZTARGET651: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7651) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19591 7652:
19592 ZTARGET652: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN

```

```

(7652) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19593 7653:
19594 ZTARGET653: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7653) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19595 7654:
19596 ZTARGET654: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7654) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19597 7655:
19598 ZTARGET655: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7655) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19599 7656:
19600 ZTARGET656: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7656) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19601 7657:
19602 ZTARGET657: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7657) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19603 7660:
19604 ZTARGET660: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7660) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19605 7661:
19606 ZTARGET661: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7661) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19607 7662:
19608 ZTARGET662: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7662) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19609 7663:
19610 ZTARGET663: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7663) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19611 7664:
19612 ZTARGET664: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7664) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19613 7665:
19614 ZTARGET665: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7665) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19615 7666:
19616 ZTARGET666: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7666) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19617 7667:
19618 ZTARGET667: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7667) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19619 7670:
19620 ZTARGET670: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7670) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19621 7671:
19622 ZTARGET671: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7671) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19623 7672:
19624 ZTARGET672: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7672) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19625 7673:
19626 ZTARGET673: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7673) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19627 7674:
19628 ZTARGET674: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN

```

```

(7674) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19629 7675:
19630 ZTARGET675: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7675) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19631 7676:
19632 ZTARGET676: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7676) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19633 7677:
19634 ZTARGET677: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7677) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19635 7700:
19636 ZTARGET700: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7700) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19637 7701:
19638 ZTARGET701: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7701) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19639 7702:
19640 ZTARGET702: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7702) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19641 7703:
19642 ZTARGET703: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7703) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19643 7704:
19644 ZTARGET704: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7704) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19645 7705:
19646 ZTARGET705: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7705) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19647 7706:
19648 ZTARGET706: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7706) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19649 7707:
19650 ZTARGET707: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7707) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19651 7710:
19652 ZTARGET710: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7710) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19653 7711:
19654 ZTARGET711: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7711) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19655 7712:
19656 ZTARGET712: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7712) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19657 7713:
19658 ZTARGET713: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7713) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19659 7714:
19660 ZTARGET714: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7714) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19661 7715:
19662 ZTARGET715: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7715) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19663 7716:
19664 ZTARGET716: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN

```

```

(7716) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19665 7717:
19666 ZTARGET717: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7717) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19667 7720:
19668 ZTARGET720: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7720) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19669 7721:
19670 ZTARGET721: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7721) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19671 7722:
19672 ZTARGET722: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7722) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19673 7723:
19674 ZTARGET723: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7723) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19675 7724:
19676 ZTARGET724: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7724) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19677 7725:
19678 ZTARGET725: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7725) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19679 7726:
19680 ZTARGET726: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7726) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19681 7727:
19682 ZTARGET727: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7727) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19683 7730:
19684 ZTARGET730: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7730) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19685 7731:
19686 ZTARGET731: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7731) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19687 7732:
19688 ZTARGET732: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7732) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19689 7733:
19690 ZTARGET733: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7733) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19691 7734:
19692 ZTARGET734: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7734) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19693 7735:
19694 ZTARGET735: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7735) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19695 7736:
19696 ZTARGET736: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7736) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19697 7737:
19698 ZTARGET737: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7737) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19699 7740:
19700 ZTARGET740: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN

```

```

(7740) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19701 7741:
19702 ZTARGET741: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7741) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19703 7742:
19704 ZTARGET742: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7742) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19705 7743:
19706 ZTARGET743: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7743) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19707 7744:
19708 ZTARGET744: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7744) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19709 7745:
19710 ZTARGET745: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7745) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19711 7746:
19712 ZTARGET746: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7746) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19713 7747:
19714 ZTARGET747: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7747) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19715 7750:
19716 ZTARGET750: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7750) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19717 7751:
19718 ZTARGET751: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7751) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19719 7752:
19720 ZTARGET752: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7752) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19721 7753:
19722 ZTARGET753: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7753) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19723 7754:
19724 ZTARGET754: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7754) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19725 7755:
19726 ZTARGET755: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7755) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19727 7756:
19728 ZTARGET756: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7756) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19729 7757:
19730 ZTARGET757: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7757) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19731 7760:
19732 ZTARGET760: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7760) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19733 7761:
19734 ZTARGET761: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7761) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19735 7762:
19736 ZTARGET762: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN

```

```

(7762) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19737 7763:
19738 ZTARGET763: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7763) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19739 7764:
19740 ZTARGET764: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7764) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19741 7765:
19742 ZTARGET765: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7765) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19743 7766:
19744 ZTARGET766: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7766) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19745 7767:
19746 ZTARGET767: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7767) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19747 7770:
19748 ZTARGET770: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7770) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19749 7771:
19750 ZTARGET771: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7771) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19751 7772:
19752 ZTARGET772: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7772) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19753 7773:
19754 ZTARGET773: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7773) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19755 7774:
19756 ZTARGET774: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7774) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19757 7775:
19758 ZTARGET775: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7775) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19759 7776:
19760 ZTARGET776: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7776) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)
19761 7777:
19762 ZTARGET777: NEXT, BUTA(RETURN), J/BUTERROR7 !COMPARE MICROADDRESS, THEN RETURN
(7777) DCS(0.00.0.0.0.0) BM(0000..00.00..00.00..000..000...0.0.0..0..0...0.0000...0..0000.0...11.111...011.111.110)

```

```

19763
19764
19765 !*****
19766 .TOC * END OF KD11-K MICRODIAGNOSTIC CODE
19767 !*****
19768
19769 .END

```







LINE	LOCN	SYMBOL	LINE	LOCN	SYMBOL	LINE	LOCN	SYMBOL	LINE	LOCN	SYMBOL
8399	7025	AGAINSR0350	14400	7155	AL1-537A	18075	4731	ALLOW763A	5887	4003	ALU114A
5946	4610	ALU115A1	6028	4015	ALU115B1	6111	4622	ALU115C1	6171	4026	ALU115D1
6242	4634	ALU116A1	6333	4040	ALU116B	6366	4646	ALU116C1	6459	4050	ALU116D
6502	4666	ALU117A1	6577	4057	ALU117B1	6540	4063	ALU117C1	6692	4664	ALU120A1
6752	4072	ALU120B1	6839	4706	ALU121A1	6913	4107	ALU121B1	6994	4670	ALU121C1
7068	4121	ALU121D1	7155	4672	ALU122A1	8041	7007	ALUCARRY1	8050	7017	ALUCARRY1A
8058	7020	ALUCARRY2	8067	7021	ALUCARRY2A	14347	7145	AR11-537A	14298	6451	AR3-537A
14323	7141	AR7-537A	7290	4540	ARITH130A1	7341	4140	ARITH130B1	7399	4520	ARITH131A1
7450	4150	ARITH131B1	7509	4542	ARITH132A1	7554	4157	ARITH132B1	10856	6114	ASIDE410C
11048	6525	ASERTFOVS00	15612	5413	BCIFCN622A	15269	5570	BCERCE20A	15641	5416	BCERCE22B
17504	7204	BAADS	17492	7177	BCX12	14407	4242	BA1-537A	14353	7146	BR11-537A
14305	7134	BR3-537A	14329	7142	BR7-537A	17885	4551	BMP0762A	16180	4313	BSFCN710A
16342	4311	BSFCN711A	16480	4316	BSFCN712A	16608	7167	BSFCN713A	16758	7176	BSFCN720A
16863	7201	BSFCN721A	16974	4324	BSFCN722A	17101	4331	BSFCN730A	17161	4337	BSFCN730B
17257	4346	BSFCN730D	17328	4764	BSFCN731A	17370	4356	BSFCN731B	17440	4364	BSFCN731D
17563	4371	BSFCN740A	17518	4373	BSFCN740B	17627	4374	BSFCN740C	17660	4401	BSFCN740D
19178	7351	BUTB00D	18972	7307	BUTBGFPSEVR	19022	7316	BUTBGSEVR	11873	6217	BUTCLR505A
14517	7136	BUTCOUNT-15-377	18985	7311	BUTCOUT7DOUT7	19118	7341	BUTD-IS-ZERO	18998	7313	BUTDM1BYTE
19160	7347	BUTOR6-7L	19139	7344	BUTD(C)A	19173	7350	BUTD(C)B	18977	7310	BUTD(C)C
2830	4376	BUTERP0R4	2886	5376	BUTERRORS	2892	6376	BUTERROR6	2898	7376	BUTERROR7
19222	7371	BUTFPFD	19191	7354	BUTFPPROC	18990	7312	BUTFPS05	19006	7314	BUTGD3-2
18964	7306	BUTINSTRI	18932	7301	BUTINSTRS	19209	7356	BUTINSTBRANCH	19204	7355	BUTINTRHIGH
19126	7342	BUTIR11B	18940	7302	BUTIR11FL TPT3-0	18924	7300	BUTIR15-12	18948	7304	BUTIR9-6
19038	7321	BUTH00D	19071	7327	BUTHEXFLAG1	19093	7333	BUTHEXFLAG2	19060	7325	BUTHFLAG7
19082	7331	BUTHFLTPTS	19105	7336	BUTHINITJAM	19027	7317	BUTHMASKPS(T)	19032	7320	BUTHNXTD00
19043	7322	BUTHNXT001	19054	7324	BUTHNXT002	19065	7326	BUTHNXT003	19076	7330	BUTHNXT004
19087	7332	BUTHNXT005	19098	7334	BUTHNXT006	19110	7340	BUTHNXT007	18956	7305	BUTHOVDR7IRS-3
19049	7323	BUTHPS(N)	19186	7352	BUTOOTHERJAM	19217	7360	BUTPREFETCHJAM	19131	7343	BUTPS15
19147	7345	BUTSERVICE	19014	7315	BUTSR1-0	18916	7276	BUTSR3-0	19155	7346	BUTVECTLOAD
10540	7062	BYTE375A	10745	7070	BYTEFIRST410	10764	7367	BYTESECOND410	17509	7205	C17X05
17497	7203	C17X12	10547	7063	CHECK375A	10570	5300	CHECK375B	12293	6254	CINS07F
18600	7247	CJESERVICET00	18454	7227	CLEAR-I-0-A	18459	7230	CLEAR-I-0-B	12106	6236	CLEAR506E
15878	7165	CLEAR624	16278	6470	CLEAR710D	17747	4404	CLEAR761A	18464	7231	CLEAR1002
18470	7232	CLEAR1004	18476	7233	CLEAR1005	18638	7255	CLRJAMT00	18585	7244	CLRSERVICET00
18674	7262	CMPPROCDAT	15618	5414	CMST0622A	7299	4135	COMP130A1	7350	4141	COMP130B1
7408	4145	COMP131A1	7459	4151	COMP131B1	8424	4702	COMP350A	8451	4244	COMP350B
8479	4246	COMP350C	8506	4250	COMP350D	8576	4704	COMP351A	8598	4257	COMP351B
8620	4261	COMP351C	8641	4263	COMP351D	8677	4714	COMP352A	8699	4267	COMP352B
8721	4271	COMP352C	8742	4273	COMP352D	8860	6004	COMP361B	8943	6012	COMP362A
8976	6015	COMP362B	9051	6022	COMP363A	9174	6034	COMP365A	9242	6042	COMP366A
9302	6047	COMP366C	9056	6057	COMP370A	9447	6064	COMP370C	9517	6072	COMP371A
10797	6106	COMP410A	10707	6111	COMP410B	10861	6115	COMP410C	10893	6120	COMP410D
10925	6123	COMP410E	11078	6134	COMP500A	11114	6137	COMP500C	11143	6141	COMP500D
11178	6144	COMP500E	13548	6350	COMP533A	13589	6354	COMP533B	13630	6357	COMP534A
13664	6362	COMP534B	13704	6365	COMP534C	13738	6370	COMP534D	13785	6374	COMP534E
13824	6402	COMP534F	13906	6411	COMP535A	13946	6415	COMP535B	13999	6422	COMP536A
14039	6426	COMP536B	14082	6431	COMP536C	14129	6436	COMP536D	14175	6442	COMP536E
14215	6446	COMP536F	14422	4302	COMP537A	15580	5412	COMP621H	17116	4333	COMP730A
17342	4354	COMP731A	17409	4361	COMP731C	14579	4301	COUNTER01	14584	4310	COUNTER02
14592	4526	COUNTER03	14603	4545	COUNTER04	14612	4557	COUNTER05	14621	4534	COUNTER06
14631	4535	COUNTER07	14641	4536	COUNTER08	14652	4525	COUNTER09	14663	4527	COUNTER10
14670	4533	COUNTER10A	14676	4531	COUNTER10B	14683	4547	COUNTER11	14690	4555	COUNTER12

L07

LINE	LOCN	SYMBOL
13851	6404	COUNTERS34G
9574	7035	CSP16XORFLYTOIRS
9678	5555	CUA3728
18518	7235	D1512A
18788	7274	DBUFINTOIRS
18782	7273	DINTOIRS
18422	7224	DISPO04
10718	6531	DNONZER0410
12009	7072	DOIT6106A
14953	5336	DOIT61082
15163	5363	DOIT61002
7944	4212	DOPA136A1
10019	5237	DOWNRITE374B2
10165	5256	DOWNRITE37402
10311	5274	DOWNRITE374F2
18558	7243	DTOIR8
18538	7240	D105-001
18230	4431	EOP001
18006	7211	EOP005
14371	7151	ER3-537A
8953	6003	EXPEC3618
9295	6046	EXPEC376C
10133	5567	EXPEC37401
10822	6110	EXPEC4108
11108	6136	EXPEC500C
11599	6175	EXPEC507A
12151	6602	EXPEC507A
14076	6424	EXPEC5368
15608	5421	EXPEC623
16602	5437	EXPEC702A
16601	6473	EXPEC7118
16697	6504	EXPEC7130
17953	4414	FILL762C
12571	7075	FLAGFPS02
18619	7252	FLAGFPS100
16474	4314	GETHOR712A
15852	5431	GETCUA624C
8176	4227	GETDC320C
14907	5330	GETIT61081
15133	5360	GETIT61001
17376	4357	GETIT7318
15665	5420	GETJAM622C
9934	5226	GETTEM374A1
10080	5244	GETTEM374C1
10226	5253	GETTEM374E1
5894	4004	GETZERUES114A
3116	5006	GOAUTO120
3220	5013	GOAUTO13A
3299	5017	GOAUTO13E
3403	5024	GOAUTO148
3508	5031	GOAUTO158
3635	5037	GOAUTO178

LINE	LOCN	SYMBOL
14359	7147	CR11-537A
9568	7031	CSP16XORSRTOIRS
18650	7257	CUA100
18590	7245	DATISERVICET00
17958	4415	DELAY762C
8408	4563	DINTOIRA350
18428	7225	DISPO05
11202	6153	DOFCC503A
14765	5313	DOIT610A1
15016	5344	DOIT610C1
7613	4163	DOPA133A1
9926	5005	DWRITE374A1
10072	5043	DWRITE374C1
10218	5262	DWRITE374E1
14365	7150	DR11-537A
17836	4411	DWILL762A
18524	7236	D111-061
18237	6452	EOP002
18065	6776	EOP006
2127	6543	EF010
8969	6014	EXPEC3628
9914	5222	EXPEC374A1
10206	5515	EXPEC374E1
10651	6113	EXPEC410C
11171	6143	EXPEC500E
11723	6205	EXPEC507E
13893	6407	EXPEC535A
14116	6434	EXPEC5360
15846	5400	EXPEC624C
16096	6400	EXPEC702B
16411	6475	EXPEC711C
17156	4376	EXPEC7308
10506	7046	FIRST375A
12577	7076	FLAGFPS03
14322	7152	FR3-537A
5554	4010	GETALTN115A1
17109	4302	GETDBUF730A
8237	4234	GETDC320E
14062	5337	GETIT61082
15172	5364	GETIT61002
17853	7210	GETIT762A
18062	7260	GETNSKPR0CDAT
9454	5231	GETTEM374A2
10100	5250	GETTEM374C2
10246	5256	GETTEM374E2
3056	5003	GOAUTO12A
3135	5007	GOAUTO12E
3242	5014	GOAUTO138
3320	5020	GOAUTO13F
3422	5025	GOAUTO14C
3527	5032	GOAUTO15C
3677	5041	GOAUTO20A

LINE	LOCN	SYMBOL
14311	7137	CR3-537A
15352	5372	CSP1L621A
18543	7241	D0500A
18595	7246	DATOSERVICET00
18752	7267	DINTOIR
18410	7222	DISPO02
18434	7226	DISPO06
11611	6177	DOFCC504A
14835	5321	DOIT610A2
15067	5352	DOIT610C2
7725	4173	DOPA134A1
9946	5070	DWRITE374A2
10092	5047	DWRITE374C2
10238	5265	DWRITE374E2
14317	7140	CR3-537A
18752	7242	DZERO
16511	7234	D115-121
18243	6510	EOP003
18275	6512	EOP007
15426	5615	EXPEC3621A
9167	6033	EXPEC375A
9087	5537	EXPEC374B1
10279	5527	EXPEC374F1
10886	6117	EXPEC4100
11262	6150	EXPEC503A
11860	6215	EXPEC507A
13934	6413	EXPEC5358
15702	5405	EXPEC621E
15939	7166	EXPEC701A
16243	6455	EXPEC710C
16562	6501	EXPEC7120
18013	4417	EXPEC762E
11056	6527	FIXPAT500
12502	7077	FLAGFPS04
11902	6231	FULLFPP506A
15535	5410	GETCUR621F
17335	4353	GETDCUF731A
14794	5314	GETIT610A1
15025	5345	GETIT610C1
17169	4340	GETIT7308
15324	5371	GETJAM620C
5031	4600	GETONES114A
10007	5235	GETTEM374B1
10153	5254	GETTEM37401
10299	5272	GETTEM374F1
3077	5004	GOAUTO128
3155	5010	GOAUTO12F
3261	5015	GOAUTO13C
3339	5021	GOAUTO13G
3444	5026	GOAUTO14D
3570	5034	GOAUTO16A
3697	5042	GOAUTO20B

LINE	LOCN	SYMBOL
14335	7143	CR7-537A
9627	6222	CUA372A
18529	7237	D1106A
18758	7270	DBUFINTOIR
8359	7353	DINTOIR350
18416	7223	DISPO03
18404	7221	DISPLAY
10709	5440	DOIT376A
14603	5327	DOIT610B1
15124	5357	DOIT61001
7803	4003	DOPA135A1
9949	5074	DOWNRITE374B1
10145	5053	DOWNRITE37401
10291	5271	DOWNRITE374F1
14341	7144	CR7-537A
10728	6533	DZERO010
12452	6270	ENFLAGS100
18350	6511	EOP004
18382	7212	EOP010
15303	6567	ERROR624A
9236	6041	EXPEC366A
10060	5557	EXPEC374C1
10785	7363	EXPEC410A
10918	6122	EXPEC410E
11414	6163	EXPEC503E
11959	6612	EXPEC506A
13006	6420	EXPEC536A
15008	5407	EXPEC621F
15008	6455	EXPEC701C
16272	6457	EXPEC7100
16648	5402	EXPEC7138
18069	4412	EXPEC763A
12446	6267	FLAG5100
12506	7073	FLAGFPS07A
12177	6244	FULLFPP507A
15695	5402	GETCUR623
8118	4002	GETDC320A
14844	5002	GETIT610A2
15076	5353	GETIT610C2
17205	4347	GETIT7300
15008	5400	GETJAM621E
18657	7373	GETPRUCDAT
10027	5240	GETTEM374B2
10173	5257	GETTEM37402
10319	5275	GETTEM374F2
3097	5005	GOAUTO12C
3177	5011	GOAUTO12G
3280	5016	GOAUTO130
3392	5023	GOAUTO14A
3488	5030	GOAUTO15A
3613	5036	GOAUTO17A
3719	5043	GOAUTO20C

LINE	LOCN	SYMBOL	LINE	LOCN	SYMBOL	LINE	LOCN	SYMBOL	LINE	LOCN	SYMBOL
3762	5045	GOBUT021A	3783	5046	GOBUT021B	3805	5047	GOBUT021C	3848	5051	GOBUT022A
3892	5053	GOBUT023A	3936	5055	GOBUT024A	3957	5056	GOBUT024B	4001	5060	GOBUT025A
4045	5062	GOBUT026A	4089	5064	GOBUT027A	4132	5066	GOBUT030A	4155	5067	GOBUT030B
4174	5070	GOBUT030C	4197	5071	GOBUT030D	4240	5073	GOBUT031A	4261	5074	GOBUT031B
4304	5076	GOBUT032A	4325	5077	GOBUT032B	4370	5101	GOBUT033A	4414	5103	GOBUT034A
4457	5105	GOBUT035A	4502	5107	GOBUT036A	4546	5111	GOBUT037A	4589	5113	GOBUT040A
4611	5114	GOBUT040B	4654	5116	GOBUT041A	4676	5117	GOBUT041B	4718	5121	GOBUT042A
4739	5122	GOBUT042B	4783	5124	GOBUT043A	4804	5125	GOBUT043B	4848	5127	GOBUT044A
4893	5131	GOBUT045A	4935	5133	GOBUT046A	4958	5134	GOBUT046B	5004	5136	GOBUT047A
5027	5137	GOBUT047B	5047	5140	GOBUT047C	5093	5142	GOBUT050A	5122	5144	GOBUT050B
5626	5210	UT105A1	5674	5213	UT105B1	5899	4005	UT114A	5913	4006	UT114A2
5959	4011	UT115A1	5991	4013	UT115A2	6008	4014	UT115A3	5975	4012	UT115A4
6036	4016	UT115B1	6068	4018	UT115B2	6085	4021	UT115B3	6052	4017	UT115B4
6119	4023	UT115C1	6134	4024	UT115C2	6151	4025	UT115C3	6179	4027	UT115D1
6194	4030	UT115D2	6211	4031	UT115D3	6250	4033	UT116A1	6264	4034	UT116A2
6279	4035	UT116A3	6296	4036	UT116A4	6312	4037	UT116A5	6340	4041	UT116B
6374	4043	UT116C1	6389	4044	UT116C2	6404	4045	UT116C3	6420	4046	UT116C4
6436	4047	UT116C5	6466	4051	UT116D	6510	4053	UT117A1	6525	4054	UT117A2
6542	4055	UT117A3	6557	4056	UT117A4	6585	4058	UT117B1	6600	4061	UT117B2
6617	4062	UT117B3	6647	4064	UT117C1	6662	4065	UT117C2	6700	4067	UT120A1
6715	4070	UT120A2	6732	4071	UT120A3	6760	4073	UT120B1	6775	4074	UT120B2
6792	4075	UT120B3	6807	4076	UT120B4	6847	4103	UT121A1	6862	4104	UT121A2
6878	4105	UT121A3	6893	4106	UT121A4	6921	4110	UT121B1	6936	4111	UT121B2
6953	4112	UT121B3	6968	4113	UT121B4	7002	4115	UT121C1	7017	4116	UT121C2
7033	4117	UT121C3	7048	4120	UT121C4	7076	4122	UT121D1	7091	4123	UT121D2
7109	4124	UT121D3	7123	4125	UT121D4	7163	4127	UT122A1	7179	4130	UT122A2
7195	4131	UT122A3	7215	4133	UT122A4	7306	4136	UT130A1	7321	4137	UT130A2
7357	4142	UT130B1	7372	4143	UT130B2	7415	4146	UT131A1	7430	4147	UT131A2
7466	4152	UT131B1	7481	4153	UT131B2	7518	4155	UT132A1	7534	4156	UT132A2
7563	4160	UT132B1	7579	4161	UT132B2	7638	4165	UT133A2	7685	4170	UT133B2
7750	4175	UT134A2	7798	4200	UT134B2	7863	4205	UT135A2	7911	4210	UT135B2
7969	4214	UT136A2	8017	4217	UT136B2	8125	4223	UT320A	8146	4224	UT320B
8184	4230	UT330C	8204	4231	UT330D	8244	4235	UT320E	8263	4236	UT320F
8429	4243	UT350A	8456	4245	UT350B	8484	4247	UT350C	8512	4251	UT350D
8581	4256	UT351A	8603	4260	UT351B	8625	4262	UT351C	8647	4264	UT351D
8772	4276	UT352A	8704	4270	UT352B	8726	4272	UT352C	8748	4274	UT352D
8860	6005	UT361D	8902	6006	UT361E	8949	6013	UT362A	8996	6016	UT362D
9017	6017	UT362E	9056	6018	UT363A	9076	6024	UT363B	9104	6026	UT364A
9131	6030	UT364B	9194	6035	UT365B	9263	6043	UT365B	9345	6053	UT367B
9391	6060	UT370A	9412	6061	UT370B	9452	6065	UT370C	9474	6066	UT370D
9522	6073	GOBUT371A	9545	6074	GOBUT371B	9571	6101	GOBUT373B	10552	7064	UT375A
10575	5301	GOBUT375B	10626	5303	UT376A	1076	6116	UT410C	10673	6121	UT410D
10930	6124	GOBUT410E	11064	6135	UT500A	11121	6140	UT500C	11149	6142	UT500D
11184	6145	UT500E	11306	6155	UT503A	11325	6156	UT503B	11346	6157	UT503C
11393	6162	UT503D	11366	6160	UT503D	11440	6165	UT503F	11461	6166	UT503G
11482	6167	UT503H	11502	6170	UT503I	11520	6171	UT503J	11539	6172	UT503K
11636	6201	GOBUT504A	11654	6202	GOBUT504B	11675	6203	GOBUT504C	11702	6205	UT504D
11749	6210	GOBUT504F	11770	6211	GOBUT504G	11791	6212	GOBUT504H	11811	6213	UT504I
11899	6221	GOBUT505B	11919	6223	GOBUT505C	12022	7071	GOBUT506A	12041	6233	GOBUT506B
12062	6234	GOBUT506C	12083	6235	GOBUT506D	12112	6237	GOBUT506E	12189	6246	GOBUT507A
12208	6247	GOBUT507B	12229	6250	GOBUT507C	12250	6251	GOBUT507D	12273	6253	GOBUT507E
12328	6261	GOBUT507F	12381	6264	GOBUT510A	12403	6265	GOBUT510B	12425	6266	GOBUT510C

LINE	LOCN	SYMBOL	LINE	LOCN	SYMBOL	LINE	LOCN	SYMBOL	LINE	LOCN	SYMBOL
12462	6272	COMUT5100	12493	6273	COMUT5100A	12509	6275	COMUT510E	12538	6277	COMUT510F
13555	6351	G UT533A	13595	6355	G UT533B	13636	6360	G UT534A	13671	6363	G UT534B
13711	6376	G UT534C	13745	6371	G UT534D	13791	6375	G UT534E	13831	6403	G UT534F
13856	6405	G UT534G	13912	6412	G UT535A	13952	6416	G UT535B	14005	6423	G UT536A
14046	6427	G UT535B	14089	6432	G UT536C	14135	6437	G UT535D	14182	6443	G UT536E
14222	6447	G UT536F	14428	4303	G UT537A	15275	5276	G UT620A	15293	5367	G UT620B
15447	5402	G UT621B	15465	5403	G UT621C	15483	5404	G UT621D	15557	5411	G UT621G
15755	7110	G UT621H	15623	5415	G UT622A	15646	5417	G UT622B	15871	5432	G UT624D
15809	5434	G UT701B	16018	5435	G UT701D	16537	6770	G UT712C	16671	6503	G UT713C
17127	4753	G UT730A	17175	4341	G UT731B	17196	4752	G UT730C	17219	4443	G UT730C1
17270	4753	G UT730D	17293	4351	G UT731E	17347	4753	G UT731A	17382	4360	G UT731B
17414	4752	G UT731C	17452	4376	G UT731D	17474	4757	G UT731E	17572	4776	G UT740A
17705	4753	G UT740B	17633	4375	G UT740C	17667	4767	G UT740D	17752	4405	G UT761A
17773	4476	G UT761B	17793	4407	G UT761C	17932	4762	G UT762B	17987	4416	G UT762D
18078	4421	G UT762F	18115	4425	G UT763B	18136	4426	G UT763C	18155	4427	G UT763D
10773	7361	G OFOR410	18019	4420	G OFET762E	10832	6112	G UT410B	11300	6154	G UT503A
11421	6164	G OPUT503E	11630	6200	G OPUT504A	11730	6207	G OPUT504E	11879	6220	G OPUT505A
5173	5147	GOTEST101A	5263	5156	GOTEST102A	5289	5160	GOTEST102B	5316	5162	GOTEST102C
5342	5164	GOTEST102D	5409	5173	GOTEST103A	5435	5175	GOTEST103B	5462	5177	GOTEST103C
5473	5201	GOTEST103D	5528	5203	GOTEST104A	5555	5205	GOTEST104B	5606	5207	GOTEST105A
5654	5212	GOTEST105B	5702	5215	GOTEST105C	5729	5217	GOTEST105D	5750	5220	GOTEST105E
7619	4164	GOTEST133A1	7669	4167	GOTEST133B1	7731	4174	GOTEST134A1	7781	4177	GOTEST134B1
7844	4204	GOTEST135A1	7894	4207	GOTEST135B1	7950	4213	GOTEST136A1	8000	4216	GOTEST136B1
9967	5232	GOTEST374A2	10040	5241	GOTEST374B2	10113	5251	GOTEST374C2	10186	5260	GOTEST374D2
10259	5267	GOTEST374E2	10332	5276	GOTEST374F2	10761	6550	GOTEST511A1	12772	7107	GOTEST511A2
12783	7111	GOTEST511B3	12794	7113	GOTEST511B4	12817	6554	GOTEST511B1	12828	7115	GOTEST511B2
12839	7117	GOTEST511B3	12850	7121	GOTEST511B4	13046	6311	GOTEST512A1	13060	6312	GOTEST512A2
13100	6315	GOTEST512B1	13115	6316	GOTEST512B2	13149	6321	GOTEST512C1	13164	6322	GOTEST512C2
13204	6325	GOTEST512D1	13219	6326	GOTEST512D2	13253	6331	GOTEST512E1	13268	6332	GOTEST512E2
13346	6355	GOTEST520A	13374	6337	GOTEST520B	13403	6341	GOTEST520C	13431	6343	GOTEST520D
13460	6345	GOTEST520E	14487	4564	GOTEST551A	14511	4560	GOTEST551B	14528	7157	GOTEST551B1
14552	5626	GOTEST551C	15949	7171	GOTEST701A	15998	6457	GOTEST701C	16074	5444	GOTEST702A
16113	6463	GOTEST702B	16219	6464	GOTEST710B	16248	6466	GOTEST710C	16284	6471	GOTEST710D
16301	6472	GOTEST710E	16387	6474	GOTEST711B	16416	6476	GOTEST711C	16435	6477	GOTEST711D
16519	5450	GOTEST712B	16567	6502	GOTEST712D	16653	5453	GOTEST713B	16702	6505	GOTEST713D
16796	6055	GOTEST720B	16820	4317	GOTEST720C	16901	4321	GOTEST721B	16925	4322	GOTEST721C
17012	4755	GOTEST722B	17036	4326	GOTEST722C	14388	7153	GR3-537A	14394	7154	HR3-537A
13535	6666	INIT533A	13576	6352	INIT533B	13887	6656	INIT535A	13980	6654	INIT536A
14110	6433	INIT536D	13542	6347	INITD533A	13582	6353	INITD533B	13623	6664	INITD534A
13658	6361	INITD534B	13698	6662	INITD534C	13732	6367	INITD534D	13778	6373	INITD534E
13817	6401	INITD534F	13899	6410	INITD535A	13940	6414	INITD535B	13993	6421	INITD536A
14033	6425	INITD536B	14075	6652	INITD536C	14122	6435	INITD536D	14168	6441	INITD536E
14209	6445	INITD536F	2931	6532	INITIALIZE01	2937	7001	INITIALIZE03	2942	7003	INITIALIZE04
2948	7004	INITIALIZE05	2953	7005	INITIALIZE06	2958	7006	INITIALIZE07	2965	4101	INITIALIZE10
2977	4102	INITIALIZE11	2978	4001	INITIALIZE12	14280	7131	INIT537A	18367	7215	INSERT02
18378	7217	INSERT03	18384	7220	INSERT04	18362	7214	INSERTOPREVNO	18373	7216	INSERTREVNO
10802	6107	INTOIR410A	18643	7256	JAMTOD	18822	4777	JAMPP001	18830	4757	JAMUPP002B
18835	4434	JAMUPP002C	18841	7213	JAMUPP002D	18847	4756	JAMUPP003	18854	4746	JAMUPP004
18851	4747	JAMPP005	18867	4435	JAMUPP006	18872	4436	JAMUPP007	18877	4437	JAMUPP010
18882	4440	JAMPP011	18176	4763	KILL764A	12933	7114	KTSRCDST01	12938	7123	KTSRCDST02
12944	7124	KTSRCDST03	12951	7125	KTSRCDST04	12958	7126	KTSRCDST04B	12967	7335	KTSRCDST05
12973	7337	KTSRCDST06	12981	7127	KTSRCDST07	12987	7130	KTSRCDST08	5252	5154	LOAD00-102A

LINE	LOCN	SYMBOL
1481		L09001-102A
1482		L09001-61082
1483		L090010
1484		L0900146
1485		L090017A
1486		L090022A
1487		L0900246
1488		L090026A
1489		L090034A
1490		L090040A
1491		L090044A
1492		L09005-351
1493		L090050B
1494		L09007-103A
1495		L09014637A
1496		L09017-103A
1497		L090710A
1498		A702B
1499		0102C
1500		0103C
1501		AT17309
1502		CC503A
1503		CC503A
1504		CC503A
1505		CC503A
1506		CC503A
1507		CC503A
1508		CC503A
1509		CC503A
1510		CC503A
1511		CC503A
1512		CC503A
1513		CC503A
1514		CC503A
1515		CC503A
1516		CC503A
1517		CC503A
1518		CC503A
1519		CC503A
1520		CC503A
1521		CC503A
1522		CC503A
1523		CC503A
1524		CC503A
1525		CC503A
1526		CC503A
1527		CC503A
1528		CC503A
1529		CC503A
1530		CC503A
1531		CC503A
1532		CC503A
1533		CC503A
1534		CC503A
1535		CC503A
1536		CC503A
1537		CC503A
1538		CC503A
1539		CC503A
1540		CC503A
1541		CC503A
1542		CC503A
1543		CC503A
1544		CC503A
1545		CC503A
1546		CC503A
1547		CC503A
1548		CC503A
1549		CC503A
1550		CC503A
1551		CC503A
1552		CC503A
1553		CC503A
1554		CC503A
1555		CC503A
1556		CC503A
1557		CC503A
1558		CC503A
1559		CC503A
1560		CC503A
1561		CC503A
1562		CC503A
1563		CC503A
1564		CC503A
1565		CC503A
1566		CC503A
1567		CC503A
1568		CC503A
1569		CC503A
1570		CC503A
1571		CC503A
1572		CC503A
1573		CC503A
1574		CC503A
1575		CC503A
1576		CC503A
1577		CC503A
1578		CC503A
1579		CC503A
1580		CC503A
1581		CC503A
1582		CC503A
1583		CC503A
1584		CC503A
1585		CC503A
1586		CC503A
1587		CC503A
1588		CC503A
1589		CC503A
1590		CC503A
1591		CC503A
1592		CC503A
1593		CC503A
1594		CC503A
1595		CC503A
1596		CC503A
1597		CC503A
1598		CC503A
1599		CC503A
1600		CC503A
1601		CC503A
1602		CC503A
1603		CC503A
1604		CC503A
1605		CC503A
1606		CC503A
1607		CC503A
1608		CC503A
1609		CC503A
1610		CC503A
1611		CC503A
1612		CC503A
1613		CC503A
1614		CC503A
1615		CC503A
1616		CC503A
1617		CC503A
1618		CC503A
1619		CC503A
1620		CC503A
1621		CC503A
1622		CC503A
1623		CC503A
1624		CC503A
1625		CC503A
1626		CC503A
1627		CC503A
1628		CC503A
1629		CC503A
1630		CC503A
1631		CC503A
1632		CC503A
1633		CC503A
1634		CC503A
1635		CC503A
1636		CC503A
1637		CC503A
1638		CC503A
1639		CC503A
1640		CC503A
1641		CC503A
1642		CC503A
1643		CC503A
1644		CC503A
1645		CC503A
1646		CC503A
1647		CC503A
1648		CC503A
1649		CC503A
1650		CC503A
1651		CC503A
1652		CC503A
1653		CC503A
1654		CC503A
1655		CC503A
1656		CC503A
1657		CC503A
1658		CC503A
1659		CC503A
1660		CC503A
1661		CC503A
1662		CC503A
1663		CC503A
1664		CC503A
1665		CC503A
1666		CC503A
1667		CC503A
1668		CC503A
1669		CC503A
1670		CC503A
1671		CC503A
1672		CC503A
1673		CC503A
1674		CC503A
1675		CC503A
1676		CC503A
1677		CC503A
1678		CC503A
1679		CC503A
1680		CC503A
1681		CC503A
1682		CC503A
1683		CC503A
1684		CC503A
1685		CC503A
1686		CC503A
1687		CC503A
1688		CC503A
1689		CC503A
1690		CC503A
1691		CC503A
1692		CC503A
1693		CC503A
1694		CC503A
1695		CC503A
1696		CC503A
1697		CC503A
1698		CC503A
1699		CC503A
1700		CC503A
1701		CC503A
1702		CC503A
1703		CC503A
1704		CC503A
1705		CC503A
1706		CC503A
1707		CC503A
1708		CC503A
1709		CC503A
1710		CC503A
1711		CC503A
1712		CC503A
1713		CC503A
1714		CC503A
1715		CC503A
1716		CC503A
1717		CC503A
1718		CC503A
1719		CC503A
1720		CC503A
1721		CC503A
1722		CC503A
1723		CC503A
1724		CC503A
1725		CC503A
1726		CC503A
1727		CC503A
1728		CC503A
1729		CC503A
1730		CC503A
1731		CC503A
1732		CC503A
1733		CC503A
1734		CC503A
1735		CC503A
1736		CC503A
1737		CC503A
1738		CC503A
1739		CC503A
1740		CC503A
1741		CC503A
1742		CC503A
1743		CC503A
1744		CC503A
1745		CC503A
1746		CC503A
1747		CC503A
1748		CC503A
1749		CC503A
1750		CC503A
1751		CC503A
1752		CC503A
1753		CC503A
1754		CC503A
1755		CC503A
1756		CC503A
1757		CC503A
1758		CC503A
1759		CC503A
1760		CC503A
1761		CC503A
1762		CC503A
1763		CC503A
1764		CC503A
1765		CC503A
1766		CC503A
1767		CC503A
1768		CC503A
1769		CC503A
1770		CC503A
1771		CC503A
1772		CC503A
1773		CC503A
1774		CC503A
1775		CC503A
1776		CC503A
1777		CC503A
1778		CC503A
1779		CC503A
1780		CC503A
1781		CC503A
1782		CC503A
1783		CC503A
1784		CC503A
1785		CC503A
1786		CC503A
1787		CC503A
1788		CC503A
1789		CC503A
1790		CC503A
1791		CC503A
1792		CC503A
1793		CC503A
1794		CC503A
1795		CC503A
1796		CC503A
1797		CC503A
1798		CC503A
1799		CC503A
1800		CC503A
1801		CC503A
1802		CC503A
1803		CC503A
1804		CC503A
1805		CC503A
1806		CC503A
1807		CC503A
1808		CC503A
1809		CC503A
1810		CC503A
1811		CC503A
1812		CC503A
1813		CC503A
1814		CC503A
1815		CC503A
1816		CC503A
1817		CC503A
1818		CC503A
1819		CC503A
1820		CC503A
1821		CC503A
1822		CC503A
1823		CC503A
1824		CC503A
1825		CC503A
1826		CC503A
1827		CC503A
1828		CC503A
1829		CC503A
1830		CC503A
1831		CC503A
1832		CC503A
1833		CC503A
1834		CC503A
1835		CC503A
1836		CC503A
1837		CC503A
1838		CC503A
1839		CC503A
1840		CC503A
1841		CC503A
1842		CC503A
1843		CC503A
1844		CC503A
1845		CC503A
1846		CC503A
1847		CC503A
1848		CC503A
1849		CC503A
1850		CC503A
1851		CC503A
1852		CC503A
1853		CC503A
1854		CC503A
1855		CC503A
1856		CC503A
1857		CC503A
1858		CC503A
1859		CC503A
1860		CC503A
1861		CC503A
1862		CC503A
1863		CC503A
1864		CC503A
1865		CC503A
1866		CC503A
1867		CC503A
1868		CC503A
1869		CC503A
1870		CC503A
1871		CC503A
1872		CC503A
1873		CC503A
1874		CC503A
1875		CC503A
1876		CC503A
1877		CC503A
1878		CC503A
1879		CC503A
1880		CC503A
1881		CC503A
1882		CC503A
1883		CC503A
1884		CC503A
1885		CC503A
1886		CC503A
1887		CC503A
1888		CC503A
1889		CC503A
1890		CC503A
1891		CC503A
1892		CC503A
1893		CC503A
1894		CC503A
1895		CC503A
1896		CC503A
1897		CC503A
1898		CC503A
1899		CC503A
1900		CC503A

LINE	LOCN	SYMBOL
1475B		L09001-610A1
1476B		L09001-610C1
1477B		L09001
1478B		L090011
1479B		L090011537A
1480B		L09001102A
1481B		L0900117A
1482B		L0900137A
1483B		L090



LINE	LOCN	SYMBOL
10519	5506	SETSP375A
14506	4625	SETSP551B
11965	6223	SETUP0M06A
13367	6336	SETL
5862	7015	SETUP00A
5843	7011	SETUPCSP14A
15197	7156	SETUPC00C
10428	7047	SFDT01A
10446	7052	SFDT01D
10470	7015	SFDT0SRH
9330	6013	SHIFT366C
18747	7266	SINTOIR
9539	6077	SINT372A
13492	7133	SINT372B
2698	5525	TEST004
2772	4775	TEST010
3013	5771	TEST012C
3169	5763	TEST012G
3272	5753	TEST0130
3278	5745	TEST014A
3473	5743	TEST015A
3598	5737	TEST017A
3711	5727	TEST020C
3833	5721	TEST022A
3986	5707	TEST025A
4147	5712	TEST030B
4253	5675	TEST031B
4398	5665	TEST034A
4574	5635	TEST040A
4703	5645	TEST042A
4833	5635	TEST044A
4918	5661	TEST047A
5108	5726	TEST050B
5303	5764	TEST102C
5449	5576	TEST103C
5592	5591	TEST105A
5690	5642	TEST105C
5706	4603	TEST114A2
5767	4604	TEST115A4
6044	#	TEST11574
6164	#	TEST11501
6257	#	TEST116A2
6258	#	TEST116B
6413	4631	TEST116C4
6517	4644	TEST117A2
6592	4640	TEST117B2
6685	4667	TEST120A1
6767	4652	TEST120B2
6854	4676	TEST121A2
6928	4662	TEST121B2
7009	4605	TEST121C2
7083	4726	TEST121D2

LINE	LOCN	SYMBOL
9757	7037	SETSP373A
14546	4651	SETSP551C
13001	6	SETUP512A
12333	6	SETUP512B
12334	6	SETUP512C
5737	7013	SETUPC05A
5837	7010	SETUPCSP15A
15132	7161	SETUPC0C
10423	7040	SFDT01A
10452	7053	SFDT01E
10476	7017	SFDT05A1
9381	6016	SHIFT370A
18777	7212	SINTOIRS
9689	5713	SINT372B
2633	4600	TEST1001
2711	5146	TEST005
2847	5777	TEST011
3108	5767	TEST012D
3205	5761	TEST013A
3291	5751	TEST013E
3394	5752	TEST014B
3500	5772	TEST015B
3628	5736	TEST017B
3746	5725	TEST021A
3877	5717	TEST023A
4029	5705	TEST026A
4166	5713	TEST030C
4288	5673	TEST032A
4442	5633	TEST035A
4602	5643	TEST040B
4731	5643	TEST042B
4878	5633	TEST045A
5019	5670	TEST047B
5161	5553	TEST101A
5330	5554	TEST1020
5476	5674	TEST1030
5619	5610	TEST105A1
5716	5636	TEST105D
5939	4601	TEST115A1
6021	4614	TEST115B1
6104	4611	TEST115C1
6196	4625	TEST115D2
6271	#	TEST116A3
6359	#	TEST116C1
6429	4630	TEST116C5
6534	4643	TEST117A3
6609	4657	TEST117B3
6707	4655	TEST120A2
6784	4651	TEST120B3
6871	4675	TEST121A3
6945	4661	TEST121B3
7025	4717	TEST121C3
7100	4737	TEST121D3

LINE	LOCN	SYMBOL
12310	6257	SETSP507F
15395	57401	SETSP521A
13007	6307	SETUP512D
13424	6343	SETUP5200
5737	7014	SETUPC07A
5737	7002	SETUPCSP16A
17275	4344	SETUPC07A
10434	7010	SFDT01B
10459	7054	SFDT05A
9015	6021	SHIFT363A
9442	6013	SHIFT370C
8343	4640	START350
13473	7122	SINTTEST01
2698	6212	TEST002
2730	4474	TEST006
3031	5775	TEST012A
3127	5765	TEST012E
3233	5757	TEST013B
3312	5750	TEST013F
3414	5762	TEST014C
3519	5722	TEST015C
3662	5733	TEST020A
3775	5723	TEST021B
3920	5715	TEST024A
4074	5703	TEST027A
4189	5734	TEST030D
4317	5671	TEST032B
4486	5661	TEST036A
4638	5651	TEST041A
4767	5641	TEST043A
4920	5631	TEST046A
5039	5730	TEST047C
5225	5521	TEST102A
5370	5531	TEST103A
5516	5547	TEST104A
5642	5652	TEST105B
5743	5620	TEST105E
5983	4616	TEST115A2
6060	4613	TEST115B2
6126	4617	TEST115C2
6203	4624	TEST115D3
6278	4637	TEST116A4
6381	4633	TEST116C2
6452	4645	TEST116D
6549	4642	TEST117A4
6633	4656	TEST117C1
6724	4654	TEST120A3
6799	4650	TEST120B4
6885	4674	TEST121A4
6960	4660	TEST121B4
7040	4716	TEST121C4
7115	4736	TEST121D4

LINE	LOCN	SYMBOL
14481	4306	SETSP551A
15771	6454	SETSP524A
13338	6334	SETUP520A
13452	6344	SETUP520E
5847	7012	SETUPCSP12A
5826	5543	SETUPCSP17A
10415	5746	SFDT05A
10440	7051	SFDT05AC
10464	7015	SFDT05AG
9230	6040	SHIFT366A
9511	6071	SHIFT371A
2816	7000	SINT010
13487	7132	SINTTEST02
2673	6631	TEST003
2749	4377	TEST007
3069	5773	TEST012B
3147	5764	TEST012F
3253	5755	TEST013C
3331	5747	TEST013G
3436	5710	TEST014D
3555	5741	TEST015A
3689	5731	TEST020A
3797	5774	TEST021C
3949	5711	TEST024B
4117	5701	TEST030A
4224	5677	TEST031A
4354	5667	TEST033A
4531	5657	TEST037A
4667	5647	TEST041B
4796	5637	TEST043B
4950	5770	TEST046B
5077	5551	TEST050A
5277	5574	TEST102B
5423	5572	TEST103B
5543	5646	TEST104B
5666	5600	TEST105B1
5874	4607	TEST114A
6000	4615	TEST115A3
6077	4612	TEST115B3
6143	4627	TEST115C3
6235	4623	TEST116A1
6305	4636	TEST116A5
6396	4632	TEST116C3
6495	4647	TEST117A1
6570	4641	TEST117B1
6655	4677	TEST117C2
6745	4653	TEST120B1
6832	4665	TEST121A1
6906	4663	TEST121B1
6987	4707	TEST121C1
7061	4727	TEST121D1
7148	4671	TEST122A1

LINE	LOCN	SYMBOL
7172	4570	TEST122A2
7314	4550	TEST130A2
7423	4544	TEST131A2
7527	4516	TEST132A2
7630	4500	TEST133A2
7742	4503	TEST134A2
7855	4506	TEST135A2
7961	4476	TEST136A2
8138	4552	TEST320B
8256	4556	TEST320F
8471	4742	TEST350C
8612	4723	TEST351C
8713	4724	TEST352C
8894	6767	TEST361E
9010	6755	TEST362E
9119	6745	TEST364B
9256	6735	TEST366B
9405	6723	TEST370B
9508	6713	TEST371B
9814	6572	TEST373B
10105	5524	TEST374C2
10512	5517	TEST375A
10666	5507	TEST410
10879	6727	TEST410D
11100	6726	TEST500C
11318	6630	TEST503B
11407	6700	TEST503E
11495	6742	TEST503I
11647	6730	TEST504B
11742	6672	TEST504F
11847	6723	TEST505A
12034	6754	TEST505B
12144	6613	TEST507A
12265	6674	TEST507E
12418	6616	TEST510C
12524	6610	TEST510F
12789	7112	TEST511A4
12845	7120	TEST511B4
13108	6546	TEST512B2
13212	6576	TEST512D2
13360	6764	TEST520B
13528	6625	TEST533A
13691	6665	TEST534C
13844	6752	TEST534G
14019	6640	TEST536B
14196	6642	TEST536F
14539	4566	TEST551C
14919	5612	TEST610B2
15145	5624	TEST610D2
15344	5617	TEST621A
15495	5605	TEST621E
15605	5601	TEST622A

LINE	LOCN	SYMBOL
7188	4734	TEST122A3
7334	4530	TEST130B1
7443	4546	TEST131B1
7547	4517	TEST132B1
7652	4501	TEST133B1
7764	4504	TEST134B1
7877	4507	TEST135B1
7983	4475	TEST136B1
8157	4562	TEST320C
8337	4701	TEST350
8499	4743	TEST350D
8634	4712	TEST351D
8735	4725	TEST352D
8917	6765	TEST362A
9033	6753	TEST363A
9146	6743	TEST365A
9278	6733	TEST366C
9429	6721	TEST370C
9606	6571	TEST372A
9898	6563	TEST374
10178	5500	TEST374D2
10563	5625	TEST375B
10790	6711	TEST410A
10911	6705	TEST410E
11136	6746	TEST500D
11338	6632	TEST503C
11433	6710	TEST503F
11513	6756	TEST503J
11667	6740	TEST504C
11762	6702	TEST504G
11891	6712	TEST505B
12054	6744	TEST506C
12201	6724	TEST507B
12296	6647	TEST507F
12458	6600	TEST510D
12755	6607	TEST511A
12812	6551	TEST511B1
13026	6542	TEST512A1
13130	6556	TEST512C1
13234	6577	TEST512E1
13388	6772	TEST520C
13569	6633	TEST533B
13725	6635	TEST534D
13879	6661	TEST535A
14067	6655	TEST536C
14413	4572	TEST537A
14744	5627	TEST610A1
14987	5535	TEST610C1
15262	5511	TEST620A
15440	5613	TEST621B
15521	5603	TEST621F
15634	5577	TEST622B

LINE	LOCN	SYMBOL
7202	4735	TEST122A4
7365	4532	TEST130B2
7474	4750	TEST131A2
7572	4472	TEST132
7679	4502	TEST133
7791	4505	TEST134
7904	4477	TEST135
8010	4473	TEST136
8197	4537	TEST320
8417	4752	TEST350A
8569	4732	TEST351A
8707	4705	TEST352A
8827	6573	TEST361A
8962	6763	TEST362B
9069	6751	TEST363B
9187	6741	TEST365B
9317	6731	TEST367A
9467	6717	TEST370D
9658	6701	TEST372B
9959	5504	TEST374A2
10251	5502	TEST374E2
10535	5513	TEST376A
10814	6704	TEST410B
10968	6751	TEST500
11164	6736	TEST500E
11379	6670	TEST503D
11453	6720	TEST503G
11531	6646	TEST503K
11698	6750	TEST504D
11783	6706	TEST504H
11912	6722	TEST505C
12076	6734	TEST506D
12221	6611	TEST507C
12363	6603	TEST510A
12475	6617	TEST510A
12767	7074	TEST511A2
12823	7106	TEST511B2
13053	6544	TEST512A2
13157	6566	TEST512C2
13261	6535	TEST512E2
13417	6762	TEST520D
13615	6667	TEST534A
13765	6663	TEST534E
13926	6637	TEST535B
14103	6641	TEST536D
14469	4575	TEST551A
14806	5622	TEST610A2
15038	5602	TEST610C2
15286	5623	TEST620B
15458	5611	TEST621C
15550	5766	TEST621G
15657	5575	TEST622C

LINE	LOCN	SYMBOL
7283	4673	TEST130A1
7392	4541	TEST131A1
7502	4521	TEST132A1
7600	4520	TEST133A1
7705	4511	TEST134A1
7819	4513	TEST135A1
7931	4515	TEST136A1
8098	4543	TEST320A
8217	4554	TEST320E
8444	4753	TEST350B
8591	4722	TEST351B
8732	4713	TEST352B
8873	6771	TEST361D
8989	6757	TEST362D
9091	6747	TEST364A
9209	6737	TEST366A
9360	6725	TEST370A
9490	6715	TEST371A
9738	6541	TEST373A
10032	5606	TEST374B2
10324	5504	TEST374F2
10619	5614	TEST376A1
10844	6707	TEST410C
11070	6673	TEST500A
11238	6677	TEST503A
11359	6601	TEST503D
11474	6732	TEST503H
11578	6615	TEST504A
11716	6760	TEST504E
11804	6716	TEST504I
11952	6621	TEST506A
12099	6605	TEST506E
12243	6557	TEST507D
12396	6675	TEST510B
12497	6604	TEST510E
12778	7110	TEST511A3
12834	7116	TEST511B3
13081	6565	TEST512B1
13185	6537	TEST512D1
13322	6627	TEST520A
13445	6775	TEST520E
13559	6634	TEST534B
13805	6636	TEST534F
13973	6657	TEST536A
14154	6653	TEST536E
14498	4576	TEST551B
14869	5533	TEST610B1
15101	5545	TEST610D1
15309	5621	TEST620C
15476	5607	TEST621D
15572	5754	TEST621H
15680	5573	TEST6223



LINE	LOCN	SYMBOL	LINE	LOCN	SYMBOL	LINE	LOCN	SYMBOL	LINE	LOCN	SYMBOL
15724	5571	TEST624A	15817	5565	TEST624B	15939	5776	TEST624C	15864	5756	TEST624D
15925	5563	TEST701A	15962	5523	TEST701B	15981	6770	TEST701C	16011	5512	TEST701D
16040	5477	TEST702A	16089	6761	TEST702B	16157	5475	TEST710A	16212	6644	TEST710B
16236	6645	TEST710C	16265	6574	TEST710D	16294	6575	TEST710E	16328	5473	TEST711A
16374	6534	TEST711B	16404	6530	TEST711C	16427	6536	TEST711D	16460	5471	TEST712A
16512	5505	TEST712B	16530	6524	TEST712C	16555	6533	TEST712D	16594	5467	TEST713A
16640	5616	TEST713B	16664	6522	TEST713C	16689	6521	TEST713D	16730	5465	TEST720A
16788	6520	TEST720B	16813	4760	TEST720C	16840	4571	TEST721A	16894	4573	TEST721B
16918	4577	TEST721C	16951	5463	TEST722A	17004	4464	TEST722B	17029	4465	TEST722C
17084	4471	TEST730A	17149	4466	TEST730B	17188	4460	TEST730C	17211	4460	TEST730C1
17233	4461	TEST730D	17296	4462	TEST730E	17321	4755	TEST731A	17353	4457	TEST731B
17401	4456	TEST731C	17428	4465	TEST731D	17467	4454	TEST731E	17476	4744	TEST740A
17591	4453	TEST740B	17619	4462	TEST740C	17653	4451	TEST740D	17726	4745	TEST761A
17765	4771	TEST761B	17786	4774	TEST761C	17689	4773	TEST762A	17899	4553	TEST762B1
17924	4751	TEST762B	17946	4740	TEST762C	17980	4741	TEST762D	18006	4730	TEST762E
18031	4711	TEST762F	18072	4761	TEST763A	18108	4720	TEST763B	18128	4721	TEST763C
18148	4710	TEST763D	11292	6714	TESTA03A	11621	6766	TESTA04A	10707	6773	TESTD410
11008	6671	TESTD500	11031	6777	TESTINHSP500	11023	6133	TESTINHSP500	18089	4424	TESTVECT763A
15400	5522	UBRK621A	15776	6255	UL 624A	13329	6774	UCOM 62A	18103	4733	VECTLOAD763A
18301	6774	VFY001	18308	4412	VFYL 2	18313	6775	VFY003	18321	4433	VFY004
18333	7375	VFY005	18340	7374	VFY006	10755	7276	WR 0410	8364	4002	WRITDF350
8371	7016	WRITESF350	17861	4315	ZAP0762A	17122	4374	Z 0730A	9919	5536	ZER0374A1
9941	5227	ZER0374A2	9992	5736	ZER0374B1	10014	5736	Z 0742	10065	5566	ZER0374C1
10087	5246	ZER0374C2	10138	5514	ZER0374D1	10160	5735	Z 0374D2	10211	5526	ZER0374E1
10233	5264	ZER0374E2	10284	5516	ZER0374F1	10306	5273	Z 0374F2	12504	6274	ZER00510E
17734	4772	ZER00SR761A	12531	6276	ZER0FLAG510E	15724	5427	Z 0748	17739	4403	ZER01T761A
10386	7044	ZER0SF	10365	7036	Z 0748 OF 04	10370	7041	Z 0748 OF 02	10392	7045	ZER0SFA
10375	7042	ZER0SDF	10381	7043	Z 0748 OF 04	19252	7400	ZTA ET400	19254	7401	ZTARGET401
19256	7402	ZTARGET402	19258	7403	ZTA ET403	19250	7404	ZTA ET404	19262	7405	ZTARGET405
19264	7406	ZTARGET406	19266	7407	ZTA ET407	19258	7410	ZTA ET410	19270	7411	ZTARGET411
19272	7412	ZTARGET412	19274	7413	ZTA ET413	19276	7414	ZTA ET414	19278	7415	ZTARGET415
19280	7416	ZTARGET416	19282	7417	ZTA ET417	19286	7420	ZTA ET420	19286	7421	ZTARGET421
19288	7422	ZTARGET422	19290	7423	ZTA ET423	19292	7424	ZTA ET424	19294	7425	ZTARGET425
19296	7426	ZTARGET426	19298	7427	ZTA ET427	19300	7430	ZTA ET430	19302	7431	ZTARGET431
19304	7432	ZTARGET432	19306	7433	ZTA ET433	19308	7434	ZTA ET434	19310	7435	ZTARGET435
19312	7436	ZTARGET436	19314	7437	ZTA ET437	19316	7440	ZTA ET440	19318	7441	ZTARGET441
19320	7442	ZTARGET442	19322	7443	ZTA ET443	19324	7444	ZTA ET444	19326	7445	ZTA ET445
19328	7446	ZTARGET446	19330	7447	ZTA ET447	19332	7450	ZTA ET450	19334	7451	ZTA ET451
19336	7452	ZTARGET452	19338	7453	ZTA ET453	19340	7454	ZTA ET454	19342	7455	ZTA ET455
19344	7456	ZTARGET456	19346	7457	ZTA ET457	19348	7460	ZTA ET460	19350	7461	ZTA ET461
19352	7462	ZTARGET462	19354	7463	ZTA ET463	19356	7464	ZTA ET464	19358	7465	ZTA ET465
19360	7466	ZTARGET466	19362	7467	ZTA ET467	19364	7470	ZTA ET470	19366	7471	ZTA ET471
19368	7472	ZTARGET472	19370	7473	ZTA ET473	19372	7474	ZTA ET474	19374	7475	ZTA ET475
19376	7476	ZTARGET476	19378	7477	ZTA ET477	19380	7500	ZTA ET500	19382	7501	ZTA ET501
19384	7502	ZTARGET502	19386	7503	ZTA ET503	19388	7504	ZTA ET504	19390	7505	ZTA ET505
19392	7506	ZTARGET506	19394	7507	ZTA ET507	19396	7510	ZTA ET510	19398	7511	ZTA ET511
19400	7512	ZTARGET512	19402	7513	ZTA ET513	19404	7514	ZTA ET514	19406	7515	ZTARGET515
19408	7516	ZTARGET516	19410	7517	ZTA ET517	19412	7520	ZTARGET520	19414	7521	ZTARGET521
19416	7522	ZTARGET522	19418	7523	ZTARGET523	19420	7524	ZTARGET524	19422	7525	ZTARGET525
19424	7526	ZTARGET526	19426	7527	ZTARGET527	19428	7530	ZTARGET530	19430	7531	ZTARGET531
19432	7532	ZTARGET532	19434	7533	ZTARGET533	19436	7534	ZTARGET534	19438	7535	ZTARGET535
19440	7536	ZTARGET536	19442	7537	ZTARGET537	19444	7540	ZTARGET540	19446	7541	ZTARGET541

LINE	LOCN	SYMBOL	LINE	LOCN	SYMBOL	LINE	LOCN	SYMBOL	LINE	LOCN	SYMBOL
19448	7542	ZTARGET542	19450	7543	ZTARGET543	19452	7544	ZTARGET544	19454	7545	ZTARGET545
19456	7546	ZTARGET546	19458	7547	ZTARGET547	19460	7550	ZTARGET550	19462	7551	ZTARGET551
19464	7552	ZTARGET552	19466	7553	ZTARGET553	19468	7554	ZTARGET554	19470	7555	ZTARGET555
19472	7556	ZTARGET556	19474	7557	ZTARGET557	19476	7560	ZTARGET560	19478	7561	ZTARGET561
19480	7562	ZTARGET562	19482	7563	ZTARGET563	19484	7564	ZTARGET564	19486	7565	ZTARGET565
19488	7566	ZTARGET566	19490	7567	ZTARGET567	19492	7570	ZTARGET570	19494	7571	ZTARGET571
19496	7572	ZTARGET572	19498	7573	ZTARGET573	19500	7574	ZTARGET574	19502	7575	ZTARGET575
19504	7576	ZTARGET576	19506	7577	ZTARGET577	19508	7600	ZTARGET600	19510	7601	ZTARGET601
19512	7602	ZTARGET602	19514	7603	ZTARGET603	19516	7604	ZTARGET604	19518	7605	ZTARGET605
19520	7606	ZTARGET606	19522	7607	ZTARGET607	19524	7610	ZTARGET610	19526	7611	ZTARGET611
19528	7612	ZTARGET612	19530	7613	ZTARGET613	19532	7614	ZTARGET614	19534	7615	ZTARGET615
19536	7616	ZTARGET616	19538	7617	ZTARGET617	19540	7620	ZTARGET620	19542	7621	ZTARGET621
19544	7622	ZTARGET622	19546	7623	ZTARGET623	19548	7624	ZTARGET624	19550	7625	ZTARGET625
19552	7626	ZTARGET626	19554	7627	ZTARGET627	19556	7630	ZTARGET630	19558	7631	ZTARGET631
19560	7632	ZTARGET632	19562	7633	ZTARGET633	19564	7634	ZTARGET634	19566	7635	ZTARGET635
19568	7636	ZTARGET636	19570	7637	ZTARGET637	19572	7640	ZTARGET640	19574	7641	ZTARGET641
19576	7642	ZTARGET642	19578	7643	ZTARGET643	19580	7644	ZTARGET644	19582	7645	ZTARGET645
19584	7646	ZTARGET646	19586	7647	ZTARGET647	19588	7650	ZTARGET650	19590	7651	ZTARGET651
19592	7652	ZTARGET652	19594	7653	ZTARGET653	19596	7654	ZTARGET654	19598	7655	ZTARGET655
19600	7656	ZTARGET656	19602	7657	ZTARGET657	19604	7660	ZTARGET660	19606	7661	ZTARGET661
19608	7662	ZTARGET662	19610	7663	ZTARGET663	19612	7664	ZTARGET664	19614	7665	ZTARGET665
19616	7666	ZTARGET666	19618	7667	ZTARGET667	19620	7670	ZTARGET670	19622	7671	ZTARGET671
19624	7672	ZTARGET672	19626	7673	ZTARGET673	19628	7674	ZTARGET674	19630	7675	ZTARGET675
19632	7676	ZTARGET676	19634	7677	ZTARGET677	19636	7700	ZTARGET700	19638	7701	ZTARGET701
19640	7702	ZTARGET702	19642	7703	ZTARGET703	19644	7704	ZTARGET704	19646	7705	ZTARGET705
19648	7706	ZTARGET706	19650	7707	ZTARGET707	19652	7710	ZTARGET710	19654	7711	ZTARGET711
19656	7712	ZTARGET712	19658	7713	ZTARGET713	19660	7714	ZTARGET714	19662	7715	ZTARGET715
19664	7716	ZTARGET716	19666	7717	ZTARGET717	19668	7720	ZTARGET720	19670	7721	ZTARGET721
19672	7722	ZTARGET722	19674	7723	ZTARGET723	19676	7724	ZTARGET724	19678	7725	ZTARGET725
19680	7726	ZTARGET726	19682	7727	ZTARGET727	19684	7730	ZTARGET730	19686	7731	ZTARGET731
19688	7732	ZTARGET732	19690	7733	ZTARGET733	19692	7734	ZTARGET734	19694	7735	ZTARGET735
19696	7736	ZTARGET736	19698	7737	ZTARGET737	19700	7740	ZTARGET740	19702	7741	ZTARGET741
19704	7742	ZTARGET742	19706	7743	ZTARGET743	19708	7744	ZTARGET744	19710	7745	ZTARGET745
19712	7746	ZTARGET746	19714	7747	ZTARGET747	19716	7750	ZTARGET750	19718	7751	ZTARGET751
19720	7752	ZTARGET752	19722	7753	ZTARGET753	19724	7754	ZTARGET754	19726	7755	ZTARGET755
19728	7756	ZTARGET756	19730	7757	ZTARGET757	19732	7760	ZTARGET760	19734	7761	ZTARGET761
19736	7762	ZTARGET762	19738	7763	ZTARGET763	19740	7764	ZTARGET764	19742	7765	ZTARGET765
19744	7766	ZTARGET766	19746	7767	ZTARGET767	19748	7770	ZTARGET770	19750	7771	ZTARGET771
19752	7772	ZTARGET772	19754	7773	ZTARGET773	19756	7774	ZTARGET774	19758	7775	ZTARGET775
19760	7776	ZTARGET776	19762	7777	ZTARGET777						

LOCN	-----0-----	-----1-----	-----2-----	-----3-----	-----4-----	-----5-----	-----6-----	-----7-----
4000	TEST001	INITIALIZE12	WRITEDF350	ALU114A	GETZER0ES114A	GOBUT114A	GOBUT114A2	SCOPE114A
4010	GETALTN115A1	GOBUT115A1	GOBUT115A4	GOBUT115A2	GOBUT115A3	ALU115B1	GOBUT115B1	GOBUT115B4
4020	GOBUT115B2	GOBUT115B3	SCOPE115B	GOBUT115C1	GOBUT115C2	GOBUT115C3	ALU115D1	GOBUT115D1
4030	GOBUT115D2	GOBUT115D3	SCOPE115D	GOBUT116A1	GOBUT116A2	GOBUT116A3	GOBUT116A4	GOBUT116A5
4040	ALU116B	GOBUT116B	SCOPE116C	GOBUT116C1	GOBUT116C2	GOBUT116C3	GOBUT116C4	GOBUT116C5
4050	ALU116D	GOBUT116D	SCOPE116D	GOBUT117A1	GOBUT117A2	GOBUT117A3	GOBUT117A4	ALU117B1
4060	GOBUT117B1	GOBUT117B2	GOBUT117B3	ALU117C1	GOBUT117C1	GOBUT117C2	SCOPE117C	GOBUT120A1
4070	GOBUT120A2	GOBUT120A3	ALU120B1	GOBUT120B1	GOBUT120B2	GOBUT120B3	GOBUT120B4	SCOPE120B
4100	LOAD010	INITIALIZE10	INITIALIZE11	GOBUT121A1	GOBUT121A2	GOBUT121A3	GOBUT121A4	ALU121B1
4110	GOBUT121B1	GOBUT121B2	GOBUT121B3	GOBUT121B4	SCOPE121C	GOBUT121C1	GOBUT121C2	GOBUT121C3
4120	GOBUT121C4	ALU121D1	GOBUT121D1	GOBUT121D2	GOBUT121D3	GOBUT121D4	SCOPE121D	GOBUT122A1
4130	GOBUT122A2	GOBUT122A3	SETD122A4	GOBUT122A4	SCOPE122A	COMP130A1	GOBUT130A1	GOBUT130A2
4140	ARITH130B1	COMP130B1	GOBUT130B1	GOBUT130B2	SCOPE130B	COMP131A1	GOBUT131A1	GOBUT131A2
4150	ARITH131B1	COMP131B1	GOBUT131B1	GOBUT131B2	SCOPE131B	GOBUT132A1	GOBUT132A2	ARITH132B1
4160	GOBUT132B1	GOBUT132B2	SCOPE132B	DOPA133A1	GOTEST133A1	GOBUT133A2	OPA133B1	GOTEST133B1
4170	GOBUT133A2	SCOPE133B	OPA134A1	DOPA134A1	GOTEST134A1	GOBUT134A2	OPA134B1	GOTEST134B1
4200	GOBUT133A2	SCOPE134B	OPA135A1	DOPA135A1	GOTEST135A1	GOBUT135A2	OPA135B1	GOTEST135B1
4210	GOBUT133A2	SCOPE135B	DOPA136A1	GOTEST136A1	GOBUT136A2	OPA136B1	GOTEST136B1	GOBUT136B2
4220	SCOPE136B	SETD320A	GETDC320A	GOBUT320A	GOBUT320B	SETONE320C	SETD320C	GETDC320C
4230	GOBUT1370C	GOBUT320D	SETONE320E	SETD320E	GETDC320E	GOBUT320F	GOBUT320F	SCOPE320
4240	SETD350	LOADSRD350	BL1-537A	GOBUT350A	COMP350B	GOBUT350B	COMP350C	GOBUT350C
4250	COMP350D	GOBUT350D	SCOPE350	NEXTPATA350	LOAD06-351	VFY003	GOBUT351A	COMP351B
4260	GOBUT351B	COMP351C	GOBUT351C	COMP351D	GOBUT351D	SCOPE351	GOBUT352A	COMP352B
4270	GOBUT352B	COMP352C	GOBUT352C	COMP352D	GOBUT352D	SCOPE352	LOAD14537A	LOAD003537A
4300	LOAD01537A	COUNTER01	COMP537A	GOBUT537A	SCOPE537A	LOADIN551A	SETSP551A	LOADADR710A
4310	COUNTER02	BUSFCN711A	SETJAM710A	BUSFCN710A	GENADR712A	ZAPD762A	BUSFCN712A	GOTEST720C
4320	LOADRET721A	GOTEST721B	GOTEST721C	SETJAM722A	BUSFCN722A	GOTEST722B	GOTEST722C	SCOPE722
4330	LOADDATA730A	BUSFCN730A	GETDBUF730A	COMP730A	ZAPDMIF730A	GOBUT730A	EXPEC730B	BUSFCN730B
4340	GETIT730B	GOBUT730B	GOBUT730C	LOADDATA730D	SETZLR730D	MANGLED730D	BUSFCN730D	GETIT730D
4350	GOBUT730D	GOBUT730E	SCOPE730	GETLUF731A	COMP731A	GOBUT731A	BUSFCN731B	GETIT731B
4360	GOBUT731B	COMP731C	GOBUT731C	LOADIR731D	BUSFCN731D	MANGLED731D	GOBUT731D	GOBUT731E
4370	SCOPE731	BUSFCN740A	LOADIR740B	BUSFCN740B	BUSFCN740C	GOBUT740C	BUTERROR4	TEST007
4400	LOADIR740D	BUSFCN740D	SCOPE740	ZEROIT761A	CLEAR761A	GOBUT761A	GOBUT761B	GOBUT761C
4410	SCOPE761	DW11L762A	VFY002	SETPRI762A	FILL762C	DELAY762C	GOBUT762D	EXPEC762E
4420	GOGET762E	GOBUT762F	EXPEC763A	READVECT763A	TESTVECT763A	GOBUT763B	GOBUT763C	GOBUT763D
4430	SCOPE763	EOP001		VFY004	JAMUPP002C	JAMUPP006	JAMUPP007	JAMUPP010
4440	JAMUPP011			GOBUT730C1		NEXT710A	NEXT711A	NEXT712A
4450	TEST730C1	TEST740D	TEST740C	TEST740B	TEST731E	NEXT710A	NEXT711A	NEXT712A
4460	TEST730C	TEST730D	TEST730E	SETPR6-762A	TEST722B	TEST731D	TEST731C	TEST731B
4470	SETADR722A	TEST730A	TEST132B2	TEST136B2	TEST722C	TEST722C	TEST730B	NEXTD762A
4500	TEST133A2	TEST133B1	TEST133B2	TEST134A2	TEST006	TEST136B1	TEST136A2	TEST135B2
4510	OPA133A1	TEST134A1	OPA134A1	TEST135A1	TEST134B1	TEST134B2	TEST135A2	TEST135B1
4520	ARITH131A1	TEST132A1	ARITH132A1	TEST133A1	OPA135A1	TEST136A1	TEST132A2	TEST132B1
4530	TEST130B1	COUNTER10B	TEST130B2	COUNTER10A	NEXT722A	COUNTER04	COUNTER03	COUNTER10
4540	ARITH130A1	TEST131A1	OPA136A1	TEST320A	COUNTER06	COUNTER07	COUNTER08	TEST320D
4550	TEST130A2	BUMPD762A	TEST320B	TEST762A1	TEST131A2	COUNTER04	TEST131B1	COUNTER11
4560	GOTEST551B	SETSR551C	TEST320C	DINTOIRA350	TEST320E	COUNTER12	TEST320F	COUNTER05
4570	TEST122A2	TEST721A	TEST537A	TEST721B	GOTEST551A	SETSP551B	TEST551C	LOAD05-351
4600	GETONES114A	TEST115A1	TEST115B4	TEST114A2	LOAD16537A	TEST551A	TEST551B	TEST721C
4610	ALU115A1	TEST115C1	TEST115B3	TEST115B2	TEST115A4	TEST121C2	TEST116B	TEST114A
4620	TEST116A3	TEST116A2	ALU115C1	TEST116A1	TEST115B1	TEST115A3	TEST115A2	TEST115C2
4630	TEST116C5	TEST116C4	TEST116C3	TEST116C2	TEST115D3	TEST115D2	TEST115D1	TEST115C3
					ALU116A1	TEST116C1	TEST116A5	TEST116A4

LOCN	----0-----	----1-----	----2-----	----3-----	----4-----	----5-----	----6-----	----7-----
4640	TEST117B2	TEST117B1	TEST117A4	TEST117A3	TEST117A2	TEST1160	ALU116C1	TEST117A1
4650	TEST127B4	TEST120B3	TEST120B2	TEST120B1	TEST120A3	TEST120A2	TEST117C1	TEST117B3
4660	TEST121B4	TEST121B3	TEST121B2	TEST121B1	ALU120A1	TEST121A1	ALU117A1	TEST120A1
4670	ALU121C1	TEST122A1	ALU122A1	TEST130A1	TEST121A4	TEST121A3	TEST121A2	TEST117C2
4700	SETONE320A	TEST350	COMP350A	RESET1R350	COMP351A	TEST352A	ALU121A1	TEST121C1
4710	TEST7630	TEST762F	TEST351D	TEST352B	COMP352A	RESTORE01	TEST121C4	TEST121C3
4720	TEST763B	TEST763C	TEST351B	TEST351C	TEST352C	TEST352D	TEST121D2	TEST121D1
4730	TEST762E	ALLOW763A	TEST351A	VECTLOAD763A	TEST122A3	TEST122A4	TEST121D4	TEST121D3
4740	TEST762C	TEST762D	TEST350C	TEST3500	TEST740A	TEST761A	JAMUPP004	JAMUPP005
4750	TEST131B2	TEST762B	TEST350A	TEST350B	LOADIR730A	TEST731A	JAMUPP003	JAMUPP002B
4760	TEST720C	TEST763A	GOBUT762B	KILL764A	BUSFCN731A	LOADIR740A	GOBUT740B	GOBUT740D
4770	.	TEST761B	ZERODSR761A	TEST762A	TEST761C	TEST010	GOBUT740A	JAMUPP001

LOCN	-----0-----	-----1-----	-----2-----	-----3-----	-----4-----	-----5-----	-----6-----	-----7-----
5000	SETDC3728	LOAD011	LOAD012A	GOBUTO12A	GOBUTO12B	GOBUTO12C	GOBUTO12D	GOBUTO12E
5010	GOBUTO12E	GOBUTO12G	SCOPE012	GOBUTO13A	GOBUTO13B	GOBUTO13C	GOBUTO13D	GOBUTO13E
5020	GOBUTO13F	GOBUTO13G	SCOPE013	GOBUTO14A	GOBUTO14B	GOBUTO14C	GOBUTO14D	SCOPE014
5030	GOBUTO15A	GOBUTO15B	GOBUTO15C	SCOPE015	GOBUTO16A	SCOPE016	GOBUTO17A	GOBUTO17B
5040	SCOPE017	GOBUTO17A	GOBUTO20B	GOBUTO20C	SCOPE020	GOBUTO21A	GOBUTO21B	GOBUTO21C
5050	SCOPE021	GOBUTO20A	SCOPE022	GOBUTO23A	SCOPE023	GOBUTO24A	GOBUTO24B	SCOPE024
5060	GOBUTO25A	SCOPE025	GOBUTO26A	SCOPE026	GOBUTO27A	SCOPE027	GOBUTO30A	GOBUTO30B
5070	GOBUTO27C	GOBUTO300	SCOPE030	GOBUTO31A	GOBUTO31B	SCOPE031	GOBUTO32A	GOBUTO32B
5100	SCOPE032	GOBUTO33A	SCOPE033	GOBUTO34A	SCOPE034	GOBUTO35A	SCOPE035	GOBUTO36A
5110	SCOPE036	GOBUTO37A	SCOPE037	GOBUTO40A	GOBUTO40B	SCOPE040	GOBUTO41A	GOBUTO41B
5120	SCOPE041	GOBUTO42A	GOBUTO42B	SCOPE042	GOBUTO43A	GOBUTO43B	SCOPE043	GOBUTO44A
5130	SCOPE044	GOBUTO45A	SCOPE045	GOBUTO46A	GOBUTO46B	SCOPE046	GOBUTO47A	GOBUTO47B
5140	GOBUTO47C	SCOPE047	GOBUTO50A	LOAD050B	GOBUTO50B	SCOPE050	TEST005	GOBUTO47B
5150	SCOPE101	LOAD02-102A	LOAD04-102A	LOAD10-102A	LOAD00-102A	LOAD0102A	GOTEST102A	LOAD0102B
5160	GOTEST102B	LOAD0102C	GOTEST102C	LOAD0102D	GOTEST102D	SCOPE102	LOAD15-103A	LOAD13-103A
5170	LOAD07-103A	LOAD17-103A	LOAD0103A	GOTEST103A	LOAD0103B	GOTEST103B	LOAD0103C	GOTEST103C
5200	LOAD0103D	GOTEST103D	SCOPE103	GOTEST104A	LOAD0104B	GOTEST104B	SCOPE104	GOTEST105A
5210	GOBUTO104A1	LOADSR105B	GOTEST105B	GOBUT105B1	LOADSR105C	GOTEST105C	LOADSR105D	GOTEST105D
5220	GOTEST104E	SCOPE105	EXPEC374A1	SUBRA372B	SCOPE624	DOWRITE374A1	GETTEM374A1	ZER0374A2
5230	DOWRITE374A2	GETTEM374A2	GOTEST374A2	SCOPE374A	DOWRITE374B1	GETTEM374B1	ZER0374B2	DOWRITE374B2
5240	GETTEM374C2	GOTEST374B2	SCOPE374B	DOWRITE374C1	GETTEM374C1	NEXT010	ZER0374C2	DOWRITE374C2
5250	GETTEM374C2	GOTEST374C2	SCOPE374C	DOWRITE374D1	GETTEM374D1	ZER0374D2	DOWRITE374D2	GETTEM374D2
5260	GOTEST374D2	SCOPE374D	DOWRITE374E1	GETTEM374E1	ZER0374E2	DOWRITE374E2	GETTEM374E2	GOTEST374E2
5270	SCOPE374E	DOWRITE374F1	GETTEM374F1	ZER0374F2	DOWRITE374F2	GETTEM374F2	GOTEST374F2	SCOPE374F
5300	CH375B	GOBUT375B	SETJ375A	GOBUT376A	SCOPE376	SCOPE551	LOAD01-610A1	LOAD05-610A1
5310	LOAD05-610A1	PSCC-DC610A1	SETT03A610A1	DOIT610A1	GETIT610A1	LOAD06-610A2	LOAD01-610A2	PSCC-DC610A2
5320	SETT03A610A2	DOIT610A2	GETIT610A2	SCOPE610A	LOAD01-610B1	PSCC-DC610B1	SETBUS610B1	DOIT610B1
5330	GETIT610B1	LOAD07-610B2	LOAD07-610B2	LOAD01-610B2	PSCC-DC610B2	SETBUS610B2	DOIT610B2	GETIT610B2
5340	SCOPE610B	LOAD01-610C1	PSCC-DC610C1	SETBUS610C1	DOIT610C1	GETIT610C1	LOAD05-610C2	LOAD01-610C2
5350	PSCC-DC610C2	SETBUS610C2	DOIT610C2	GETIT610C2	SCOPE610C	PSCC-DC610D1	SETBUS610D1	DOIT610D1
5360	GETIT610D1	PSCC-DC610D2	SETBUS610D2	DOIT610D2	GETIT610D2	SCOPE610D	GOBUT610A	GOBUT620B
5370	LOAD620C	GETJAM620C	CSP1L621A	SETBUS621A	LOC621A	SETRET621A	BUTER621A	SETFLG621A
5400	LOADFLG621A	SETSR621A	GOBUT621B	GOBUT621C	GOBUT621D	EXPEC621E	GETJAM621E	EXPEC621F
5410	GETCU621F	GOBUT621G	COMP621H	BCIFCN622A	CNST0622A	GOBUT622A	BCERC622B	GOBUT622B
5420	GETJAM622C	EXPEC623	GETCU623	SCOPE623	SETBRK624A	LOAD0624A	SETRET624A	ZER01R624B
5430	EXPEC624C	GETCU624C	GOBUT624D	SCOPE702	GOBUT701B	GOBUT701D	SCOPE701	EXPEC702A
5440	DOIT376A	SETLED702A	SETKT702A	LOAD0702A	GOTEST702A	GOBUT701D	SCOPE711	SCOPE712
5450	GOTEST712B	SCOPE713	EXPEC713B	GOTEST713B	SCOPE721	SCOPE710	LOAD0722A	SCOPE712
5460			LOADRET720A	TEST722A	LOADRET713A	TEST720A	LOADRET712A	TEST713A
5470	LOADRET711A	TEST712A	LOADRET710A	TEST711A	MASK702A	TEST710A	MASK701A	TEST702A
5500	TEST374D2	SCOPE372B	TEST374E2	TEST711A	TEST374F2	TEST712B	SETSP375A	TEST410
5510	LOAD01-610D1	TEST620A	TEST701D	LOAD0372B	ZER0374D1	EXPEC374E1	ZER0374F1	TEST375A
5520	LOAD0101A	TEST102A	UBRK621A	TEST376A	TEST374C2	TEST004	ZER0374E1	EXPEC374F1
5530	LOAD01-102A	TEST103A	LOADIR610A1	TEST701B	LOADIR610B1	TEST610C1	ZER0374A1	EXPEC374B1
5540	LOAD0104A	TEST105A	LOADSR105A	TEST610B1	LOADIR610C1	TEST610D1	LOAD16-103A	TEST104A
5550	LOAD047A	TEST050A	LOAD050A	TEST101A	TEST102D	CUA372B	ZER0374B1	EXPEC374C1
5560	LOAD046A	TEST047A	LOADIR624A	TEST701A	TEST102C	TEST624B	ZER0374C1	EXPEC374D1
5570	BCERC620A	TEST624A	TEST103B	TEST623	TEST102B	TEST622C	TEST103C	TEST622B
5600	TEST105B1	TEST622A	TEST610C2	TEST621F	TEST374A2	TEST621E	TEST374B2	TEST621D
5610	TEST105A1	TEST621C	TEST610B2	TEST621B	TEST376A1	ERROR621A	TEST713B	TEST621A
5620	TEST105E	TEST620C	TEST610A2	TEST620B	TEST610D2	TEST375B	GOTEST551C	TEST610A1
5630	LOAD045A	TEST046A	LOAD044A	TEST045A	LOAD043A	TEST044A	TEST105D	TEST043B

LOCN	-----0-----	-----1-----	-----2-----	-----3-----	-----4-----	-----5-----	-----6-----	-----7-----
5640	LOAD042A	TEST043A	TEST105C	TEST042B	LOAD041A	TEST042A	TEST104B	TEST041B
5650	LOAD040A	TEST041A	TEST105B	TEST040B	LOAD037A	TEST040A	LOAD036A	TEST037A
5660	LOAD035A	TEST036A	LOAD034A	TEST035A	LOAD033A	TEST034A	LOAD032A	TEST033A
5670	TEST047B	TEST032B	LOAD031A	TEST032A	TEST103D	TEST031B	LOAD030A	TEST031A
5700	LOAD027A	TEST030A	LOAD026A	TEST027A	LOAD025A	TEST026A	LOAD024A	TEST025A
5710	TEST014D	TEST024B	TEST030B	TEST030C	LOAD023A	TEST024A	LOAD022A	TEST023A
5720	LOAD021A	TEST022A	TEST015C	TEST021B	LOAD020A	TEST021A	TEST050B	TEST020C
5730	TEST047C	TEST020B	LOAD017A	TEST020A	TEST030D	TEST017B	LOAD016A	TEST017A
5740	LOAD015A	TEST016A	LOAD014A	TEST015A	LOAD013A	TEST014A	SDF105A	TEST013G
5750	TEST013F	TEST013E	TEST014B	TEST013D	TEST621H	TEST013C	TEST624D	TEST013B
5760	SETIRA	TEST013A	TEST014C	TEST012C	TEST012F	TEST012E	TEST621G	TEST012D
5770	TEST046B	TEST012C	TEST015B	TEST012B	TEST021C	TEST012A	TEST624C	TEST011

LOCN	-----0-----	-----1-----	-----2-----	-----3-----	-----4-----	-----5-----	-----6-----	-----7-----
6000	SETBYTEE410	LOADRES361A	LOADSR361A	EXPEC361B	COMP361B	GOBUT361D	GOBUT361E	SETRES362A
6010	SETDOC362A	LOADRES362A	COMP362A	GOBUT362A	EXPEC362B	COMP362B	GOBUT362D	GOBUT362E
6020	SETDOC363A	SHIFT363A	COMP363A	GOBUT363A	GOBUT363B	SETDOC364A	GOBUT364A	SETDOC364B
6030	GOBUT364B	SETRES365A	LOADRES365A	EXPEC365A	COMP365A	GOBUT365B	SETRES366A	LOADRES366A
6040	SHIFT365A	EXPEC366A	COMP366A	GOBUT366B	SETDOC366C	SHIFT366C	EXPEC366C	COMP366C
6050	SETRES367A	LOADRES367A	LOADSR367A	GOBUT367B	SETRES370A	LOADRES370A	SHIFT370A	COMP370A
6060	GOBUT370B	GOBUT370B	SETDOC370C	SHIFT370C	COMP370C	GOBUT370C	GOBUT370D	SETRES371A
6070	LOADRES371A	SHIFT371A	COMP371A	GOBUT371A	GOBUT371B	SCOPE371	SETDOC372A	SUBRA372A
6100	PAGE372B	GOBUT373B	SCOPE373	LOADRES374	LOADDCS506A	SETBYTEG410	COMP410A	INTOIR410A
6110	EXPEC410B	COMP410B	GOBUT410B	EXPEC410C	ASIDE410C	COMP410C	GOBUT410C	EXPEC410D
6120	COMP410D	GOBUT410D	EXPEC410E	COMP410E	GOBUT410E	SCOPE410	SETFETCHD500	SETFETCHES00
6130	SETFETCHF500	SETFETCHG500	LOADIR500	TESTINHSP500	COMP500A	GOBUT500B	EXPEC500C	COMP500C
6140	GOBUT500C	COMP500D	GOBUT500D	EXPEC500E	COMP500E	GOBUT500E	SCOPE500F	LOADFPS503A
6150	EXPEC503A	MASK503A	LOADFC503A	DOFC503A	GOBUT503A	GOBUT503B	GOBUT503B	GOBUT503C
6160	GOBUT503D	LOADIR503D	GOBUT503D	EXPEC503E	GOBUT503E	GOBUT503F	GOBUT503G	GOBUT503H
6170	GOBUT503I	GOBUT503J	GOBUT503K	SCOPE503	LOADFPS504A	EXPEC504A	LOADFC504A	DOFC504A
6200	GOBUT504A	GOBUT504A	GOBUT504B	GOBUT504C	LOADIR504D	GOBUT504D	EXPEC504E	GOBUT504E
6210	GOBUT504F	GOBUT504G	GOBUT504H	GOBUT504I	SCOPE504	EXPEC505A	SETFLAG505A	BUTCLRS05A
6220	GOBUT505A	GOBUT505B	CUA372A	GOBUT505C	GOBUT505	SETUCONS06A	PSH1506A	PSH1506A
6230	PSL0506A	FUDGEPS506A	MFS01	GOBUT506B	GOBUT506C	GOBUT506D	CLEAR506E	GOBUT506E
6240	SCOPE506	PSH1507A	PSH1507A	PSL0507A	FUDGEPS507A	LOADDCS507A	GOBUT507A	GOBUT507B
6250	GOBUT507C	GOBUT507D	TEST002	GOBUT507E	CINS07F	UBRK624A	SETPS507F	SETSR507F
6260	SETCINS07F	GOBUT507F	SCOPE507	LOADPRI0510A	GOBUT510A	GOBUT510B	GOBUT510C	FLAGS10D
6270	ENFLAGS10D	SETFLAGS10D	GOBUT510D	GOBUT510DA	ZERO0510E	GOBUT510E	ZEROFLAGS10E	GOBUT510F
6300	SCOPE510	SCOPE511A	MFS002	MFS003	MFS004	SCOPE511B	SETD624A	SETUP512B
6310	SETPSFLAGS12A1	GOTESTS12A1	GOTESTS12A2	SCOPE512A	SETPSFLAGS12B1	GOTESTS12B1	GOTESTS12B2	SETIR512C1
6320	SETPSFLAGS12C1	GOTESTS12C1	GOTESTS12C2	SCOPE512C	SETPSFLAGS12D1	GOTESTS12D1	GOTESTS12D2	SETIR512E1
6330	SETPSFLAGS12E1	GOTESTS12E1	GOTESTS12E2	SCOPE512E	SETUP520A	GOTESTS20A	SETUP520B	GOTESTS20B
6340	SETUP520C	GOTESTS20C	SETUP520D	GOTESTS20D	SETUP520E	GOTESTS20E	SCOPE520	INITD533A
6350	COMP533A	GOBUT533A	INIT533	INITD533B	COMP533	GOBUT533B	SCOPE533B	COMP534A
6360	GOBUT534A	INITD534B	COMP534B	GOBUT534B	SCOPE534B	COMP534C	GOBUT534C	INITD534D
6370	COMP534D	GOBUT534D	SCOPE534D	INITD534E	COMP534E	GOBUT534E	BUTERROR6	NEXTPAT500
6400	LOADCNTR534F	INITD534F	COMP534F	GOBUT534F	COUNTER534G	GOBUT534G	SCOPE534G	EXPEC535A
6410	INITD535A	COMP535A	GOBUT535A	EXPEC535B	INITD535B	COMP535B	GOBUT535B	SCOPE535B
6420	EXPEC536A	INITD536A	COMP536A	GOBUT536A	EXPEC536B	INITD536B	COMP536B	GOBUT536B
6430	SCOPE536B	COMP536C	GOBUT536C	INIT536D	EXPEC536D	INITD536D	COMP536D	GOBUT536D
6440	SCOPE536D	INITD536E	COMP536E	GOBUT536E	LOADCR536F	INITD536F	COMP536F	GOBUT536F
6450	SCOPE536F	AR3-537A	EOP002	SETFLG624A	SETSH624A	EXPEC701C	LOADBA701C	GOTEST701C
6460	EXPEC702B	SETKT702B	LOADBA702B	GOTEST702B	GOTEST710B	EXPEC710C	GOTEST710C	EXPEC710D
6470	CLEAR710D	GOTEST710D	GOTEST710E	EXPEC711B	GOTEST711B	EXPEC711C	GOTEST711C	GOTEST711D
6500	GOBUT712C	EXPEC712D	GOTEST712D	GOBUT713C	EXPEC713D	GOTEST713D	GOTEST720B	.....
6510	EOP003	EOP004	EOP007	TEST712D	TEST712C	ASSERTFOV500	TEST711D	FIXPAT500
6520	TEST720B	TEST713D	TEST713C	DZER0410	TEST711B	TESTS12E2	SETIR512B1	TESTS12D1
6530	TEST711C	DNONZER0410	INITIALIZED01	ERROR010	TESTS12A2	MFSSEXPEC0	TESTS12B2	MFSSEXPEC1
6540	LOOP372A	TEST373A	TESTS12A1	MFS005	GOTESTS11B1	SETUP512A	TESTS12C1	TESTS07D
6550	GOTESTS11A1	TESTS11B1	MFS005	RETURN373A	SETIR512A1	TESTS12B1	TESTS12C2	ERROR624A
6560	SETBYTEE410	TEST500	RETURN373A	TEST374	TEST710D	TESTS12B1	TESTS12D2	TESTS12E1
6570	SETRES361A	TEST372A	TEST373B	TEST361A	TESTS10E	TESTS06E	TESTS12D2	TESTS11A
6600	TESTS10D	TEST503DA	EXPEC507A	TESTS10A	TESTS10E	TESTS04A	SETONE510A	TESTS10DA
6610	TESTS10F	TEST507C	EXPEC506A	TESTS07A	LOADFLAG503A	TESTS04A	TESTS10C	TESTS10DA
6620	LOAD505A	TEST506A	LOADFLAG504A	TESTS05A	UCONS20A	TESTS33A	SETIR512D1	TESTS20A
6630	TEST503B	TEST003	TEST503C	TESTS33B	TESTS34B	TESTS34D	TESTS34F	TESTS35B

LOCN	-----0-----	-----1-----	-----2-----	-----3-----	-----4-----	-----5-----	-----6-----	-----7-----
6640	TEST536B	TEST536D	TEST536F	NEWCTR537A	TEST710B	TEST710C	TEST503K	TEST507F
6650	LOAD536E	SETEMIT537A	INITD536C	TEST536E	INIT536A	TEST536C	INIT535A	TEST536A
6660	LOADCNTR534E	TEST535A	INITD534C	TEST534E	INITD534A	TEST534C	INIT533A	TEST534A
6670	TEST500D	TESTD500	TEST504F	TEST500A	TEST507E	TEST510B	SEIFETCH8500	TEST503A
6700	TEST503E	TEST372B	TEST504G	LOAD372A	TEST410B	TEST410E	TEST504H	TEST410C
6710	TEST503F	TEST410A	TEST505B	TEST371B	TESTA503A	TEST371A	TEST504I	TEST370D
6720	TEST503G	TEST370C	TEST505C	TEST370B	TEST507B	TEST370A	TEST500C	TEST410D
6730	TEST504B	TEST367A	TEST503H	TEST366C	TEST506D	TEST366B	TEST500E	TEST366A
6740	TEST504C	TEST365B	TEST503I	TEST365A	TEST506C	TEST364B	TEST500D	TEST364A
6750	TEST504D	TEST363B	TEST534G	TEST363A	TEST506B	TEST362E	TEST503J	TEST362D
6760	TEST504E	TEST702B	TEST520D	TEST362B	TEST520B	TEST362A	TESTA504A	TEST361E
6770	TEST701C	TEST361D	TEST520C	TESTD410	VFY001	TEST520E	EOP006	TESTINHASP500



LOCN	0	1	2	3	4	5	6	7
7000	SUM10	INITIALIZED03	SETUPCSP16A	INITIALIZED04	INITIALIZED05	INITIALIZED06	INITIALIZED07	ALUCARRY1
7010	SETUPCSP15A	SETUPCSP14A	SETUPCSP12A	SETUPCSP05A	SETUPCSP07A	SETUPCSP00A	WRITESF350	ALUCARRY1A
7020	ALUCARRY2	ALUCARRY2A	RESETIR350A	NEXTPAT350	LOADHI350	AGAINSR0350	RESETIR350B	RESETIR350C
7030	FASTC2	CSP1E RTORSE	STORED03	RESTORE04	SETIR373A	CSP16XONFLTOIR	ZEROSF020F04	SETSR373A
7040	SFDF03A	ZEROJ0JF02	ZEROSFDF	ZEROSF0FA	ZEROSF	ZEROSFA	FIRST375A	SFDF0SRA
7050	SFDF0S4B	SFDF0S1C	SFDF0SRO	SFDF0SRE	SFDF0SRF	SFDF0SRG	SFDF0SRH	SFDF0SRI
7060	SETBYTEC410	STR00375A	BYTE375A	CHECK375A	GOBUT375A	NEXTPAT410	SETBYTED410	LOADUCON506A
7070	BYTEFIRST410	G001537A	DOITS06A	FLAGPSS0100	TEST511A2	FLAGPSS02	FLAGPSS03	FLAGPSS04
7100	PSS0100	P000002	PSS010003	PSS010004	PSS010005	PSS010006	TEST511B2	GOTEST511A2
7110	TEST511A3	G01511A3	TEST511A4	GOTEST511A4	KTSRCOST01	GOTEST511B2	TEST511B3	GOTEST511B3
7120	TEST511B4	GOTEST511B4	SUMTEST01	KTSRCOST02	KTSRCOST03	KTSRCOST04	KTSRCOST04B	KTSRCOST07
7130	KTSRCOST08	INIT537A	SUBTEST02	SUBTEST03	BR3-537A	SETRESA537A	BUTCOUNT-1S-377CR3-537A	BR3-537A
7140	CR3-537A	AR7-537A	CR7-537A	DR7-537A	DR7-537A	AR11-537A	BR11-537A	CR11-537A
7150	AL1-537A	ER3-537A	CR3-537A	NR3-537A	AL1-537A	SETUPPSCC#00C	GOTEST551B1	GOTEST551B1
7160	W011621H	SETUPPSCC#00C02	PSCC0SR3-0	PSCC0SR3-0AA	PSCC0SR3-0BB	CLEAR624	EXPEC701A	SFCN713A
7170	LOAD701A	GOTEST701A	LOADIR720A	SETADR721A	SETADR720A	SETJAM720A	BUSFCN720A	LX12
7200	SETJAM721A	BUSFCN721A	M#K762A	C17X12	BOX05	C17X05	SETPRI752C	PHI06762A
7210	GETIT762A	EQ#005	EOP10	JAMUPP002D	INSERT0PREVNO	INSERT02	INSERTREVNO	INSERT03
7220	INSERT04	DISPLAY	DISP002	DISP003	DISP004	DISP005	DISP006	CLEAR-I-0-A
7230	CLEAR-I-0-B	CLEARI002	CLEARI004	CLEARI005	D115-121	D1512A	D111-061	D1106A
7240	D105-001	D0500A	DZERO	DTOIRB	CLRSERVICET00	DATISERVICET00	DATOSERVICET00	CJESERVICET00
7250	SERVICET00	PE#T00	FLAGPST00	PST00	ODDJAMT00	CLRJAMT00	JAMT00	CUAT00
7260	GETMSKPROC DAT	MSKPROC DAT	CHMPROC DAT	RESETPROC DAT	LOADFPSCC	LOADFPSCC02	SRINTOIR	DINTOIR
7270	DUFINTOIR	RESE TUONP	SRINTOIRS	DINTOIRS	DUFINTOIRS	RESE TUONPS	BUTSR3-0	BTBGFPSERV
7300	BUTIR15-12	BUTINSTRS	BUTIR11FLTPT3-0NEXT007	DINTOIR350	BUTIR9-6	BUTMOVDR7IRS-3	BUTINSTR1	BUTMGPSERV
7310	EUTD(C)C	BUTOUT7DOUT7	BUTFPS05	EXPEC410A	BUTGD3-2	BUTSR1-0	BUTBGSEVL	BUTHMASKPS(T)
7320	BUTHNXT000	BUTH000	BUTHNXT001	GETPROC DAT	BUTHNXT002	BUTHFLAG7	BUTHNXT003	BUTHMEXFLAG1
7330	BUTHNXT004	BUTHFLTPTS	BUTHNXT005	VFY006	BUTHNXT006	KTSRCOST05	BUTHINITJAM	KTSRCOST06
7340	BUTHNXT007	BUTD-1S-ZERO	BUTIR11B	VFY005	BUTD(C)A	BUTSERVICE	BUTVECTLOAD	BUTDR6-7L
7350	BUTHNXT008	BUTB#00	BUTOTHERJAM	VFY006	BUTFPPROC	BUTINTRHIGH	BUTINSTRBRANCH	LOADHI350
7360	BUTPREFETCHJAM	GOFOR410	NEXT721A	VFY006	NEXT720A	NEXT713A	BUTINSTRBRANCH	BYTESECOND410
7370	SETRESA537A	BUTPF0		VFY006	VFY006	VFY005	BUTERR07	
7400	ZTARGET400	ZTARGET401	ZTARGET402	ZTARGET403	ZTARGET404	ZTARGET405	ZTARGET406	ZTARGET407
7410	ZTARGET410	ZTARGET411	ZTARGET412	ZTARGET413	ZTARGET414	ZTARGET415	ZTARGET416	ZTARGET417
7420	ZTARGET420	ZTARGET421	ZTARGET422	ZTARGET423	ZTARGET424	ZTARGET425	ZTARGET426	ZTARGET427
7430	ZTARGET430	ZTARGET431	ZTARGET432	ZTARGET433	ZTARGET434	ZTARGET435	ZTARGET436	ZTARGET437
7440	ZTARGET440	ZTARGET441	ZTARGET442	ZTARGET443	ZTARGET444	ZTARGET445	ZTARGET446	ZTARGET447
7450	ZTARGET450	ZTARGET451	ZTARGET452	ZTARGET453	ZTARGET454	ZTARGET455	ZTARGET456	ZTARGET457
7460	ZTARGET460	ZTARGET461	ZTARGET462	ZTARGET463	ZTARGET464	ZTARGET465	ZTARGET466	ZTARGET467
7470	ZTARGET470	ZTARGET471	ZTARGET472	ZTARGET473	ZTARGET474	ZTARGET475	ZTARGET476	ZTARGET477
7500	ZTARGET500	ZTARGET501	ZTARGET502	ZTARGET503	ZTARGET504	ZTARGET505	ZTARGET506	ZTARGET507
7510	ZTARGET510	ZTARGET511	ZTARGET512	ZTARGET513	ZTARGET514	ZTARGET515	ZTARGET516	ZTARGET517
7520	ZTARGET520	ZTARGET521	ZTARGET522	ZTARGET523	ZTARGET524	ZTARGET525	ZTARGET526	ZTARGET527
7530	ZTARGET530	ZTARGET531	ZTARGET532	ZTARGET533	ZTARGET534	ZTARGET535	ZTARGET536	ZTARGET537
7540	ZTARGET540	ZTARGET541	ZTARGET542	ZTARGET543	ZTARGET544	ZTARGET545	ZTARGET546	ZTARGET547
7550	ZTARGET550	ZTARGET551	ZTARGET552	ZTARGET553	ZTARGET554	ZTARGET555	ZTARGET556	ZTARGET557
7560	ZTARGET560	ZTARGET561	ZTARGET562	ZTARGET563	ZTARGET564	ZTARGET565	ZTARGET566	ZTARGET567
7570	ZTARGET570	ZTARGET571	ZTARGET572	ZTARGET573	ZTARGET574	ZTARGET575	ZTARGET576	ZTARGET577
7600	ZTARGET600	ZTARGET601	ZTARGET602	ZTARGET603	ZTARGET604	ZTARGET605	ZTARGET606	ZTARGET607
7610	ZTARGET610	ZTARGET611	ZTARGET612	ZTARGET613	ZTARGET614	ZTARGET615	ZTARGET616	ZTARGET617
7620	ZTARGET620	ZTARGET621	ZTARGET622	ZTARGET623	ZTARGET624	ZTARGET625	ZTARGET626	ZTARGET627
7630	ZTARGET630	ZTARGET631	ZTARGET632	ZTARGET633	ZTARGET634	ZTARGET635	ZTARGET636	ZTARGET637

LOCN	-----0-----	-----1-----	-----2-----	-----3-----	-----4-----	-----5-----	-----6-----	-----7-----
7640	ZTARGET640	ZTARGET641	ZTARGET642	ZTARGET643	ZTARGET644	ZTARGET645	ZTARGET646	ZTARGET647
7650	ZTARGET650	ZTARGET651	ZTARGET652	ZTARGET653	ZTARGET654	ZTARGET655	ZTARGET656	ZTARGET657
7660	ZTARGET660	ZTARGET661	ZTARGET662	ZTARGET663	ZTARGET664	ZTARGET665	ZTARGET666	ZTARGET667
7670	ZTARGET670	ZTARGET671	ZTARGET672	ZTARGET673	ZTARGET674	ZTARGET675	ZTARGET676	ZTARGET677
7700	ZTARGET700	ZTARGET701	ZTARGET702	ZTARGET703	ZTARGET704	ZTARGET705	ZTARGET706	ZTARGET707
7710	ZTARGET710	ZTARGET711	ZTARGET712	ZTARGET713	ZTARGET714	ZTARGET715	ZTARGET716	ZTARGET717
7720	ZTARGET720	ZTARGET721	ZTARGET722	ZTARGET723	ZTARGET724	ZTARGET725	ZTARGET726	ZTARGET727
7730	ZTARGET730	ZTARGET731	ZTARGET732	ZTARGET733	ZTARGET734	ZTARGET735	ZTARGET736	ZTARGET737
7740	ZTARGET740	ZTARGET741	ZTARGET742	ZTARGET743	ZTARGET744	ZTARGET745	ZTARGET746	ZTARGET747
7750	ZTARGET750	ZTARGET751	ZTARGET752	ZTARGET753	ZTARGET754	ZTARGET755	ZTARGET756	ZTARGET757
7760	ZTARGET760	ZTARGET761	ZTARGET762	ZTARGET763	ZTARGET764	ZTARGET765	ZTARGET766	ZTARGET767
7770	ZTARGET770	ZTARGET771	ZTARGET772	ZTARGET773	ZTARGET774	ZTARGET775	ZTARGET776	ZTARGET777

PAGE	---USED---	---OPEN---
	OCTAL/DEC.	OCTAL/DEC.
4	773/507	5/ 5
5	774/508	4/ 4
6	772/506	6/ 6
7	775/509	3/ 3
TOTAL	3756/2030	22/ 18