SEQ 0001

```
1
2
3
            .REM _
```

## IDENTIFICATION

PRODUCT CODE: AC-E935B-MC

PRODUCT NAME: CXLKAB0 LK11 MODULE

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

## 1. ABSTRACT

"LKA" IS AN "IOMOD" THAT EXERCISES ONE LK11 PUSH BUTTON INTERFACE.
INCLUDED IN THIS MODULE IS A LOGIC TEST OF THE INTERFACE.
AFTER THE LOGIC TEST HAS BEEN PREFORMED, THE MODULE WILL
LOAD A SEQUENCE OF DIFFERENT PATTERN'S INTO THE LAMP DATA REGISTER.
THESE DIFFERENT PATTERNS WILL AID IN DETECTING INOPERATIVE LAMPS IN THE BUTTON BOX.
IF THE OPERATOR DEPRESSES A BUTTON AFTER THE MODULE HAS BEEN RUNNING,
THE MODULE WILL ENTER A "COMPLEMENT LAST SWITCH VALUE" LOOP. THIS LOOP
WILL BE EXITED IF NO BUTTON IS DEPRESSED FOR 10 SECONDS TO THE
LAMP DATA TESTING.

## 2. REQUIREMENTS

HARDWARE: LK11 PUSH BUTTON INTERFACE AND BUTTON BOX.

STORAGE:: LKA REQUIRES:
1. DECIMAL WORDS: 1007
2. OCTAL WORDS: 01757
3. OCTAL BYTES: 3736

## 3. PASS DEFINITION

ONE PASS OF LKA MODULE CONSISTS OF ONE ITERATION OF THE EIGHT
DIFFERENT LIGHT PATTERNS.

## 4. EXECUTION TIME

LKA RUNNING ALONE ON PDP-11/05 TAKES
APPROXIMATELY ONE MINUTE.

SEQ 0003

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 160060, VECTOR: 360, BR1: 4

REQUIRED PARAMETERS:

NONE

6. DEVICE/OPTION SETUP

THE LK-11-A INTERFACE AND BUTTON BOX MUST BE CONNECTED.

7. MODULE OPERATION

THE MODULE WILL BEGIN BY TESTING THE ABLILTY OF THE LDR (LAMP DATA REGISTER)
PBR (PUSH BUTTON REGISTER) AND THE PBSR (PUSH BUTTON CONTROL/STATUS)
REGISTER TO FUNCTION PROPERLY.

THE MODULE WILL NOW START TESTING THE LAMPS IN THE PUSH BUTTON BOX
BY LOADING DIFFERENT PATTERNS INTO THE LAMP DATA REGISTER.

8. OPERATION OPTIONS

NONE

9. NON STANDARD PRINTOUTS

ALL PRINTOUTS HAVE STANDARD MEANINGS AS REPRESENTED IN
DEC/X11 DOCUMENTATION.

10. ENVIROMENT

#1 11/10 WITH 16K OF MEMORY
RK-11-D DISK CONTROLLER WITH 1 DRIVE
LK11 PUSH BUTTON OPTION

#2 11/45 WITH 24K OF MEMORY (16K CORE + 8K MOS)
KT-11-D MEMORY MANAGEMENT
RK-11-D DISK CONTROLLER WITH 1 DRIVE
LK11 PUSH BUTTON OPTION



.LIST SEQ,BIN

```
                        IOMOD   <LKAB >,160060,360,4,,,79.,101
                        MODULE  140000,LKAB ,160060,360,4,,,79.,101
                        .TITLE  LKAB DEC/X11 SYSTEM EXERCISER MODULE
                ;       DDXCOM  VERSION  6      23-MAY-78
                        .LIST   BIN
                ;*****************************************************************************
000000'         BEGIN:
000000' 045514  MODNAM: .ASCII  /LKAB / ;MODULE NAME.
000005'    000  XFLAG:  .BYTE   OPEN              ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 160060  ADDR:   160060+0                 ;1ST DEVICE ADDR.
000010' 000360  VECTOR: 360+0                    ;1ST DEVICE VECTOR.
000012'    200  BR1:    .BYTE   PRTY4+0         ;1ST BR LEVEL.
000013'    000  BR2:    .BYTE   PRTY+0          ;2ND BR LEVEL.
000014' 000001  DVID1:  +1                       ;DEVICE INDICATOR 1.
000016' 000000  SR1:    OPEN                     ;SWITCH REGISTER 1
000020' 000000  SR2:    OPEN                     ;SWITCH REGISTER 2
000022' 000000  SR3:    OPEN                     ;SWITCH REGISTER 3
000024' 000000  SR4:    OPEN                     ;SWITCH REGISTER 4
                ;*****************************************************************************
000026' 140000  STAT:   140000                   ;STATUS WORD.
000030' 000276' INIT:   START                    ;MODULE START ADDR.
000032' 000224' SPOINT: MODSP                    ;MODULE STACK POINTER.
000034' 000000  PASCNT: 0                        ;PASS COUNTER.
000036' 000117  ICONT:  79.                      ;# OF ITERATIONS PER PASS=79.
000040' 000000  ICOUNT: 0                        ;LOC TO COUNT ITERATIONS
000042' 000000  SOFCNT: 0                        ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000  HRDCNT: 0                        ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000  SOFPAS: 0                        ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000  HRDPAS: 0                        ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000  SYSCNT: 0                        ;# OF SYS ERRORS ACCUMULATED
000054' 000000  RANNUM: 0                        ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'         CONFIG:                          ;RESERVED FOR MONITOR USE
000056' 000000  RES1:   0                        ;RESERVED FOR MONITOR USE
000060' 000000  RES2:   0                        ;RESERVED FOR MONITOR USE
000062' 000000  SVR0:   OPEN                     ;LOC TO SAVE R0.
000064' 000000  SVR1:   OPEN                     ;LOC TO SAVE R1.
000066' 000000  SVR2:   OPEN                     ;LOC TO SAVE R2.
000070' 000000  SVR3:   OPEN                     ;LOC TO SAVE R3.
000072' 000000  SVR4:   OPEN                     ;LOC TO SAVE R4.
000074' 000000  SVR5:   OPEN                     ;LOC TO SAVE R5.
000076' 000000  SVR6:   OPEN                     ;LOC TO SAVE R6.
000100' 000000  CSRA:   OPEN                     ;ADDR OF CURRENT CSR.
000102'         SBADR:                           ;ADDR OF GOOD DATA, OR
000102' 000000  ACSR:   OPEN                     ;CONTENTS OF CSR.
000104'         WASADR:                          ;ADDR OF BAD DATA, OR
000104' 000000  ASTAT:  OPEN                     ;STATUS REG CONTENTS.
000106'         ERRTYP:                          ;TYPE OF ERROR
000106' 000000  ASD:    OPEN                     ;EXPECTED DATA.
000110' 000000  AWAS:   OPEN                     ;ACTUAL DATA.
000112' 000320' RSTRT:  RESTRT                   ;RESTART ADDRESS AFTER END OF PASS
000114' 000000  WDTO:   OPEN                     ;WORDS TO MEMORY PER ITERATION
000116' 000000  WDFR:   OPEN                     ;WORDS FROM MEMORY PER ITERATION
000120' 000000  INTR:   OPEN                     ;# OF INTERRUPTS PER ITERATION
000122' 000101  IDNUM:  101                      ;MODULE IDENTIFICATION NUMBER=101
        000040          .REPT   SPSIZ            ;MODULE STACK STARTS HERE.
                        .NLIST
```



```
                        .WORD   0
                        .LIST
                        .ENDR
000224'         MODSP:
                ;*****************************************************************************
```

```
                        ;ADDRESSES AND VECTORS
                        ;
000224' 000006'         PBSR:   ADDR                    ;PUSH BUTTON STATUS REGISTER
000226' 000010'         PBR:    ADDR+2                  ;PUSH BUTTON DATA REGISTER
000230' 000012'         LDR:    ADDR+4                  ;LAMP DATA REGISTER
                        ;
000232' 000010'         VECT:   VECTOR                  ;LK-11-A INTERRUPT VECTOR
000234' 000012'         VECTR1: VECTOR+2
                        ;
                        ;TEMP LOCATIONS
                        ;
000236' 000000          BITSAV: 0
000240' 000000          TEMP:   0
000242' 000000          TEMP1:  0
000244' 000000          TEMP2:  0
000246' 000000          TEMPL:  0
000250'                 UNEXPT:
                        ;-------------------------------------------------------------------
000250' 000004  000000' 000256'         $IRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                        ;-------------------------------------------------------------------
000256' 012767  000011  177622  1$:     MOV     #11,ERRTYP              ;ILLEGAL INTERRUPT
                        ;*****************************************************************
000264' 104405  000000' 000000          $RDERR$,BEGIN,NULL      ;UNEXPECTED INTERRUPT FROM LK-11-A
                        ;*****************************************************************
000272' 104410  000000'                 END$,BEGIN              ;

000276' 012767  000001  177614  START:  MOV     #1,INTR         ;1 INTERRUPT/ITERATION
000304' 012767  000031  177604          MOV     #25.,WDFR       ;25 WORDS FROM MEM/ITERATION
000312' 012767  000031  177574          MOV     #25.,WDTO       ;25 WORDS TO MEM/ITERATION
000320' 000240          RESTRT: NOP
000322' 016767  177460  177674          MOV     ADDR,PBSR               ;LOAD THE LK-11 ADDRESSES
000330' 016767  177452  177672          MOV     ADDR,LDR
000336' 016767  177444  177662          MOV     ADDR,PBR
000344' 062767  000002  177654          ADD     #2,PBR
000352' 062767  000004  177650          ADD     #4,LDR

000360' 016767  177424  177644          MOV     VECTOR,VECT             ;LOAD THE LK-11 VECTOR
000366' 016767  177416  177640          MOV     VECTOR,VECTR1
000374' 062767  000002  177632          ADD     #2,VECTR1
000402' 012777  000250' 177622          MOV     #UNEXPT,@VECT           ;LOAD FALSE INTR. RETURN
000410' 005077  177620                  CLR     @VECTR1
```



```
                        ;LOAD ALL 1'S AND READ BACK LDR
000414' 012767  177777  177462  LD1:    MOV     #-1,ASTAT               ;LOAD EXPECTED DATA
000422' 016777  177456  177600          MOV     ASTAT,@LDR              ;LOAD ALL ONES INTO LDR
000430' 017767  177574  177444          MOV     @LDR,ACSR               ;READ ACTUAL DATA
000436' 026767  177442  177436          CMP     ASTAT,ACSR              ;COMPARE RESULTS
000444' 001406                          BEQ     LD25                    ;BRANCH IF EQUAL
000446' 012767  000025  177432          MOV     #25,ERRTYP      ;BIT STUCK
                        ;*****************************************************************
000454' 104405  000000' 000000          $RDERR$,BEGIN,NULL      ;FAILED TO LOAD ALL ONES
                        ;*****************************************************************

                        ;LOAD AND READ '25' PATTERN INTO LDR
000462' 012767  125252  177414  LD25:   MOV     #125252,ASTAT           ;LOAD EXPECTED DATA
000470' 016777  177410  177532          MOV     ASTAT,@LDR              ;LOAD PATTERN INTO LDR
000476' 017767  177526  177376          MOV     @LDR,ACSR               ;READ ACTUAL DATA
000504' 026767  177374  177370          CMP     ASTAT,ACSR              ;COMPARE RESULTS
000512' 001406                          BEQ     LD52                    ;BRANCH IF EQUAL
000514' 012767  000025  177364          MOV     #25,ERRTYP      ;BIT STUCK
                        ;*****************************************************************
000522' 104405  000000' 000000          $RDERR$,BEGIN,NULL      ;FAILED TO LOAD PATTERN
                        ;*****************************************************************

                        ;LOAD AND READ '52' PATTERN INTO LDR
000530' 012767  052525  177346  LD52:   MOV     #52525,ASTAT            ;LOAD EXPECTED DATA
000536' 016777  177342  177464          MOV     ASTAT,@LDR              ;LOAD PATTERN INTO LDR
000544' 017767  177460  177330          MOV     @LDR,ACSR               ;READ ACTUAL DATA
000552' 026767  177326  177322          CMP     ASTAT,ACSR              ;COMPARE RESULTS
000560' 001406                          BEQ     FLT1                    ;BRANCH IF EQUAL
000562' 012767  000025  177316          MOV     #25,ERRTYP      ;BIT STUCK
                        ;*****************************************************************
000570' 104405  000000' 000000          $RDERR$,BEGIN,NULL      ;FAILED TO LOAD PATTERN
                        ;*****************************************************************

                        ;FLOAT A ONE THROUGH LDR
000576' 012767  000001  177300  FLT1:   MOV     #BIT0,ASTAT             ;LOAD EXPECTED DATA
000604' 016777  177274  177416  2$:     MOV     ASTAT,@LDR              ;LOAD LDR BIT
000612' 017767  177412  177262          MOV     @LDR,ACSR               ;READ ACTUAL DATA
000620' 026767  177260  177254          CMP     ASTAT,ACSR              ;COMPARE RESULTS
000626' 001406                          BEQ     1$                      ;BRANCH IF EQUAL
000630' 012767  000025  177250          MOV     #25,ERRTYP      ;BIT STUCK
                        ;*****************************************************************
000636' 104405  000000' 000000          $RDERR$,BEGIN,NULL      ;FAILED TO SET LDR BIT
                        ;*****************************************************************
000644' 006167  177234          1$:     ROL     ASTAT                   ;TEST NEXT BIT
000650' 103355                  BCC     2$                      ;CONTINUE THROUGH LDR
```

```
                        ;FLOAT ZERO THROUGH LDR
000652' 012767  177776  177224   FLT0:   MOV     #177776,ASTAT           ;LOAD EXPECTED DATA
000660' 016777  177220  177342   2$:     MOV     ASTAT,@LDR              ;LOAD LDR BIT
000666' 017767  177336  177206           MOV     @LDR,ACSR               ;READ ACTUAL DATA
000674' 026767  177204  177200           CMP     ASTAT,ACSR              ;COMPARE RESULTS
000702' 001406                           BEQ     1$                      ;BRANCH IF EQUAL
000704' 012767  000025  177174           MOV     #25,ERRTYP      ;BIT STUCK
                                         ;*****************************************************************
000712' 104405  000000' 000000           HRDERS,BEGIN,NULL       ;FAILED TO CLEAR LDR BIT
                                         ;*****************************************************************
000720' 006167  177160           1$:     ROL     ASTAT                   ;TEST NEXT BIT
000724' 103355                           BCC     2$                      ;CONTINUE THROUGH LDR

                        ;LOAD AND READ PBR USING BIT 0
000726' 012767  177777  177150           MOV     #-1,ASTAT               ;LOAD EXPECTED DATA
000734' 012777  000001  177264           MOV     #BIT0,@PBR              ;SET BIT 0
000742' 017767  177260  177132           MOV     @PBR,ACSR               ;READ ACTUAL DATA
000750' 026767  177130  177124           CMP     ASTAT,ACSR              ;COMPARE RESULTS
000756' 001406                           BEQ     CLBIT0                  ;BRANCH IF EQUAL
000760' 012767  000025  177120           MOV     #25,ERRTYP      ;BIT STUCK
                                         ;*****************************************************************
000766' 104405  000000' 000000           HRDERS,BEGIN,NULL       ;FAILED TO SET PBR
                                         ;*****************************************************************

                        ;CLEAR PBR USING BIT 0
000774' 005067  177104   CLBIT0: CLR     ASTAT                   ;EXPECTED DATA
001000' 012777  177776  177220           MOV     #177776,@PBR            ;CLEAR BIT 0
001006' 017767  177214  177066           MOV     @PBR,ACSR               ;READ ACTUAL DATA
001014' 026767  177064  177060           CMP     ASTAT,ACSR              ;COMPARE RESULTS
001022' 001406                           BEQ     LDPB                    ;BRANCH IF EQUAL
001024' 012767  000025  177054           MOV     #25,ERRTYP      ;BIT STUCK
                                         ;*****************************************************************
001032' 104405  000000' 000000           HRDERS,BEGIN,NULL       ;FAILED TO CLEAR PBR
                                         ;*****************************************************************

                        ;LOAD AND READ BACK PB FLAG
001040' 005077  177160   LDPB:   CLR     @PBSR                   ;CLEAR STATUS REGISTER
001044' 012767  100000  177032           MOV     #BIT15,ASTAT            ;LOAD EXPECTED DATA
001052' 016777  177026  177144           MOV     ASTAT,@PBSR             ;LOAD STATUS REG.
001060' 017767  177140  177014           MOV     @PBSR,ACSR              ;READ ACTUAL DATA
001066' 026767  177012  177006           CMP     ASTAT,ACSR              ;COMPARE RESULTS
001074' 001406                           BEQ     PBCLR                   ;BRANCH IF EQUAL
001076' 012767  000025  177002           MOV     #25,ERRTYP      ;BIT STUCK
                                         ;*****************************************************************
001104' 104405  000000' 000000           HRDERS,BEGIN,NULL       ;FAILED TO SET BIT 15
                                         ;*****************************************************************
```



```
                        ;TEST PB FLAG CLEARED BY CLR PBSR
001112' 005077  177106   PBCLR:  CLR     @PBSR                   ;CLEAR STATUS REGISTER
001116' 005067  176762           CLR     ASTAT                   ;EXPECTED DATA
001122' 052777  100000  177074           BIS     #BIT15,@PBSR            ;SET PB FLAG
001130' 005077  177070           CLR     @PBSR                   ;CLAR STATUS REGISTER
001134' 017767  177064  176740           MOV     @PBSR,ACSR              ;READ ACTUAL DATA
001142' 026767  176736  176732           CMP     ASTAT,ACSR              ;COMPARE RESULTS
001150' 001406                           BEQ     PBCLRH                  ;BRANCH IF EQUAL
001152' 012767  000025  176726           MOV     #25,ERRTYP      ;BIT STUCK
                                         ;*****************************************************************
001160' 104405  000000' 000000           HRDERS,BEGIN,NULL       ;FAILED TO CLEAR PB FLAG
                                         ;*****************************************************************

                        ;TEST PB FLAG CLEARED WHEN READ PBR H
001166' 005077  177032   PBCLRH: CLR     @PBSR                   ;CLEAR STATUS REGISTER
001172' 005067  176706           CLR     ASTAT                   ;LOAD EXPECTED DATA
001176' 012777  100000  177020           MOV     #BIT15,@PBSR            ;SET PB FLAG
001204' 017767  177016  177026           MOV     @PBR,TEMP               ;READ PB REGISTER
001212' 017767  177006  176662           MOV     @PBSR,ACSR              ;READ ACTUAL DATA
001220' 026767  176660  176654           CMP     ASTAT,ACSR              ;COMPARE RESULTS
001226' 001406                           BEQ     PBCLRD                  ;BRANCH IF EQUAL
001230' 012767  000025  176650           MOV     #25,ERRTYP      ;BIT STUCK
                                         ;*****************************************************************
001236' 104405  000000' 000000           HRDERS,BEGIN,NULL       ;FAILED TO CLEAR PB FLAG
                                         ;*****************************************************************

                        ;TEST  PB FLAG IS CLEARED BY READ STATUS WHEN INTR ENARLE IS CLEARED
001244' 005077  176754   PBCLRD: CLR     @PBSR                   ;CLEAR STATUS REG
001250' 005067  176630           CLR     ASTAT                   ;EXPECTED DATA
001254' 012777  100000  176742           MOV     #BIT15,@PBSR            ;SET PB FLAG
001262' 017767  176736  176750           MOV     @PBSR,TEMP              ;READ SR
001270' 017767  176730  176604           MOV     @PBSR,ACSR              ;READ ACTUAL DATA
001276' 026767  176602  176576           CMP     ASTAT,ACSR              ;COMPARE RESULTS
001304' 001406                           BEQ     LDIE                    ;BRANCH IF EQUAL
001306' 012767  000025  176572           MOV     #25,ERRTYP      ;BIT STUCK
                                         ;*****************************************************************
001314' 104405  000000' 000000           HRDERS,BEGIN,NULL       ;READ SR FAILED TO CLEAR PB FLAG
                                         ;*****************************************************************

                        ;LOAD AND READ INTERRUPT ENABLE BIT,ALSO UNEXPECTED INTRPT TEST
001322' 005077  176676   LDIE:   CLR     @PBSR                   ;CLEAR PB STATUS REGISTER
001326' 005077  176674           CLR     @PBR                    ;CLEAR BUTTON REGISTER
001332' 012767  040000  176544           MOV     #BIT14,ASTAT            ;LOAD EXPECTED DATA
001340' 016777  176540  176656           MOV     ASTAT,@PBSR             ;LOAD STATUS REGISTER
001346' 017767  176652  176526           MOV     @PBSR,ACSR              ;READ ACTUAL DATA
001354' 026767  176524  176520           CMP     ASTAT,ACSP              ;COMPARE RESULTS
001362' 001406                           BEQ     TSBIT                   ;BRANCH IF EQUAL
001364' 012767  000025  176514           MOV     #25,ERRTYP      ;BIT STUCK
                                         ;*****************************************************************
001372' 104405  000000' 000000           HRDERS,BEGIN,NULL       ;FAILED TO LOAD INTERRUPT ENABLE
                                         ;*****************************************************************
```

```
                                ;TEST BITC INSTRUCTION CLEARS LDR
001400' 005077  176624          TSBITC: CLR     @LDR                    ;CLEAR LAMP DATA REGISTER
001404' 012767  000001  176624          MOV     #BIT0,BITSAV            ;INITIALIZE BITSAV
001412' 012777  177777  176610  2$:     MOV     #-1,@LDR                ;SET REIGSTER TO ALL 1'S
001420' 012767  177777  176456          MOV     #-1,ASTAT               ;EXPECTED
001426' 046767  176604  176450          BIC     BITSAV,ASTAT            ;DATA
001434' 046777  176576  176566          BIC     BITSAV,@LDR             ;CLEAR PBSR BIT
001442' 017767  176562  176432          MOV     @LDR,ACSR               ;READ ACTAL DATA
001450' 026767  176430  176424          CMP     ASTAT,ACSR              ;COMPARE RESULTS
001456' 001406                          BEQ     1$                      ;BRANCH IF EQUAL
001460' 012767  000025  176420          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
001466' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;FAILED TO CLEAR LDT BIT
                                        ;*****************************************************************
001474' 006167  176536          1$:     ROL     BITSAV                  ;SHIFT TO TEST NEXT BIT
001500' 103344                          BCC     2$                      ;BRANCH IF NOT DONE ALL BITS


                                ;TEST BITS INTRUCTION SETS LDR
001502' 012767  000001  176526  TSBITS: MOV     #BIT0,BITSAV            ;INITIAIALIZE BITSAV
001510' 005077  176514          2$:     CLR     @LDR                    ;CLEAR LAMP DATA REGISTER
001514' 016767  176516  176362          MOV     BITSAV,ASTAT            ;DATA
001522' 056777  176510  176500          BIS     BITSAV,@LDR             ;SET LDT BIT
001530' 017767  176474  176344          MOV     @LDR,ACSR               ;READ ACTUAL DATA
001536' 026767  176342  176336          CMP     ASTAT,ACSR              ;COMPARE RESULTS
001544' 001406                          BEQ     1$                      ;BRANCH IF EQUAL
001546' 012767  000025  176332          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
001554' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;FAILED TO SET LDR BIT
                                        ;*****************************************************************
001562' 006167  176450          1$:     ROL     BITSAV                  ;SHIFT TO TEST NEXT BIT
001566' 103350                          BCC     2$                      ;BRANCH BACK IF NOT DONE ALL BITS


                                ;TEST BISB INSTRUCTION SETS LDR
001570' 005077  176434  TSBISB: CLR     @LDR                    ;CLEAR LAMP DATA REGISTER
001574' 012777  052525  176426          MOV     #52525,@LDR             ;LOAD ALTERNATING 1'S IN LDR
001602' 012767  052777  176274          MOV     #52777,ASTAT            ;EXPECTED DATA
001610' 152777  177777  176412          BISB    #-1,@LDR                ;SET LOWER BYTE TO ALL 1'S
001616' 017767  176406  176256          MOV     @LDR,ACSR               ;READ ACTUAL DATA
001624' 026767  176254  176250          CMP     ASTAT,ACSR              ;COMPARE RESULTS
001632' 001406                          BEQ     1$                      ;BRANCH IF EQUAL
001634' 012767  000025  176244          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
001642' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;FAILED TO SET LOWER BYTE & LEAVE UPPER BYTE THE SAME
                                        ;*****************************************************************
001650' 012777  052525  176352  1$:     MOV     #52525,@LDR             ;LOAD ALTERNATING 1'S IN LDR
001656' 016767  176346  176362          MOV     LDR,TEMPL               ;GET ADDRESS OF LDR
001664' 005267  176356                  INC     TEMPL                   ;GET ADDRESS OF UPPER BYTE OF LDR
001670' 012767  177525  176206          MOV     #177525,ASTAT           ;EXPECTED DATA
001676' 152777  177777  176342          BISB    #-1,@TEMPL              ;SET UPPER BYTE TO ALL 1'S
001704' 017767  176320  176170          MOV     @LDR,ACSR               ;READ ACTUAL DATA
001712' 026767  176166  176162          CMP     ASTAT,ACSR              ;COMPARE RESULTS
001720' 001406                          BEQ     TSBICB                  ;BRANCH IF EQUAL
001722' 012767  000025  176156          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
001730' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;FAILED TO SET UPPER BYTE & LEAVE LOWER BYTE THE SAME
```



```
                                ;*****************************************************************
```

```
                                        ;TEST BICB INSTRUCTION CLEARS LDR

001736' 005077  176266          TSBICB: CLR     @LDR                    ;CLEAR LAMP DATA REGISTER
001742' 012777  052525  176260          MOV     #52525,@LDR             ;LOAD ALTERNATING 1'S IN LSR
001750' 012767  052400  176126          MOV     #52400,ASTAT            ;EXPECTED DATA
001756' 142777  177777  176244          BICB    #-1,@LDR                ;CLEAR LOWER BYTE OF LDR
001764' 017767  176240  176110          MOV     @LDR,ACSR               ;READ ACTUAL DTA
001772' 026767  176106  176102          CMP     ASTAT,ACSR              ;COMPARE RESULTS
002000' 001406                          BEQ     1$                      ;BRANCH IF EQUAL
002002' 012767  000025  176076          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
002010' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;FAILED TO CLEAR LOWER BYTE & LEAVE UPPER BYTE THE SAME
                                        ;*****************************************************************
002016' 012777  052525  176204  1$:     MOV     #52525,@LDR             ;LOAD ALTERNATING 1'S IN LDR
002024' 016767  176200  176214          MOV     LDR,TEMPL               ;GET ADDRESS OF LDR
002032' 005267  176210                  INC     TEMPL                   ;GET ADDRESS OF UPPER BYTE OF LDR
002036' 017767  000025  176040          MOV     #125,ASTAT              ;EXPECTED DATA
002046' 142777  177777  176174          BICB    #-1,@TEMPL              ;CLEAR UPPER BYTE OF LDR
002052' 017767  176152  176022          MOV     @LDR,ACSR               ;READ ACTUAL DATA
002060' 026767  176020  176014          CMP     ASTAT,ACSR              ;COMPARE RESULTS
002066' 001406                          BEQ     NOCHN1                  ;BRANCH IF EQUAL
002070' 012767  000025  176010          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
002076' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;FAILED TO CLEAR UPPER BYTE & LEAVE LOWER BYTE THE SAME.
                                        ;*****************************************************************

                                        ;TEST THAT PBSR AND LDR DON'T CHANGE

002104' 005077  176114          NOCHN1: CLR     @PBSR                   ;CLEAR STATUS REGISTER
002110' 005067  175770                  CLR     ASTAT                   ;LOAD EXPECTED DATA
002114' 012777  052525  176106          MOV     #52525,@LDR             ;LOAD PATTERN INTO LDR
002122' 017767  176076  175752          MOV     @PBSR,ACSR              ;READ STATUS REGISTER
002130' 026767  175750  175744          CMP     ASTAT,ACSR              ;COMPARE RESULTS
002136' 001406                          BEQ     1$                      ;GO TO NEXT SECTION
002140' 012767  000025  175740          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
002146' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;PBSR CHANGED IN ERROR
                                        ;*****************************************************************
002154' 005077  176050          1$:     CLR     @LDR                    ;CLEAR LDR
002160' 012777  100000  176036          MOV     #BIT15,@PBSR            ;SET BIT 15 IN PBSR
002166' 017767  176036  175706          MOV     @LDR,ACSR               ;READ LDR DATA
002174' 026767  175704  175700          CMP     ASTAT,ACSR              ;COMPARE RESULTS
002202' 001406                          BEQ     NOCHN2                  ;GO TO NEXT TEST
002204' 012767  000025  175674          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
002212' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;LDR CHANGED IN ERROR
                                        ;*****************************************************************
```



```
                                        ;TEST THAT PBR AND LDR DON'T CHANGE

002220' 005077  176002          NOCHN2: CLR     @PBR                    ;CLEAR PBR
002224' 012777  000001  175774          MOV     #BIT0,@PBR              ;SET ALL 1'S IN PBR
002232' 012767  177777  175644          MOV     #177777,ASTAT           ;LOAD EXPECTED DATA
002240' 012777  052525  175762          MOV     #52525,@LDR             ;LOAD PATTERN INTO LDR
002246' 017767  175754  175626          MOV     @PBR,ACSR               ;READ PBR
002254' 026767  175624  175620          CMP     ASTAT,ACSR              ;COMPARE RESULTS
002262' 001406                          BEQ     1$                      ;GO TO NEXT SECTION
002264' 012767  000025  175614          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
002272' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;PBR CHANGED IN ERROR
                                        ;*****************************************************************
002300' 005077  175724          1$:     CLR     @LDR                    ;CLEAR LDR
002304' 005077  175716                  CLR     @PBR                    ;CLEAR PBR
002310' 005067  175570                  CLR     ASTAT                   ;LOAD EXPECTED DATA
002314' 012777  000001  175704          MOV     #BIT0,@PBR              ;SET PBR TO ALL ONES
002322' 017767  175702  175552          MOV     @LDR,ACSR               ;READ LDR
002330' 026767  175550  175544          CMP     ASTAT,ACSR              ;READ LDR
002336' 026767  175542  175536          CMP     ASTAT,ACSR              ;COMPARE RESULTS
002344' 001406                          BEQ     NOCHN3                  ;GO TO NEXT TEST
002346' 012767  000025  175532          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
002354' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;LDR CHANGED IN ERROR
                                        ;*****************************************************************

                                        ;TEST THAT PBSR AND PBR DON'T CHANGE

002362' 005077  175640          NOCHN3: CLR     @PBR                    ;CLEAR PBR
002366' 005067  175512                  CLR     ASTAT                   ;LOAD EXPECTED DATA
002372' 012777  052525  175624          MOV     #52525,@PBSR            ;LOAD PATTERN INTO PBSR
002400' 017767  175622  175474          MOV     @PBR,ACSR               ;READ PBR
002406' 026767  175472  175466          CMP     ASTAT,ACSR              ;COMPARE RESULTS
002414' 001406                          BEQ     1$                      ;GO TO NEXT SECTION
002416' 012767  000025  175462          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
002424' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;PBR CHANGED IN ERROR
                                        ;*****************************************************************
002432' 005077  175566          1$:     CLR     @PBSR                   ;CLEAR PBSR
002436' 012777  000001  175562          MOV     #BIT0,@PBR              ;SET PBR TO ALL ONES
002444' 017767  175554  175430          MOV     @PBSR,ACSR              ;READ PBSR
002452' 042767  000036  175422          BIC     #36,ACSR                ;MASK "ENCODER" BITS
002460' 026767  175420  175414          CMP     ASTAT,ACSR              ;COMPARE RESULTS
002466' 001406                          BEQ     2$                      ;BR IF NO ERROR
002470' 012767  000025  175410          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
002476' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;PBSR CHANGED IN ERROR
                                        ;*****************************************************************
002504' 005077  175516          2$:     CLR     @PBR                    ;CLEAR BUTTON REGISTER
002510' 005077  175514                  CLR     @LDR                    ;CLEAR LAMPS
002514' 005077  175504                  CLR     @PBSR                   ;CLEAR STATUS
```

```
                                        ;LOAD PATTERNS INTO THE LIGHTS FROM A TABLE
002520' 012767  002656' 000122  LIGHT:  MOV     #TABLE,10$              ;LOAD POINTER
002526' 017767  000116' 000116  1$:     MOV     @10$,11$                ;GET A VALUE
002534' 104413  000000'                 ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                                                ;MONITOR SHALL TEST END OF PASS
002540' 012767  001000  000106          MOV     #BIT9,12$               ;LOAD A COUNTER
002546' 016777  000100  175454          MOV     11$,@LDR                ;LOAD THE LIGHT REGISTER
002554'                         2$:
002554' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002560' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002564' 005367  000064                  DEC     12$                     ;DELAY
002570' 100371                          BPL     2$                      ;BR IF NOT DONE
002572' 062767  000002  000050          ADD     #2,10$                  ;UPDATE TABLE
002606' 005777  175420          3$:     TST     @#SR                    ;TEST FOR BUTTON PUSHED
002604' 100350                          BPL     1$                      ;BR IF NONE
002606' 017767  175414  175424          MOV     @#BR,TEMP               ;READ THE BUTTON
002614' 016777  175420  175406          MOV     TEMP,@LDR               ;LOAD LAMPS
002622' 012767  001000  000024          MOV     #BIT9,12$               ;LOAD A COUNTER
002630'                         4$:
002630' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002634' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002640' 005367  000010                  DEC     12$                     ;DELAY
002644' 100371                          BPL     4$                      ;BR BACK
002646' 000754                          BR      3$

002650' 000000          10$:    0
002652' 000000          11$:    0
002654' 000000          12$:    0
```



```
                                        ;TABLE OF VALUES TO BE LOADED INTO THE "LAMP DATA REGISTER"
002656'                 TABLE:
002656' 000001                  BIT0
002660' 000002                  BIT1
002662' 000004                  BIT2
002664' 000010                  BIT3
002666' 000020                  BIT4
002670' 000040                  BIT5
002672' 000100                  BIT6
002674' 000200                  BIT7
002676' 000400                  BIT8
002700' 001000                  BIT9
002702' 002000                  BIT10
002704' 004000                  BIT11
002706' 010000                  BIT12
002710' 020000                  BIT13
002712' 040000                  BIT14
002714' 100000                  BIT15

002716' 174637                  174637
002720' 174637                  174637
002722' 003140                  3140
002724' 003140                  3140
002726' 125252                  125252
002730' 125252                  125252
002732' 052525                  52525
002734' 052525                  52525
002736' 007417                  7417
002740' 007417                  7417
002742' 170360                  170360
002744' 170360                  170360

002746' 000001                  BIT0
002750' 000022                  BIT4!BIT1
002752' 000444                  BIT6!BIT5!BIT2
002754' 011110                  BIT12!BIT9!BIT6!BIT3
002756' 022200                  BIT13!BIT10!BIT7
002760' 044000                  BIT14!BIT11
002762' 100000                  BIT15

002764' 000010                  BIT3
002766' 000204                  BIT7!BIT2
002770' 004102                  BIT11!BIT6!BIT1
002772' 102041                  BIT15!BIT10!BIT5!BIT0
002774' 041020                  BIT14!BIT9!BIT4
002776' 020400                  BIT13!BIT8
003000' 010000                  BIT12
```

SEQ 0016

```
630
631   003002' 100000                    BIT15
632   003004' 044000                    BIT14!BIT11
633   003006' 022200                    BIT13!BIT10!BIT7
634   003010' 011110                    BIT12!BIT9!BIT6!BIT3
635   003012' 000444                    BIT8!BIT5!BIT2
636   003014' 000022                    BIT4!BIT1
637   003016' 000001                    BIT0
638
639   003020' 010000                    BIT12
640   003022' 020400                    BIT13!BIT8
641   003024' 041020                    BIT14!BIT9!BIT4
642   003026' 102041                    BIT15!BIT10!BIT5!BIT0
643   003030' 004102                    BIT11!BIT6!BIT1
644   003032' 000204                    BIT7!BIT2
645   003034' 000010                    BIT3
646
647   003036' 000001                    BIT0
648   003040' 000002                    BIT1
649   003042' 000004                    BIT2
650   003044' 000010                    BIT3
651   003046' 000200                    BIT7
652   003050' 004000                    BIT11
653   003052' 100000                    BIT15
654   003054' 040000                    BIT14
655   003056' 020000                    BIT13
656   003060' 010000                    BIT12
657   003062' 000400                    BIT8
658   003064' 000020                    BIT4
659   003066' 000040                    BIT5
660   003070' 000100                    BIT6
661   003072' 002000                    BIT10
662   003074' 001000                    BIT9
663
664   003076' 177777                    -1
665   003100' 000000                    0
666   003102' 177777                    -1
667   003104' 000000                    0
668   003106' 177777                    -1
669   003110' 000000                    0
670
671   003112' 123456                    123456                    ;TERM
672   003114' 000310                    .BLKW   200.
673   003734' 123456                    .WORD   123456
674           000001                    .END
```



SEQ 0017

```
ACSR      000102R   185#  249*  250   260*  261   271*  272   282*  283   295*  296   308*  309
                    319*  320   331*  332   344*  345   358*  359   372*  373   385*  386   400*
                    401   416*  417   431*  432   443*  444   456*  457   468*  469   481*  482
                    490*  491   504*  505   515*  516   517   529*  530   538*  539*  540
ADDR      000006R   151#  206   207   208   235   236   237
ADDR22=   001000    203#
ASB       000106R   189#
ASTAT     000104R   187#  247*  248   250   258*  259   261   269*  270   272   280*  281   283
                    289*  293*  294   296   302*  306*  309   317*  320   322   329*  330   332   341*
                    345   355*  359   369*  373   383*  384   386   397*  398*  401   414*  417
                    429*  432   441*  444   454*  457   466*  469   479*  482   491   502*  505
                    513*  516   517   527*  530   540
AWAS      000110R   190#
BEGIN     000000R   148#  222   226   228   254   265   276   287   300   313   324   336   349
                    363   377   390   405   421   436   448   461   473   486   495   509   521
BITSAV    000236R   534   544   554   559   560   570   571
BIT0   =  000001    215#  395*  398   399   407*  412*  414   415   423*
                    703#  280   307   395   412   501   514   537   584   614   625   637   642
                    647
BIT1   =  000002    203#  585   615   624   636   643   648
BIT10  =  002000    203#  594   618   625   633   642   661
BIT11  =  004000    203#  595   619   624   632   643   652
BIT12  =  010000    203#  596   617   628   634   639   656
BIT13  =  020000    203#  597   618   627   633   640   655
BIT14  =  040000    203#  598   619   626   632   641   654
BIT15  =  100000    203#  329   342   356   370   489   599   620   625   631   642   653
BIT2   =  000004    203#  586   616   623   635   644   649
BIT3   =  000010    203#  587   617   622   634   645   650
BIT4   =  000020    203#  588   615   626   636   641   658
BIT5   =  000040    203#  589   616   625   635   642   659
BIT6   =  000100    203#  590   617   624   634   643   660
BIT7   =  000200    203#  591   618   623   633   644   651
BIT8   =  000400    203#  592   616   627   635   640   657
BIT9   =  001000    203#  556   568   593   617   626   634   641   662
BREAK$=   104407    203#  559   560   570   571
BR1       000012R   153#
BR2       000013R   154#
BTOD$  =  104421    203#
CDATA$=   104412    203#
CLBITO    000774R   310   317#
CONFIG    000056R   173#
CSRA      000100R   183#
DATCK$=   104411    203#
DATER$=   104404    203#
DVID1     000014R   155#
ENDIT$=   104413    203#  554
END$   =  104410    203#  228
ERRTYP    000106R   188#  224*  252*  263*  274*  285*  298*  311*  322*  334*  347*  361*  375*
                    388#  403*  419*  434*  446*  459*  471*  484*  493*  507*  519*  532*  542*
EXIT$  =  104400    203#
FLTO      000652R   293#
FLTI      000576R   273   280#
GETPA$=   104415    203#
GWBUF$=   104414    203#
HRDCNT    000044R   166#
HRDER$=   104405    203#  226   254   265   276   287   300   313   324   336   349   363   377
```

```
                        390     405     421     436     448     461     473     486     495     509     521     534     544
HRDPAS  000050R         170#
ICONT   000036R         165#
ICOUNT  000040R         166#
IDNUM   000122R         195#
IMIT    000030R         162#
INTR    000120R         194#    231*
LDIE    001322R         374     381#
LDPB    001040R         321     328#
LDR     000230R         208#    236*    239*    248*    249     259*    260     270*    271     281*    282     294*    295
                        394*    396*    399*    400     413*    415*    416     427*    428*    430*    431     438*    439
                        443     452*    453*    455*    456     463*    464     468     480*    488*    490     503*    511*
                        515     547*    557*    567*
LD1     000414R         247#
LD25    000462R         251     258#
LD52    000530R         262     269#
LIGHT   002520R         552#
MAP22S= 104416          203#
MODNAM  000000R         149#
MODSP   000224R         163     201#
MSGWS   = 104403        203#
MSGSS   = 104402        203#
MSGS    = 104401        203#
NOCHN1  002104R         470     478#
NOCHN2  002220R         498     506#
NOCHN3  002462R         518     526#
NULL    = 000000        203#    226     254     265     276     287     300     313     324     336     349     363     377
                        390     405     421     436     448     461     473     486     495     509     521     534     544
OPEN    = 000000        150     156     157     158     159     176     177     178     179     180     181     182     183
                        185     187     189     190     192     193     194     203#
OTOAS   = 104420        203#
PASCNT  000034R         164#
PBCLR   001112R         333     340#
PBCLRD  001244R         360     368#
PBCLRH  001166R         346     354#
PBR     000226R         207#    237*    238*    307*    308     318*    319     357     382*    500*    501*    504     512*
                        514*    526*    529     537*    546*    566
PBSR    000224R         206#    235*    328*    330*    331     340*    342*    343*    344     354*    356*    358     368*
                        370*    371     372     381*    384*    385     478*    481     489*    528*    536*    538     548*
                        564
PIRQS   = 000004        203#    222
POPSP   = 005726        203#
POPSP2= 022626          203#
PRTY    = 000000        154     203#
PRTY0   = 000000        203#
PRTY1   = 000040        203#
PRTY2   = 000100        203#
PRTY3   = 000140        203#
PRTY4   = 000200        153     203#
PRTY5   = 000240        203#
PRTY6   = 000300        203#
PRTY7   = 000340        203#
PS      = 177776        203#
PSW     = 177776        203#
PUSH    = 005746        203#
PUSH2   = 024646        203#
```



```
RAND$   = 104417        203#
RANNUM  000054R         172#
RESTRT  000320R         191     234#
RES1    000056R         174#
RES2    000060R         175#
RSTRT   000112R         191#
SBADR   000102R         184#
SOFCNT  000042R         167#
SOFER$= 104406          203#
SOFPAS  000046R         169#
SPOINT  000032R         163#
SPSIZ   = 000040          1#    196
SR1     000016R         156#
SR2     000020R         157#
SR3     000022R         158#
SR4     000024R         159#
START   000276R         162     231#
STAT    000026R         161#
SVR0    000062R         176#
SVR1    000064R         177#
SVR2    000066R         178#
SVR3    000070R         179#
SVR4    000072R         180#
SVR5    000074R         181#
SVR6    000076R         182#
SYSCNT  000052R         171#
TABLE   002656R         552     583#
TEMP    000240R         216#    357*    371*    566*    567
TEMPL   000246R         219#    439*    440*    442*    464*    465*    467*
TEMP1   000242R         217#
TEMP2   000244R         218#
TRPDFD= 000022          203#
TSBICB  001736R         445     452#
TSBISB  001570R         427#
TSBITC  001400R         387     394#
TSBITS  001502R         412#
UNEXPT  000250R         220#    244
VECT    000232R         210#    241*    244*
VECTOR  000010R         152#    210     211     241     242
VECTR1  000234R         211#    242*    243*    245*
WASADR  000104R         186#
WDFR    000116R         193#    232*
WDTO    000114R         192#    233*
XFLAG   000005R         150#
.       = 003736R       672#

. ABS.  000000      000
        003736      001

ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

XLKAB0,XLKAB0/SOL/CRF:SYM=DDXCOM,XLKAB0
RUN-TIME: 1 2 .3 SECONDS
```

SEQ 0020

RUN-TIME RATIO: 24/4=5.4
CORE USED:  7K  (13 PAGES)