

DH11

DH11 DIAGNOSTIC MD-11-DZDHM-A

EP-DZDHM-A-DL-A

FEB 1976

COPYRIGHT ©1976

digital

FICHE 1 OF 2

Made in U.S.A.

DZDHMA
SEQ

This page contains a grid of 100 diagnostic test results, arranged in 10 rows and 10 columns. Each cell in the grid contains a small table or diagram, likely representing a specific diagnostic test or component check. The content is dense and technical, typical of a diagnostic manual. The top-left cell contains the text 'DZDHMA SEQ'. The rest of the page is filled with various data points, possibly test results, component identifiers, and diagnostic procedures.

DH11

DH11 DIAGNOSTIC
MD-11-DZDHM-A

EP-DZDHM-A-DL-A

FEB 1976

COPYRIGHT ©1976

digital

FICHE 2 OF 2

Made In U.S.A.

111 11112 11113 11114 11115 11116 11117 11118 11119 11120 11121 11122 11123 11124 11125 11126 11127 11128 11129 11130 11131 11132 11133 11134 11135 11136 11137 11138 11139 11140 11141 11142 11143 11144 11145 11146 11147 11148 11149 11150 11151 11152 11153 11154 11155 11156 11157 11158 11159 11160 11161 11162 11163 11164 11165 11166 11167 11168 11169 11170 11171 11172 11173 11174 11175 11176 11177 11178 11179 11180 11181 11182 11183 11184 11185 11186 11187 11188 11189 11190 11191 11192 11193 11194 11195 11196 11197 11198 11199 11200

4010	012314	004767	012014
4011	012320	012704	037400
4012	012324	062702	000016
4013	012330	016103	000016
4014	012334	004767	010336
4015	012340	004567	010552
4016	012344	025144	
4017	012346	031563	
4018	012350	104057	
4050	012352	000640	

```

JSR PC,SAPS
MOV #37400,R4
ADD #SSR,R2
MOV SSR(R1),R3
JSR PC,SUER2A
JSR RS,SUNUM
LINE
EM57+44
ERROR 57
BR 18

```

```

;GO SAVE PSW
;SET UP S/B DATA
;SET UP REGADR
;SAVE WAS DATA
;GO SET UP ERROR INFO
;PUT LINE NO. IN MSG

;READING SILO FAILED TO DEC SSR OR
;STORAGE OVFL SET AT WRONG COUNT
;GO TRY NEXT LINE

```


TX CLOCK LINE "N" SIGNALS ON UART PIN 40

(1)									
(1)									
(1)									
4055	012364	012767	012414	166516	%	MOV	828,SLPERR	:SET UP ERROR LOOP RETURN	
4056	012372	004767	010426		18:	JSR	PC,SELIN	:GO SELECT A LINE TO TEST	
4057	012376	000534				BR	TST42	:BR IF TESTED ALL SELECTED LINES	
4058	012400	012767	002100	166610		MOV	82100,STMP7	:INIT TI START WITH LOWEST SPEED	
4059	012406	012767	177777	012362		MOV	8-1,TIMEC	:INIT RELATIVE TIME CHECKER	
4060	012414	012711	004000		29:	MOV	8BIT11,(R1)	:CLEAR THE DM11	
4061	012420	156711	012520		39:	BISB	LINE,(R1)	:SELECT IT IN THE SCR	
4062	012424	012761	177775	000010		MOV	8-3,BCR(R1)	:SET BYTE COUNT TO XFER 3 CHARS	
4063	012432	005061	000006			CLR	CAR(R1)	:GET TEST DATA STARTING AT LOC. 0	
4064	012436	016761	166554	000004		MOV	STMP7,LPR(R1)	:SELECT A XMIT SPEED	
4065	012444	016761	011742	000012		MOV	LINMSK,BAR(R1)	:ACTIVATE THE TRANSMITTER	
4066									
4067	012452	012767	000001	012512		MOV	81,TIMEA	:INIT TIMER A	
4068	012460	005067	012510			CLR	TIMEB	:INIT TIMER B	
4069	012464	005711			48:	TST	(R1)	:XMITTR DONE SET YET ?	
4070	012466	100437				BMI	55	:BR IF YES	
4071	012470	004767	011456			JSR	PC,TIMEIT	:CALL THE TIMER	
4072	012474	000773				BR	45	:TIMER ROUTINE WILL MOVE RETURN PC	
4073								:AROUND THIS BRANCH IF TIME OUT OCCURS	
4074									
4075	012476	016767	166514	166474		MOV	STMP7,STMP0	:SAVE AND SET UP THE SPEED CODE	
4076	012504	000367	166470			SWAB	STMP0		
4077	012510	006267	166464			ASR	STMP0		
4078	012514	006267	166460			ASR	STMP0		
4079	012520	042767	177760	166452		BIC	8177760,STMP0		
4080	012526	011103				MOV	(R1),R3	:GET THE WAS DATA	
4081	012530	042703	000200			BIC	8BIT07,R3	:CLEAR UNINTERESTING BITS	
4082	012534	010102				MOV	R1,R2	:MAKE REGADR = DEVADR	
4083	012536	012704	100000			MOV	8BIT15,R4	:SET UP S/B DATA	
4084	012542	156704	012376			BISB	LINE,R4		
4085	012546	004767	010124			JSR	PC,SUER2A	:GO SET UP ERROR INFO	
4086	012550	004567	010340			JSR	RS,SUNUM	:GO SET LINE NO. IN MSG	
4087	012556	025144				LINE			
4088	012560	031076				EM50+53			
4089	012562	104053				ERROR	53	:TIMED OUT WAITING FOR XMIT DONE	
4090	012564	000426				BR	88	:GO TEST NEXT SPEED	
4091									
4092	012566	016703	012402		58:	MOV	TIMEB,R3	:GET THE WAS COUNT	
4093	012572	016704	012400			MOV	TIMEC,R4	:GET LASTR CHECK COUNT	
4094	012576	020304				CMP	R3,R4	:COMPARE RELATIVE TIMES	
4095	012600	103420				BLO	88	:BR IF THIS SPEED FASTER THAN LAST	
4096								:SPEED TESTED	
4097									
4098	012602	004767	011526		78:	JSR	PC,SAPS	:SAVE THE ERROR PSW	
4099	012606	016702	166404			MOV	STMP7,R2	:GET SPEED CODE AND RIGHT JUSTIFY	
4100	012612	000302				SWAB	R2		
4101	012614	006202				ASR	R2		
4102	012616	006202				ASR	R2		
4103	012620	042702	177760			BIC	8177760,R2	:STRIP AWAY ALL JUNK	
4104	012624	004767	010046			JSR	PC,SUER2A	:GO SET UP ERROR INFO	
4105	012630	004567	010262			JSR	RS,SUNUM	:GO PUT LINE NO. IN MSG	

4106	012634	025144				LINE		
4107	012636	026441				EM17+41		
4108	012640	104017				ERROR	17	; TRANSMITTER SPEED INCORRECT
4109								
4110	012642	016767	012326	012326	BS:	MOV	TIMEB, TIMEC	; SET UP NEW CHECK TIMER COUNT
4111	012650	062767	002100	166340		ADD	#2100, STMP7	; GENERATE NEXT SPEED
4112	012656	022767	035600	166332		CMP	#35600, STMP7	; DONE ALL SPEEDS ?
4113	012664	001253				BNE	25	; BR IF NOT
4114	012666	000641				BR	15	; GO TEST NEXT LINE
4115								

4141	013020	006367	166154		ASL	STMP0		
4142	013024	006367	166150		ASL	STMP0		
4143	013030	000367	166144		SWAB	STMP0		
4144	013034	042767	177760	166136	BIC	#177760,STMP0		
4145	013042	011103			MOV	(R1),R3	;GET THE WAS DATA	
4146	013044	010102			MOV	R1,R2	;MAKE REGADR = DEVARDR	
4147	013046	012704	100200		MOV	#BIT15+BIT07,R4	;SET UP S/B DATA	
4148	013052	156704	012066		BISB	LINE,R4		
4149	013056	004767	007614		JSR	PC,SUER2A	;GO SET UP ERROR INFO	
4150	013062	004567	010030		JSR	RS,SUNUM	;GO SET LINE NO. IN MSG	
4151	013066	025144			LINE			
4152	013070	026720			EM22+51			
4153	013072	104054			ERROR	54	;TIMED OUT WAITING FOR CHAR AVAIL	
4154	013074	000426			BR	85	;GO TEST NEXT SPEED	
4155								
4156	013076	016703	012072	55:	MOV	TIMEB,R3	;GET THE WAS COUNT	
4157	013102	016704	012070		MOV	TIMEC,R4	;GET THE CHECK COUNT	
4158	013106	020304			CMP	R3,R4	;COMPARE RELATIVE TIMES	
4159	013110	103420			BLO	85	;BR IF TIME INDICATES THIS SPEED FASTER	
4160							;THAN LAST SPEED	
4161								
4162	013112	004767	011216	75:	JSR	PC,SAPS	;SAVE THE ERROR PSM	
4163	013116	016702	166074		MOV	STMP7,R2	;GET SPEED CODE AND RIGHT JUSTIFY	
4164	013122	006302			ASL	R2		
4165	013124	006302			ASL	R2		
4166	013126	000302			SWAB	R2		
4167	013130	042702	177760		BIC	#177760,R2	;STRIP AWAY ALL JUNK	
4168	013134	004767	007536		JSR	PC,SUER2A	;GO SET UP ERROR INFO	
4169	013140	004567	007752		JSR	RS,SUNUM	;GO PUT LINE NO. IN MSG	
4170	013144	025144			LINE			
4171	013146	026600			EM20+36			
4172	013150	104020			ERROR	20	;RECEIVER SPEED INCORRECT	
4173								
4174	013152	016767	012016	012016	85:	MOV	TIMEB,TIMEC	;SET UP NEW CHECK TIMER COUNT
4175	013160	062767	002100	166030	ADD	#2100,STMP7	;GENERATE NEXT SPEED	
4176	013166	022767	035600	166022	CMP	#35600,STMP7	;DONE ALL SPEEDS ?	
4177	013174	001255			BNE	25	;BR IF NOT	
4178	013176	000643			BR	15	;GO TEST NEXT LINE	
4179								

NB1 LPR 00 H Fh?

```

(1)
(1)
4184 013210 012767 013236 165672 X
4185 013216 004767 007602 15:
4186 013222 000511
4187 013228 012705 025160
4188 013230 005002
4189 013236 012567 165760 25:
4190 013242 012711 004000 35:
4191 013248 156711 011676
4192 013254 012761 177777 000010
4193 013260 012761 001216 000006
4194 013266 012761 033500 000004
4195 013270 050261 000004
4196 013274 156761 011112 000012
4197
4198 013302 012767 000001 011662
4199 013310 005067 011660
4200 013314 105711 45:
4201 013316 100424
4202 013320 004767 010626
4203 013324 000773
4204
4205
4206 013326 004767 011002
4207 013332 011103
4208 013334 042703 177560
4209 013340 012704 000200
4210 013344 156704 011574
4211 013350 004767 007322
4212 013354 004567 007536
4213 013360 025144
4214 013362 026720
4215 013364 104055
4216 013366 000422
4217
4218
4219 013370 016103 000002 55:
4220 013374 012704 000200
4221 013400 156704 011540
4222 013404 000304
4223 013406 156704 165604
4224 013412 020304
4225 013414 001407
4226
4227 013416 004767 007250
4228 013422 004567 007470
4229 013426 025144
4230 013430 026761
4231 013432 104023
4232
4233 013434 005202 65:
4234 013436 022702 000004
4235 013442 001273
4236 013444 000664

```

```

MOV #35,SLPERR ;SET UP ERROR LOOP RETURN
JSR PC,SELIN ;GO SELECT A LINE TO TEST
BR TST44 ;BR IF DONE ALL SELECTED LINES
MOV #DATA2,R5 ;GET POINTER TO DATA TABLE
CLR R2 ;INIT R2 TO START AT CHAR LENGTH OF 5 BITS
MOV (R5)+,STMP7 ;PUT TEST CHAR IN XMIT BUFFER
MOV #BIT11,(R1) ;CLEAR THE DM11
BISB LINE,(R1) ;SELECT THE LINE
MOV #-1,BCR(R1) ;SET BYTE COUNT TO -1
MOV #STMP7,CAR(R1) ;SET CURRENT ADDRESS REG
MOV #33500,LPR(R1) ;SET BAUD RATE TO 9600
BIS R2,LPR(R1) ;SELECT CHAR LENGTH
BISB LIMSK,BAR(R1) ;ACTIVATE THE SELECTED LINE

MOV #1,TIMEA ;INIT TIMER A
CLR TIMEB ;INIT TIMER B
TSTB (R1) ;RCVR DONE YET ??
BMI 55 ;BR IF YES
JSR PC,TIMEIT ;CALL THE TIMER
BR 45 ;TIMER ROUTINE WILL MOVE RETURN PC
;AROUND THIS BRANCH IF TIME OUT OCCURS

JSR PC,SAPS ;SAVE THE ERROR PSW
MOV (R1),R3 ;GET THE SCR
BIC #177560,R3 ;CLEAR UNINTERESTING BITS
MOV #200,R4 ;SET UP S/B DATA
BISB LINE,R4
JSR PC,SUER2A ;GO SET UP ERROR INFO
JSR R5,SUNUM ;GO SET LINE NO. IN MSG
LINE
EM22+51
ERROR 55 ;CHAR AVAIL FAILED TO SET ON TIME
BR 65 ;GO TEST NEXT CHAR LENGTH

MOV NRC(R1),R3 ;GET THE WAS DATA
MOV #200,R4 ;SET UP THE S/B DATA IN R4
BISB LINE,R4
SWAB R4
BISB STMP7,R4
CMP R3,R4 ;WAS THE RCVD DATA CORRECT ??
BEQ 65 ;BR IF YES

JSR PC,SUER2 ;GO SET UP THE ERROR INFO
JSR R5,SUNUM ;GO PUT LINE NO. IN MSG
LINE
EM23+36
ERROR 23 ;DATA COMPARE ERROR

INC R2 ;DO NEXT CHAR LENGTH ON SELECTED LINE
CMP #4,R2 ;HAVE WE DONE ALL FOUR CHAR LENGTHS ??
BNE 25 ;BR IF NOT
BR 15 ;GO DO NEXT LINE

```


(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)

- 2. IF THE FAULT GIVES DATA ERRORS BUT AFFECTS ONLY CERTAIN PATTERNS ON ONE LINE THE FAULT IS MOST LIKELY A "UART" CHIP.
- 3. IF THE FAULT GIVES DATA ERRORS BUT AFFECTS ONLY CERTAIN PATTERNS ON ALL LINES SUSPECT THE DATA PATHS EXTERNAL TO, THE M7280 MODULES.
- 4. IF THE FAULT CAUSES NO DATA ERRORS BUT GIVES "BAR", "BCR", OR "CAR" ERRORS SUSPECT THE M7278 OR M7277 MODULES RESPECTFULLY.

KEY LOGIC (REFER TO TEST 43)

4240	013456	012767	013500	165424	X	MOV	#25,SLPERR	:SET UP ERROR LOOP RETURN
4241	013464	004767	007334		1S:	JSR	PC,SELINE	:GO SELECT A LINE TO TEST
4242	013470	000401				BR	11\$:BR IF ALL SELECTED LINES DONE
4243	013472	000402				BR	2\$:GO TEST IT
4244	013474	000167	007424		11S:	JMP	#\$:EXIT TEST
4245	013500	016701	010674		2S:	MOV	DHADR,R1	:RESET DEVDADR
4246	013504	012711	004000			MOV	#BIT11,(R1)	:CLEAR THE DH11
4247	013510	156711	011430			BISB	LINE,(R1)	:SET SELECT BITS IN SCR
4248	013514	012761	033436	000006		MOV	#TBUF,CAR(R1)	:SET UP BUS ADDRESS REG
4249	013522	012761	177400	000010		MOV	#-400,BCR(R1)	:SET UP BYTE COUNT
4250	013530	012761	033503	000004		MOV	#33503,LPR(R1)	:SET LINE PARAMETERS
4251	013536	012705	032436			MOV	#RBUF,R5	:SET UP POINTER TO INPUT DATA BUFFER
4252	013542	052711	001000			BIS	#BIT09,(R1)	:SET MAINT MODE BIT
4253	013546	016761	010640	000012		MOV	LINMSK,BAR(R1)	:ACTIVATE THE SELECTED LINE
4254	013554	012767	000002	011410		MOV	#2,TIMEA	:INIT TIMER A
4255	013562	005067	011406			CLR	TIMEB	:INIT TIMER B
4256	013566	105711			3S:	TSTB	(R1)	:RCVR DONE YET ??
4257	013570	100425				BMI	4\$:BR IF YES
4258	013572	004767	010354			JSR	PC,TIMEIT	:CALL THE TIMER
4259	013576	000773				BR	3\$:TIMER ROUTINE WILL MOVE RETURN PC :AROUND THIS BRANCH IF TIME OUT OCCURS
4260	013600	004767	010530			JSR	PC,SAPS	:SAVE THE ERROR PSM
4261	013604	010102				MOV	R1,R2	:SET UP REGADR
4262	013606	011203				MOV	(R2),R3	:GET THE WAS DATA
4263	013610	042703	176400			BIC	#176400,R3	:CLEAR JUNK BITS
4264	013614	012704	001200			MOV	#1200,R4	:SET UP S/B DATA
4265	013620	156704	011320			BISB	LINE,R4	
4266	013624	004767	007046			JSR	PC,SUER2A	:GO SET UP ERROR INFO
4267	013630	004567	007262			JSR	R5,SUNUM	:PUT LINE NO. IN MESSAGE
4268	013634	025144				LINE		
4269	013636	026720				EM22+51		
4270	013640	104022				ERROR	22	:CHAR AVAIL TIMEOUT
4271	013642	000710				BR	1\$:GO TRY NEXT LINE
4272	013644	016125	000002		4S:	MOV	NRC(R1),(R5)+	:SAVE THE RECEIVED DATA
4273	013650	022705	033436			CMP	#RBUF+1000,R5	:INPUT BUFFER FULL ??
4274	013654	001344				BNE	3\$:BR IF NOT

4280	013656	012702	033436		MOV	#TBUF,R2		;SET UP POINTER TO OUTPUT BUFFER
4281	013662	012701	032436		MOV	#RBUF,R1		;SET UP POINTER TO INPUT BUFFER
4282	013666	111204		5S:	MOVB	(R2),R4		;SET UP S/B DATA IN R4
4283	013670	042704	177400		BIC	#177400,R4		
4284	013674	000304			SWAB	R4		
4285	013676	156704	011242		BISB	LINE,R4		
4286	013702	152704	000200		BISB	#200,R4		
4287	013706	000304			SWAB	R4		
4288	013710	011103			MOV	(R1),R3		;GET THE WAS DATA
4289	013712	020304			CMP	R3,R4		;DATA CORRECT ??
4290	013714	001407			BEQ	6S		;BR IF YES
4291								
4292	013716	004767	006750		JSR	PC,SUER2		;GO SET UP ERROR INFO
4293	013722	004567	007170		JSR	RS,SUNUM		;PUT LINE NO. IN MESSAGE
4294	013726	025144			LINE			
4295	013730	030176			EM37+33			
4296	013732	104037			ERROR	37		;DATA COMPARE ERROR
4297								
4298								
4299	013734	005202		6S:	INC	R2		;UPDATE DATA BUFFER POINTERS
4300	013736	062701	000002		ADD	#2,R1		
4301	013742	022701	033436		CMP	#RBUF+1000,R1		;COMPARED ALL 256. CHARS ??
4302	013746	001347			BNE	5S		;BR IF NOT
4303								
4304	013750	016701	010424		MOV	DHADR,R1		;RESET DEVAOR
4305	013754	010102			MOV	R1,R2		;SET UP REGADR
4306	013756	062702	000012		ADD	#BAR,R2		
4307	013762	005712			TST	(R2)		;WAS THE "BAR" ALL ZEROES ??
4308	013764	001413			BEQ	7S		;BR IF YES
4309								
4310	013766	004767	010342		JSR	PC,SAPS		;SAVE THE ERROR PSM
4311	013772	011203			MOV	(R2),R3		;GET THE WAS DATA
4312	013774	005004			CLR	R4		;SET UP S/B DATA
4313	013776	004767	006674		JSR	PC,SUER2A		;GO SET UP ERROR INFO
4314	014002	004567	007110		JSR	RS,SUNUM		;PUT LINE NO. IN MESSAGE
4315	014006	025144			LINE			
4316	014010	030241			EM40+40			
4317	014012	104040			ERROR	40		; "BAR" REG NOT ALL ZEROES
4318								
4319	014014	010102		7S:	MOV	R1,R2		;SET UP REGADR
4320	014016	062702	000010		ADD	#BCR,R2		
4321	014022	005712			TST	(R2)		;BYTE COUNT REG ALL ZEROES ?
4322	014024	001413			BEQ	71S		;BR IF BYTE COUNT ZERO
4323								
4324	014026	004767	010302		JSR	PC,SAPS		;SAVE THE ERROR PSM
4325	014032	011203			MOV	(R2),R3		;GET THE WAS DATA
4326	014034	005004			CLR	R4		;SET UP THE S/B DATA
4327	014036	004767	006634		JSR	PC,SUER2A		;GO SET UP ERROR INFO
4328	014042	004567	007050		JSR	RS,SUNUM		;PUT LINE NO. IN MESSAGE
4329	014046	025144			LINE			
4330	014050	026046			EM10+44			
4331	014052	104010			ERROR	10		;BYTE COUNT NOT ALL ZEROES
4332								
4333	014054	010102		71S:	MOV	R1,R2		;SET UP REGADR

4334	014056	062702	000006		ADD	#CAR, R2	
4335	014062	022712	034036		CMF	#TBUF+400, (R2)	; DID "CAR" INCREMENT PROPERLY ?
4336	014066	001414			BEQ	72S	; BR IF YES
4338	014070	004767	010240		JSR	PC, SAPS	; SAVE THE ERROR PSW
4339	014074	011203			MOV	(R2), R3	; GET THE WAS DATA
4340	014076	012704	034036		MOV	#TBUF+400, R4	; SET UP S/B DATA
4341	014102	004767	006570		JSR	PC, SUER2A	; GO SET UP ERROR INFO
4342	014106	004567	007004		JSR	RS, SUNUM	; GO PUT LINE NO IN MESSAGE
4343	014112	025144			LINE		
4344	014114	025777			EM7+47		
4345	014116	104007			ERROR	7	; "CAR" NOT UPDATED CORRECTLY
4346							
4347	014120	000167	177340	72S:	JMP	1S	; GO DO NEXT LINE
4348							
4349	014124	000240		8S:	NOP		; EXIT POINT

4353
 (3)
 (3)
 (2) 014126 000004
 (1) 014130 012767 000045 165076
 4354

```

;*****
;#TEST 45 BASIC PARITY LOGIC TEST - ALL LINES - ODD PARITY
;*****
TST45: SCOPE
        MOV     #STN-1,STESTN    ;;SET TEST NUMBER IN MAIL BOX

```

```

.REM     X
TEST ABSTRACT:
*****

```

THIS TEST VERIFIES THE ODD PARITY FUNCTION FOR ALL SELECTED LINES USING THE "BREAK" FUNCTION TO FORCE PARITY ERRORS. REFER TO THE FLOW CHARTS IN THE PROGRAM DOCUMENTATION FOR TEST SEQUENCES.

ERRORS:

1. [ERROR 22] IS CALLED TO REPORT RCVR TIMEOUT
2. [ERROR 33] IS CALLED TO REPORT DATA/PARITY ERRORS

```

SYNC:   M7277 SH3 INIT A H EF2
*****

```

DEBUG:

1. IF FAULT AFFECTS ALL LINES SUSPECT THE M7278 MODULE.
2. IF IT AFFECTS ONLY ONE LINE SUSPECT THE "UART" MODULE FOR THAT LINE.

KEY LOGIC:

					M7278	SH7	PEN LPRO4 L	FF2
							PEV LPRO5 L	FN1
4355	014136	012767	014152	164744	%	MOV	#25,SLPERR	;;SET UP ERROR LOOP RETURN
4356	014144	004767	006654		15:	JSR	PC,SELIN	;;GO SELECT A LINE NO.
4357	014150	000506				BR	TST46	;;BR IF ALL SELECTED LINES DONE
4358	014152	012711	004000		25:	MOV	#BIT11,(R1)	;;CLEAR OUT THE DH11
4359	014156	012704	000260			MOV	#260,R4	;;SET UP THE S/B DATA IN R4
4360	014162	156704	010756			BISB	LINE,R4	
4361	014166	000304				SWAB	R4	
4362	014170	156711	010750			BISB	LINE,(R1)	;;SET LINE NO. IN SCR
4363	014174	012767	000377	010754		MOV	#377,TDATA1	;;LOAD XMIT BUFFER WITH TEST CHARACTER
4364	014202	012761	073563	000004		MOV	#73563,LPR(R1)	;;SET UP THE LINE PARAMETERS
4365	014210	012761	177777	000010		MOV	#-1,BCR(R1)	;;LOAD THE BYTE COUNT REG
4366	014216	012761	025156	000006		MOV	#TDATA1,CAR(R1)	;;LOAD THE BUS ADDR REG
4367	014224	016761	010162	000014		MOV	LINMSK,BKR(R1)	;;SET BREAK BIT FOR SELECTED LINE
4368	014232	016761	010154	000012		MOV	LINMSK,BAR(R1)	;;ACTIVATE THE XMITTR
4369								
4370	014240	012767	000001	010724		MOV	#1,TIMEA	;;INIT TIMER A
4371	014246	005067	010722			CLR	TIMEB	;;INIT TIMER B
4372	014252	105711			35:	TSTB	(R1)	;;RCVR DONE YET ??

4373	014254	100423		BMI	48		:BR IF YES
4374	014256	004767	007670	JSR	PC, TIMEIT		:CALL THE TIMER
4375	014262	000773		BR	38		:TIMER ROUTINE WILL MOVE RETURN PC
4376							:AROUND THIS BRANCH IF TIME OUT OCCURS
4377							
4378	014264	004767	010044	JSR	PC, SAPS		:SAVE THE ERROR PSM
4379	014270	011103		MOV	(R1), R3		:GET THE WAS DATA
4380	014272	012704	100200	MOV	#100200, R4		:SET UP THE S/B DATA
4381	014276	156704	010642	BISB	LINE, R4		
4382	014302	010102		MOV	R1, R2		:SET UP REGADR
4383	014304	004767	006366	JSR	PC, SUER2A		:GO SET UP ERROR INFO
4384	014310	004567	006602	JSR	RS, SUNUM		:PUT LINE NO. IN MESSAGE
4385	014314	025144		LINE			
4386	014316	026720		EM22+51			
4387	014320	104022		ERROR	22		:TIMED OUT WAITING FOR DATA AVAIL
4388	014322	000710		BR	18		:GO TEST NEXT LINE
4389							
4390	014324	016103	000002	45: MOV	NRC(R1), R3		:GET THE WAS DATA
4391	014330	020304		CMR	R3, R4		:CORRECT DATA RECEIVED ??
4392	014332	001704		BEQ	18		:BR IF YES
4393							
4394	014334	004767	007774	JSR	PC, SAPS		:SAVE THE ERROR PSM
4395	014340	010102		MOV	R1, R2		:SET UP THE REGADR
4396	014342	062702	000002	ADD	#NRC, R2		
4397	014346	004767	006324	JSR	PC, SUER2A		:GO SET UP ERROR INFO
4398	014352	004567	006540	JSR	RS, SUNUM		:PUT LINE NO. IN MESSAGE
4399	014356	025144		LINE			
4400	014360	027712		EM33+40			
4401	014362	104333		ERROR	33		:INCORRECT DATA OR PARITY ERROR
4402	014364	000667		BR	18		:GO TEST NEXT LINE

4405
(3)
(3)
(2)
(1)
4406
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)

014366 000004
014370 012767 000046 164636

```
*****  
;#TEST 46 MULTI-LINE PARITY DATA TEST - ALL SELECTED LINES  
*****  
#TST46: SCOPE  
MOV #STN-1,$STSTN ;;SET TEST NUMBER IN MAIL BOX
```

```
.REM %  
TEST ABSTRACT:  
*****  
  
THIS TEST VERIFIES ALL SELECTED LINES CAN TRANSMIT AND RECEIVE  
A BINARY COUNT PATTERN WHEN RUN CONCURRENTLY. ALL CHAR LENGTHS (5, 6, 7,  
AND 8 BITS) ARE TESTED WITH BOTH EVEN AND ODD PARITY CHECKING SPECIFIED.  
THE TEST ACTUALLY INCLUDES EIGHT SUB-TESTS - THE PARAMETERS FOR EACH  
SUB-TEST RETRIEVED FROM A TABLE TAGGED "PRTYTB;". REFER TO THE FLOW  
CHARTS IN THE PROGRAM DOCUMENT FOR A DETAILED DESCRIPTION OF THE TEST  
SEQUENCES.
```

- ERRORS:

1. [ERROR 41] IS CALLED TO REPORT FALSE RECEIVER INTRNS.
2. [ERROR 42] IS CALLED TO REPORT SILO OVERFLOW ERRORS
3. [ERROR 34] IS CALLED TO REPORT PARITY/DATA ERRORS
4. [ERROR 35] IS CALLED TO REPORT TEST TIMEOUTS

SYNC: (NONE)

DEBUG: (REFER TO TEST 45)

KEY LOGIC: (REFER TO TEST 45)

4407 014376 012767 014414 164504
4408 014404 012705 024462
4409 014410 005067 164602
4410 014414 162705 000004
4411 014420 005367 164572
4412 014424 022705 024516
4413 014430 001456
4414 014430 012706 001100
4415 014436 016701 007736
4416 014442 012567 164546
4417 014446 012567 164540
4418 014452 005267 164540
4419 014456 012711 004000
4420 014462 004767 006612
4421 014466 016167 000004 164504
4422 014474 016767 007710 164504
4423 014502 004767 007612
4424 014506 016702 007670
4425 014512 012722 014604

```
%  
MOV #15,$LPERR ;SET UP THE ERROR LOOP RETURN  
MOV #PRTYTB+4,$R5 ;SET UP POINTER TO TEST PARAMETERS  
CLR $TMP7 ;START WITH SUB TEST #00  
1$: SUB #4,$R5 ;RESET POINTER FOR ERROR LOOPS  
DEC $TMP7 ;RESET SUB TEST # FOR ERROR LOOP  
2$: CMP #PRTYTB+40,$R5 ;DONE ALL 8. SUB TESTS ??  
BEQ 21$ ;OR IF YES  
MOV #STACK,$SP ;RESET STACK POINTER FOR ERROR LOOPS  
MOV $DADR,$R1 ;RESET DEADR FOR ERROR LOOPS  
MOV ($R5)+,$TMP6 ;GET THE BYTE COUNT PARAMETER  
MOV ($R5)+,$TMP5 ;GET THE LINE PARAMETERS  
INC $TMP7 ;GENERATE NEW SUB-TEST NO.  
MOV #BIT11,$(R1) ;CLEAR THE D11  
JSR PC,$UPPAR ;GO SET UP PARAMETERS  
MOV LPA($R1),$TMP0 ;SAVE CURRENT LINE PARAMETERS  
MOV L$INSEL,$TMP3 ;SAVE SELECTED LINES PARAMETER  
JSR PC,$L$S2 ;GO LOCK OUT INTRNS  
MOV $D$VCT,$R2 ;SET UP THE VECTOR  
MOV #38,$(R2)+ ;GO TO 38 ON RCVR INTERRUPT
```

```

4426 014516 116712 010416      MOVB   DHALVL (R2)
4427 014522 012711 000100      MOV    #100 (R1)      ;ENABLE CHAR AVAIL INTERRUPTS
4428 014526 016767 007656      MOV    LINSEL,LINACT ;FLAG ALL SELECTED LINES ACTIVE
4429 014534 016761 007650      MOV    LINSEL,BAR(R1) ;ACTIVATE ALL SELECTED LINES
4430 014542 116767 164334      MOVB   $STNM,STMP2   ;SAVE THE TEST NO.
4431 014550 042767 177400      BIC    #177400,STMP2
4432 014556 004767 007522      JSR    PC,CHPS1      ;GO CLEAR PSW
4433 014562 000167 000176      JNP    7$           ;GO WAIT FOR INTERRUPTS
4434
4435 014566 012706 001100      21$:  MOV    #STACK,SP ;RESTORE THE SP
4436 014572 004767 007506      JSR    PC,CHPS1      ;GO CLEAR PSW
4437 014576 004767 007322      JSR    PC,RESTRP     ;RESTORE TRAP CATCHER
4438 014602 000536      BR     TST47         ;GO TO NEXT TEST
4439
4440 ;RECEIVER INTERRUPT SERVICE ROUTINE
4441
4442 014604 105711      3$:  TSTB   (R1)        ;CHAR AVAIL SET
4443 014606 100404      BMI    4$           ;BR IF YES
4444
4445 014610 012711 004000      MOV    #BIT11,(R1)  ;CLEAR OUT THE DH11
4446 014614 104041      ERROR  41           ;RCVR FALSE INTERRUPT - CHAR AVAIL NOT SET
4447 014616 000702      BR     2$           ;GO TRY NEXT SUB TEST
4448
4449 014620 032711 040000      4$:  BIT    #BIT14,(R1) ;SILO OVERFLOW ??
4450 014624 001404      BEQ    5$           ;BR IF NOT
4451
4452 014626 012711 004000      MOV    #BIT11,(R1)  ;CLEAR OUT THE DH11
4453 014630 104042      ERROR  42           ;SILO OVERFLOW ERROR
4454 014634 000673      BR     2$           ;GO TRY NEXT SUB TEST
4455
4456 014636 016103 000002      5$:  MOV    NRC(R1),R3   ;GET THE WAS DATA
4457 014642 010302      MOV    R3,R2        ;EXTRACT AND SAVE LINE NO.
4458 014644 000302      SWAB   R2
4459 014646 042702 177760      BIC    #177760,R2
4460 014652 010267 164332      MOV    R2,STMP4
4461 014656 006302      ASI    R2
4462 014660 026203 032436      CMP    RBUF(R2),R3  ;GENERATE TABLE OFFSETR
4463 014664 001426      BEQ    6$           ;CORRECT DATA RECEIVED ??
4464
4465 014666 004767 007442      JSR    PC,SAPS      ;SAVE THE ERROR PSW
4466 014672 012711 004000      MOV    #BIT11,(R1)  ;CLEAR OUT THE DH11
4467 014676 016204 032436      MOV    RBUF(R2),R4  ;SET UP S/B DATA
4468 014702 062701 000002      ADD    #NRC,R1      ;SET UP WAS ADDRESS
4469 014706 062702 032436      ADD    #RBUF,R2     ;SET UP S/B ADDRESS
4470 014712 004767 005760      JSR    PC,SUER2A    ;GO SET UP ERROR INFO
4471 014716 004567 006174      JSR    RS,SUNUM     ;PUT LINE NO. IN MESSAGE
4472 014722 001210      STMP4
4473 014724 027766      EM34+S1
4474 014726 004567 006164      JSR    RS,SUNUM     ;PUT SUBTEST NO. IN MESSAGE
4475 014732 001216      STMP7
4476 014734 030004      EM34+67
4477 014736 104034      ERROR  34           ;PARITY DATA COMPARE ERROR
4478 014740 000631      BR     2$           ;GO TRY NEXT SUBTEST
4479

```

```

4480 014742 105262 032436      6S:  YNCB      RBUF(R2)      ;GENERATE NEW RCVD DATA
4481 014746 005262 024660      INC          MULPTB(R2)   ;COUNT ONE BYTES RECEIVED
4482 014752 001003      BNE          61S      ;BR IF NOT DONE
4483 014754 046267 024616 007674  BIC          LINBIT(R2),LINACT ;FLAG THIS LINE DONE
4484 014762 000002      61S:  RTI          ;RETURN TO WAIT ROUTINE
4485
4486      ;WAIT ROUTINE
4487
4488 014764 012767 000002 010200  7S:  MOV          #2,TIMEA      ;INIT TIMER A
4489 014772 005067 010176      CLR          TIMEB        ;INIT TIMER B
4490 014776 005761 000012      8S:  TST          BAR(R1)    ;ALL LINES DONE XMITTING ??
4491 015002 001413      BEQ          9S          ;BR IF YES
4492 015004 004767 007142      JSR          PC,TIMEIT    ;CALL THE TIMER
4493 015010 000772      BR          8S          ;TIMER ROUTINE WILL MOVE RETURN PC
4494      ;AROUND THIS BRANCH IF TIME OUT OCCURS
4495
4496 015012 016167 000012 164166      MOV          BAR(R1),STMP3 ;SAVE THE ACTIVE LINES FLAG
4497 015020 012711 004000      MOV          #BIT11,(R1)  ;CLEAR OUT THE DH11
4498 015024 104035      ERROR       35          ;TIMED OUT WAITING FOR TRANSMITTERS TO FINISH
4499 015026 000167 177372      JMP          2S          ;GO TRY NEXT SUBTEST
4500
4501
4502 015032 012767 000001 010132  9S:  MOV          #1,TIMEA      ;INIT TIMER A
4503 015040 005067 010130      CLR          TIMEB        ;INIT TIMER B
4504 015044 005767 007606      10S: TST          LINACT     ;ALL CHARS RECEIVED ?
4505 015050 001411      BEQ          11S         ;BR IF YES
4506 015052 004767 007074      JSR          PC,TIMEIT    ;CALL THE TIMER
4507 015056 000772      BR          10S         ;TIMER ROUTINE WILL MOVE RETURN PC
4508      ;AROUND THIS BRANCH IF TIME OUT OCCURS
4509
4510 015060 016767 007572 164120      MOV          LINACT,STMP3 ;SET UP ACTIVE LINE PARAMETER
4511 015066 012711 004000      MOV          #BIT11,(R1)  ;CLEAR OUT THE DH11
4512 015072 104035      ERROR       35          ;SILO EMPTY TIMEOUT
4513 015074 000167 177324      11S: JMP          2S          ;GO TRY NEXT SUB TEST

```



```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
4517 015110 012767 015172 163772 %      MOV      #25,SLPERR      ;SET UP ERROR LOOP RETURN
4518 015116 005067 164064      CLR      STMP3          ;INIT I/O DATA FLAG
4519 015122 012711 004000      MOV      #BIT11,(R1)   ;CLEAR THE DH11
4520 015128 012705 024516      MOV      #AETAB,R5     ;GET POINTER TO AUTO ECHO DATA TABLE
4521 015134 005067 007256      CLR      LMSK1         ;INIT BIT TEST MARKER
4522 015136 000261      SEC      13S           ;SET "C" BIT FOR MARKER
4523 015140 000401      BR       13S          ;GO SHIFT MASK
4524 015142 000241      CLC      15S          ;INIT THE "C" BIT
4525 015144 006167 007244      ROL     LMSK1         ;SHIFT BIT MARKER
4526 015150 001407      BEQ     12S          ;BR IF DONE ALL LINES
4527 015152 012504      MOV     (R5)+,R4     ;SET UP THE S/B DATA
4528 015154 036767 007234 007226      BIT     LMSK1,LINSEL  ;TEST THIS LINE ?
4529 015162 001767      BEQ     1S           ;BR IF NOT
4530 015164 004767 005634      JSR     PC,SELINE    ;GO SELECT A LINE TO TEST
4531 015170      JSR     12S         ;
(1) 015170 000522      BR      6S          ;BR IF ALL SELECTED LINES TESTED
4532 015172 004767 006000      JSR     PC,CLCABC    ;GO CLEAR "CAR" AND "BCR" MEMORIES
4533 015176 116711 007742      MOV     LINE,(R1)   ;SET SELECT BITS IN SCR REG
4534 015202 012761 177777 000010      MOV     #1,BCR(R1)  ;SET UP TO XFER ONE CHAR
4535 015210 010561 000006      MOV     R5,CAR(R1)  ;SET UP THE BUS ADDRESS REG
4536 015214 162761 000002 000006      SUB     #2,CAR(R1)  ;CORRECT BUS ADDRESS
4537 015222 012767 000100 163766      MOV     #100,STMP7  ;COUNT 64 CHARS TO BE RECEIVED IN AUTO ECHO
4538 015230 005067 163760      CLR     STMP6        ;INIT CHAR COUNTER
4539 015234 012761 133503 000004      MOV     #133503,LPR(R1);SET UP LINE PARAMETER REG
4540 015242 016761 007144 000012      MOV     LINMSK,BAR(R1);ACTIVATE THE LINE
4541
4542 015250 012767 000002 007714      MOV     #2,TIMEA     ;INIT TIMER A
4543 015256 005067 007712      CLR     TIMEB        ;INIT TIMERB
4544 015262 105711      TSTB   (R1)         ;CHAR AVAIL SET ??
4545 015264 100427      BMI    4S           ;BR IF YES
4546 015266 004767 006660      JSR     PC,TIMEIT   ;CALL THE TIMER
4547 015272 000773      BR     3S           ;TIMER ROUTINE WILL MOVE RETURN PC
4548
4549
4550 015274 004767 007034      JSR     PC,SAPS     ;SAVE THE ERROR PSM
4551 015300 005061 000004      CLR     LPR(R1)     ;TURN OFF AUTO ECHO MODE
4552 015304 010102      MOV     R1,R2       ;MAKE REGADR = DEVRDR
4553 015306 011103      MOV     (R1),R3     ;GET THE WAS DATA
4554 015310 042703 100000      BIC    #BIT15,R3   ;CLEAR JUNK BIT
4555 015314 012774 000200      MOV     #200,R4     ;SET UP S/B DATA
4556 015320 156704 007620      BISB   LINE,R4

```

```

7402 "OR" GATE CHIPS E38 OR E41
74157 MUX CHIPS E39 OR 342
E35 - PIN 2 STUCK LOW
SH4
M7289 SH3 AE GO L EK1
AE SCAN MUX E22 PIN 10
SH4 SAMPLE STATUS H E21-12
M7288 SH5, 7, 9, 11
AE ENABLE "NM" H CONTROL FLOPS
74174 CHIPS PIN 15

```

```

4557 015324 004767 005346      JSR    PC,SUER2A      ;GO SET UP ERROR INFO
4558 015330 004567 005562      JSR    RS,SUNUM      ;GO SET LINE NO. IN MSG
4559 015334 025144              LINE
4560 015336 027116      EM24+35
4561 015340 104024      ERROR    24          ;DATA AVAIL FAILED TO SET ON TIME
4562 015342 000677      BR      15          ;GO TRY NEXT LINE
4563
4564 015344 005267 163644      45:    INC    STMP6      ;COUNT ONE CHAR RECVD
4565 015350 016103 000002      MOV    NRC(R1),R3    ;GET THE WAS DATA
4566 015354 020304      CMP    R3,R4         ;WAS CHAR AUTO ECHOED CORRECTLY ?
4567 015356 001417      BEQ    55            ;BR IF YES
4568
4569 015360 004767 006750      JSR    PC,SAPS       ;SAVE THE ERROR PSM
4570 015364 005061 000004      CLR    LPR(R1)      ;DISABLE AUTO ECHO
4571 015370 010102      MOV    R1,R2        ;SET UP REGADR
4572 015372 062702 000002      ADD    #NRC,R2
4573 015376 004767 005274      JSR    PC,SUER2A    ;GO SET UP ERROR INFO
4574 015402 004567 005510      JSR    RS,SUNUM    ;PUT LINE NO. IN ERROR MSG
4575 015406 025144              LINE
4576 015410 027116      EM24+35
4577 015412 104024      ERROR    24          ;CHAR AUTO ECHOED INCORRECTLY
4578 015414 000652      BR      15          ;GO TRY NEXT LINE
4579
4580 015416 005367 163574      55:    DEC    STMP7      ;COUNT ONE CHAR READ OUT OF 64
4581 015422 003317      BGT    35            ;BR IF NOT LAST ONE
4582 015424 100646      BMI    15            ;BR IF LAST ONE READ
4583 015426 042761 100000 000004      BIC    #BIT15,LPR(R1) ;DISABLE AUTO ECHO
4584 015434 000712      BR      35            ;GO READ LAST CHAR
4585
4586 015436 005167 163544      65:    COM    STMP3      ;TOGGLE I/O FLAG
4587 015442 001406      BEQ    TST50        ;BR IF DONE BOTH I/O DATA
4588 015444 005067 007474      CLR    LINE         ;INIT LINE NO TO 00
4589 015450 012705 024556      MOV    #AETAB0,R5   ;SET POINTER TO 0'S TABLE
4590 015454 000167 177452      JMP    75            ;REPEAT TEST FOR ZERO PATTERNS
4591

```

```

4594 ;*****
      (3) ;*TEST 50 AUTO ECHO TEST 2 - ALL LINES
      (3) ;*****
      (2) 015460 000004 TSTSD: SCOPE
      (1) 015462 012767 000050 163544 MOV #STN-1,STESTN ;;SET TEST NUMBER IN MAIL BOX
4595 .REM %
      (1) TEST ABSTRACT:
      (1) *****
      (1)
      (1) THIS TEST IS SIMILAR TO TEST 47 EXCEPT ALL SELECTED LINES OTHER
      (1) THAN THE A.E. TEST LINE ARE ACTIVELY TURNING AROUND A BINARY COUNT
      (1) TEST PATTERN IN NON-AUTO ECHO MODE AND THE A.E. TEST LINE IS TESTED
      (1) FOR ALL 1'S DATA ONLY.
      (1)
      (1) ERRORS:
      (1) *****
      (1)
      (1) 1. [ERROR 32] IS CALLED TO REPORT A.E. TEST TIMEOUTS
      (1) 2. [ERROR 31] IS CALLED TO REPORT ALL DATA COMPARE ERRORS
      (1)
      (1) SYNC: M7277 SH4 LOAD BAR LB+HB L CN2
      (1) *****
      (1)
      (1) DEBUG:
      (1) *****
      (1)
      (1) REFER TO TEST 47
      (1)
      (1) KEY LOGIC:
      (1) *****
      (1)
      (1) REFER TO TEST 47
      (1)
      (1) %
4596 015470 012767 015544 163412 MOV #25,SLPERR ;SET UP ERROR LOOP RETURN
4597 015476 012705 024516 MOV #AETAB,RS ;SET POINTER TO A.E. TEST DATA TABLE
4598 015502 005067 006706 CLR LMSK1 ;INIT BIT TEST MASK
4599 015506 000261 SEC ;GENERATOR MARKER BIT IN "C"
4600 015510 000401 BR 125 ;GO SHIFT MASK
4601 015512 000241 15: CLC ;INIT THE "C" BIT
4602 015514 006167 006674 125: ROL LMSK1 ;SHIFT TEST BIT
4603 015520 001410 BEQ 115 ;BR IF TESTED ALL LINES
4604 015522 012567 163470 MOV (RS)+,STMP7 ;GET THE A.E. TEST DATA FOR THIS LINE
4605 015526 036767 006662 006654 BIT LMSK1,LINSEL ;TEST THIS LINE ?
4606 015534 001766 BEQ 15 ;BR IF NOT
4607 015536 004767 005262 JSR PC,SELINE ;GO SELECT A LINE
4608 015542 115: BR TST51 ;;BR IF DONE ALL SELECTED LINES
      (2) 015542 000571
4609
4610 015544 016701 006630 25: MOV DHADR,R1 ;RESET DEVADR IN CASE OF ERROR LOOP
4611 015550 012711 004000 MOV #BIT11,(R1) ;CLEAR OUT THE DH11
4612 015554 004767 006414 JSR PC,SETALL ;GO SET UP FOR BINARY COUNT XFER ON
4613 ;ALL LINES OTHER THAN THE SELECTED ONE
4614 015560 156711 007360 BISB LINE,(R1) ;SELECT THE LINE FOR A.E. TEST

```

4615	015564	010561	000006		MOV	R5,CAR(R1)	:SET BUS ADDR TO XMIT TEST CHAR	
4616	015570	162761	000002	000006	SUB	#2,CAR(R1)	:CORRECT THE ADDRESS	
4617	015576	012761	177777	000010	MOV	#-1,BCR(R1)	:XMIT ONE CHAR ON THIS LINE	
4618	015604	012761	133503	000004	MOV	#133503,LPR(R1)	:DO IT AT 9600 BAUD/8 BITS	
4619	015612	116767	163264	163364	MOVB	\$STNM,\$TMP2	:SAVE THE TEST NO.	
4620	015620	042767	177400	163356	BIC	#177400,\$TMP2		
4621	015626	046767	006560	007022	BIC	LINMSK,LINACT	:MAKE THIS LINE APPEAR INACTIVE	
4622	015634	016761	006550	000012	MOV	LINSEL,BAR(R1)	:ACTIVATE ALL SELECTED TRANSMITTERS	
4623								
4624	015642	012767	000002	007322	21S:	MOV	#2,TIMEA	:INIT TIMER A
4625	015650	005067	007320			CLR	TIMEB	:INIT TIMER B
4626	015654	016103	000002		3S:	MOV	NRC(R1),R3	:GET THE WAS DATA
4627	015660	100414				BMI	4S	:BR IF YES
4628	015662	004767	006264			JSR	PC,TIMEIT	:CALL THE TIMER
4629	015666	000772				BR	3S	:TIMER ROUTINE WILL MOVE RETURN PC
4630								:AROUND THIS BRANCH IF TIME OUT OCCURS
4631								
4632	015670	016167	000004	163302		MOV	LPR(R1),\$TMP0	:SAVE THE CURRENT "LPR"
4633	015676	004567	005214			JSR	RS,SUNUM	:PUT LINE NO. IN MESSAGE
4634	015702	025144				LINE		
4635	015704	027607				EM32+3S		
4636	015706	104032				ERROR	32	:AUTO ECHO TIMEOUT
4637	015710	000700				BR	1S	:GO TRY NEXT LINE
4638								
4639	015712	010304			4S:	MOV	R3,R4	:EXTRACT LINE NUMBER OF RCVD CHAR
4640	015714	000304				SWAB	R4	
4641	015716	042704	177760			BIC	#177760,R4	
4642	015722	010402				MOV	R4,R2	:SAVE IT IN R2
4643	015724	006302				ASL	R2	:GENERATE TABLE INDEX IN R2
4644	015726	126704	007212			CMPB	LINE,R4	:IS THIS THE A.E. TEST LINE ??
4645	015732	001426				BEQ	5S	:BR IF YES
4646								
4647	015734	026203	032436			CMP	RBUF(R2),R3	:RCVD DATA CORRECT ??
4648	015740	001447				BEQ	6S	:BR IF IT WAS
4649								
4650	015742	004767	006366			JSR	PC,SAPS	:SAVE THE ERROR PSW
4651	015746	010467	163240			MOV	R4,\$TMP5	:SAVE THE LINE NUMBER
4652	015752	016204	032436			MOV	RBUF(R2),R4	:SET UP S/B DATA
4653	015756	062702	032436			ADD	#RBUF,R2	:SET UP S/B ADDRESS
4654	015762	012701	177703			MOV	#177703,R1	:SET UP THE WAS ADDRESS
4655	015766	004767	004704			JSR	PC,SUER2A	:GO SET UP ERROR INFO
4656	015772	004567	005120			JSR	RS,SUNUM	:PUT LINE NO. IN MESSAGE
4657	015776	001212				\$TMP5		
4658	016000	027452				EM31+4S		
4659	016002	104031				ERROR	31	:NON-ECHO DATA COMPARE ERROR
4660	016004	000167	177502			JMP	1S	:GO TRY NEXT LINE
4661								
4662								
4663	016010	020367	163202		5S:	CMP	R3,\$TMP7	:CHAR ECHOED OK ??
4664	016014	001427				BEQ	7S	:BR IF YES
4665								
4666	016016	004767	006312			JSR	PC,SAPS	:SAVE THE ERROR PSW
4667	016022	012702	001216			MOV	#\$TMP7,R2	:SAVE THE S/B ADDRESS
4668	016026	016704	163164			MOV	\$TMP7,R4	:SAVE THE S/B DATA

4669	016032	012701	177703		MOV	#177703,R1	:SAVE THE WAS ADDRESS
4670	016036	004767	004634		JSR	PC,SUER2A	:GO SET UP ERROR INFO
4671	016042	004567	005050		JSR	RS,SUNUM	:GO SET UP LINE NO. IN MESSAGE
4672	016046	025144			LINE		
4673	016050	027452			EM31+45		
4674	016052	104031			ERROR	31	:AUTO ECHO LINE DATA ERROR
4675	016054	000167	177432		JMP	1\$:GO TRY NEXT LINE
4676							
4677	016060	105262	032436	6\$:	INCB	RBUF(R2)	:GENERATE NEXT EXPECTED DATA ON THIS LINE
4678	016064	001266			BNE	21\$:BR IF ITS NOT BACK TO 000
4679	016066	046267	024616	006562	BIC	LINBIT(R2),LINACT	:INDICATE THIOS LINE DONE 256 BYTES
4680	016074	005767	006556	7\$:	TST	LINACT	:ALL LINES INACTIVE
4681	016100	001260			BNE	21\$:BR IF NOT
4682	016102	042761	100000	000004	BIC	#BIT15,LPR(R1)	:TURN OFF THE A.E. BIT
4683	016110	105761	000017		TSTB	SSR+1(R1)	:SILO EMPTY ??
4684	016114	001002			BNE	8\$:BR IF NOT
4685	016116	000167	177370		JMP	1\$:GO TEST NEXT LINE
4686	016122	000167	177514	8\$:	JMP	21\$:GO EMPTY IT

```

4689 ;*****
(3) ;*TEST 51 AUTO ECHO TEST 3 - ALL LINES
(3) ;*****
(2) 016126 000004 TST51: SCOPE
(1) 016130 012767 000051 163076 MOV #STN-1,STESTN ;;SET TEST NUMBER IN MAIL BOX
4690 .REN X

```

```

(1) TEST ABSTRACT:
(1) *****
(1)
(1) THIS TEST IS IDENTICAL TO TEST 47 EXCEPT ALL SELECTED LINES
(1) ARE ACTIVATED CONCURRENTLY RATHER THAN ONE AT A TIME AND ONLY
(1) THE ALL 1'S DATA IS USED.

```

```

(1) ERRORS:
(1) *****
(1)
(1) 1. [ERROR 36] IS CALLED TO REPORT "DATA AVAIL" TIMEOUTS
(1) 2. [ERROR 31] IS CALLED TO REPORT A.E. DATA ERRORS

```

```

(1) SYNC: M7277 SH4 LOAD BAR LB+HB L CN2
(1) *****

```

```

(1) DEBUG:
(1) *****

```

```

(1) REFER TO TEST 47

```

```

(1) KEY LOGIC:
(1) *****

```

```

(1) REFER TO TEST 47

```

```

(1) X
4691 016136 012767 016144 162744 MOV #15,SLPERR ;SET UP THE ERROR LOOP RETURN
4692 016144 012711 004000 1S: MOV #BIT11,(R1) ;CLEAR OUT THE DM11
4693 016150 012705 000020 MOV #20,R5 ;INIT COUNTER TO SET UP 16. LINES
4694 016154 012702 024516 MOV #AETAB,R2 ;SET UP POINTER TO AUTO ECHO TEST DATA
4695 016160 012703 032376 MOV #ACNT,R3 ;R3 POINTS TO TABLE OF CHAR COUNTERS
4696 016164 010261 000006 2S: MOV R2,CA(R1) ;SET UP BUS ADDRESS REG
4697 016170 012761 177777 000010 MOV #-1,BCR(R1) ;SET UP BYTE COUNT REG
4698 016176 012761 131403 000004 MOV #131403,LPR(R1) ;SET UP LINE PARAMETERS
4699 016204 005023 CLR (R3)+ ;CLEAR A COUNTER
4700 016206 062702 000002 ADD #2,R2 ;UPDATE POINTERS
4701 016212 005211 INC (R1) ;SELECT NEXT LINE
4702 016214 005305 DEC R5 ;COUNT ONE DONE
4703 016216 001362 BNE 2S ;BR TILL 16. DONE
4704 016220 116767 162656 162756 MOVB $STNM,STMP2 ;SAVE THE TEST NO.
4705 016226 042767 177400 162750 BIC #177400,STMP2
4706 016234 016767 006150 006414 MOV LINSEL,LINACT ;SET FLAG TO INDICATE ALL 16. ACTIVE
4707 016242 016761 006142 000012 MOV LINSEL,BAR(R1) ;ACTIVATE ALL XMITTERS
4708
4709 016250 012767 000002 006714 MOV #2,TIMEA ;INIT TIMER A
4710 016256 005067 006712 CLR TIMEB ;INIT TIMERB
4711 016262 105711 3S: TSTB (R1) ;CHAR AVAIL SET YET ?

```

4712	016264	100410				BNI	45		:BR IF YES
4713	016266	004767	005660			JSR	PC, TIMEIT		:CALL THE TIMER
4714	016272	000773				BR	35		:TIMER ROUTINE WILL MOVE RETURN PC
4715									:AROUND THIS BRANCH IF TIME OUT OCCURS
4716									
4717	016274	016167	000004	162676		MOV	LPR(R1), STMP0		:SAVE THE "LPR" REG
4718	016302	104036				ERROR	36		:DATA AVAILABLE TIMEOUT
4719	016304	000453				BR	TST52		:EXIT TEST ON ERROR
4720									
4721	016306	016103	000002		45:	MOV	NRC(R1), R3		:GET THE WAS DATA
4722	016312	010302				MOV	R3, R2		:BUILD AND SAVE LINE NO.
4723	016314	000302				SWAB	R2		
4724	016316	042702	177760			BIC	#177760, R2		
4725	016322	010267	162666			MOV	R2, STMP6		:SAVE THE LINE NO.
4726	016326	006302				ASL	R2		:GENERATE TABLE OFFSET
4727	016330	005262	032376			INC	RCNT(R2)		:COUNT THE CHARACTER
4728	016334	020362	024516			CMP	R3, AETAB(R2)		:IS THE DATA CORRECT ??
4729	016340	001420				BEQ	55		:BR IF YES
4730									
4731	016342	004767	005766			JSR	PC, SAPS		:SAVE THE ERROR PSW
4732	016346	016204	024516			MOV	AETAB(R2), R4		:GET THE S/B DATA
4733	016352	062702	024516			ADD	#AETAB, R2		:GENERATE S/B ADDRESS
4734	016356	062701	000002			ADD	#NRC, R1		:GENERATE THE WAS ADDRESS
4735	016362	004767	004310			JSR	PC, SUER2A		:GO SET UP ERROR INFO
4736	016366	004567	004524			JSR	RS, SUNUM		:PUT LINE NO. IN MESSAGE
4737	016372	001214				STMP6			
4738	016374	027452				EM31+45			
4739	016376	104031				ERROR	31		:DATA COMPARE ERROR
4740	016400	000415				BR	TST52		:EXIT TEST ON ERROR
4741									
4742	016402	022762	000100	032376	55:	CMP	#100, RCNT(R2)		:DONE 64. CHARS ON THIS LINE ?
4743	016410	001324				BNE	35		:BR IF NOT
4744	016412	016711	162576			MOV	STMP6, (R1)		:SELECT LINE IN SCR REG
4745	016416	042761	100000	000004		BIC	#BIT15, LPR(R1)		:TURN OFF A.E. BIT
4746	016424	046267	024616	006224		BIC	LINBIT(R2), LINACT		:ALL LINES INACTIVE ??
4747	016432	001313				BNE	35		:BR IF NOT

M7278 SHS THRU SH8

74175 REGISTER CHIPS E51, E38, E67, E60
7400 DRIVERS E45, E46, E75, E76

```

(1)
(1)
(1)
(1)
(1)
(1)
4752 016444 012767 016524 162436 X MOV #28,SLPERR ;SET UP ERROR LOOP RETURN
4753 016452 012705 024720 MOV #BRKTAB,RS ;SET UP POINTER TO BREAK DATA TABLE
4754 016456 005067 005732 CLR LMSK1 ;INIT BIT TEST MASK
4755 016462 000261 SEC ;SET BIT MARKER IN "C"
4756 016464 000401 BR 128 ;GO SHIFT MASK
4757 016466 000241 18: CLC ;INIT THE "C" BIT
4758 016470 006167 005720 128: ROL LMSK1 ;SHIFT TEST MARKER
4759 016474 001411 BEQ 118 ;BR IF ALL LINES DONE
4760 016476 012504 MOV (RS)+,R4 ;GET TEST DATA FOR THIS LINE
4761 016500 036767 005710 005702 BIT LMSK1,LINSEL ;LINE SELECTED ?
4762 016506 001767 BEQ 18 ;BR IF NOT
4763 016510 004767 004310 JSR PC,SELINE ;GO SELECT A LINE TO TEST
4764 016514 000401 BR 118 ;BR IF DONE ALL SELECTED LINES
4765 016516 000402 BR 28 ;GO TEST THE SELECTED LINE
4766 016520 000167 000454 118: JMP 98 ;GO EXIT TEST
4767 016524 012711 004000 28: MOV #BIT11,(R1) ;CLEAR THE DM11
4768 016530 004767 004442 JSR PC,CLCABC ;GO CLR THE "CAR" AND "BCR" MEMORIES
4769 016534 116711 006404 MOV#B LINE,(R1) ;SELECT THE LINE
4770
4771 016540 012761 025200 000006 MOV #NULL,CAR(R1) ;SET UP TO OUTPUT TWO NULL CHARS
4772 016546 012761 177776 000010 MOV #2,BCR(R1) ;SET BYTE COUNT TO 2
4773 016554 012761 033503 000004 MOV #33503,LPR(R1) ;SET UP LINE PARAMETERS
4774 016562 016761 005624 000012 MOV LINMSK,BAR(R1) ;ACTIVATE SELECTED LINE
4775
4776 016570 012767 000001 006374 MOV #1,TIMEA ;INIT TIMER A
4777 016576 005067 006372 CLR TIMEB ;INIT TIMER B
4778 016602 122761 000002 000017 38: CMPB #2,SSR+1(R1) ;TWO CHARS RECEIVED ??
4779 016610 001432 BEQ 48 ;BR IF YES
4780 016612 004767 005334 JSR PC,TIMEIT ;CALL THE TIMER
4781 016616 000771 BR 38 ;TIMER ROUTINE WILL MOVE RETURN PC
4782 ; AROUND THIS BRANCH IF TIME OUT OCCURS
4783
4784 016620 004767 005510 JSR PC,SAPS ;SAVE THE ERROR PSM
4785 016624 010467 162352 MOV R4,$TMP1 ;SAVE S/B DATA
4786 016630 010102 MOV R1,R2 ;SET UP REGADR
4787 016632 062702 000016 ADD #SSR,R2
4788 016636 011203 MOV (R2),R3 ;GET THE WAS DATA
4789 016640 042703 100377 BIC #100377,R3 ;CLEAR JUNK
4790 016644 012704 000002 MOV #2,R4 ;SET UP S/B DATA
4791 016650 000304 SWAB R4
4792 016652 004767 004020 JSR PC,SUER2A ;GO SET UP ERROR INFO
4793 016656 004567 004234 JSR RS,SUNUM ;GO PUT LINE NO. IN MESSAGE
4794 016662 025144 LINE
4795 016664 027155 EM25+34
4796 016666 016704 162310 MOV $TMP1,R4 ;RESTORE S/B DATA
4797 016672 104025 ERROR 25 ;TIMED OUT WAITING FOR TWO NULLS
4798 016674 000674 BR 18 ;GO TRY NEXT LINE
4799

```

4800	016676	012711	004000		4S:	MOV	#BIT11,(R1)	:CLEAR THE INTERFACE
4801	016702	116711	006236			MOV	LINE,(R1)	:SELECT THE LINE
4802	016706	012761	033436	000006		MOV	#TBUF,CAR(R1)	:SET UP BUS ADDRESS REG FOR XMITTR
4803	016714	012761	177400	000010		MOV	#-400,BCR(R1)	:SET BYTE COUNT TO XMIT 256(10) CHARS
4804	016722	012761	033503	000004		MOV	#33503,LPR(R1)	:SET UP LINE PARAMETERS
4805	016730	016761	005456	000014		MOV	LINMSK,BKR(R1)	:SET BREAK BIT FOR ACTIVE LINE
4806	016736	016761	005450	000012		MOV	LINMSK,BAR(R1)	:ACTIVATE THE SELECTED LINE
4807								
4808	016744	012767	000005	006220		MOV	#5,TIMEA	:INIT TIMER A
4809	016752	005067	006216			CLR	TIMEB	:INIT TIMER B
4810	016756	005761	000012		5S:	TST	BAR(R1)	:BAR BIT CLEARED ??
4811	016762	001426				BEQ	6S	:BR IF 0 YES
4812	016764	004767	005162			JSR	PC,TIMEIT	:CALL THE TIMER
4813	016770	000772				BR	5S	:TIMER ROUTINE WILL MOVE RETURN PC
4814								:AROUND THIS BRANCH IF TIME OUT OCCURS
4815								
4816	016772	004767	005336			JSR	PC,SAPS	:SAVE THE ERROR PSW
4817	016776	010467	162200			MOV	R4,\$TMP1	:SAVE THE S/B DATA
4818	017002	010102				MOV	R1,R2	:SET UP REGADR
4819	017004	062702	000012			ADD	#BAR,R2	
4820	017010	011203				MOV	(R2),R3	:GET THE WAS DATA
4821	017012	005004				CLR	R4	:SET UP S/B DATA
4822	017014	004767	003656			JSR	PC,SUER2A	:GO SET UP ERROR INFO
4823	017020	004567	004072			JSR	RS,SUNUM	:PUT LINE NO IN MESSAGE
4824	017024	025144				LINE		
4825	017026	027155				EM25+34		
4826	017030	016704	162146			MOV	\$TMP1,R4	:RESTORE THE S/B DATA
4827	017034	104025				ERROR	25	:BAR BIT FAILED TO CLEAR
4828	017036	000613				BR	1S	:GO TRY NEXT LINE
4829								
4830	017040	122761	000001	000017	6S:	CMPS	#1,SSR+1(R1)	:ONE CHAR RECEIVED ?
4831	017046	001430				BEQ	7S	:BR IF YES
4832								
4833	017050	004767	005260			JSR	PC,SAPS	:SAVE THE ERROR PSW
4834	017054	010467	162122			MOV	R4,\$TMP1	:SAVE THE S/B DATA
4835	017060	010102				MOV	R1,R2	:SET UP REGADR
4836	017062	062702	000016			ADD	#SSR,R2	
4837	017066	011203				MOV	(R2),R3	:GET THE WAS DATA
4838	017070	042703	100377			BIC	#100377,R3	:CLEAR JUNK
4839	017074	012704	000001			MOV	#1,R4	:SET UP S/B DATA
4840	017100	000304				SWAB	R4	
4841	017102	004767	003570			JSR	PC,SUER2A	:GO SET UP ERROR INFO
4842	017106	004567	004004			JSR	RS,SUNUM	:GO PUT LINE NO. IN MESSAGE
4843	017112	025144				LINE		
4844	017114	027155				EM25+34		
4845	017116	016704	162060			MOV	\$TMP1,R4	:RESTORE THE S/B DATA
4846	017122	104025				ERROR	25	:FAILED TO RECEIVE THE ONE CHAR
4847	017124	000167	177336			JMP	1S	:GO TRY NEXT LINE
4848								
4849								
4850	017130	016103	000002		7S:	MOV	NRC(R1),R3	:GET THE WAS DATA
4851	017134	020304				CMPS	R3,R4	:WAS IT A BREAK CHAR ?
4852	017136	001002				BNE	8S	:BR IF NOT CORRECT
4853	017140	000167	177322			JMP	1S	:GO TEST NEXT LINE

4854								
4855	017144	004767	005164	88:	JSR	PC, SAPS		:SAVE THE ERROR PSW
4856	017150	010102			MOV	R1, R2		:SET UP REGADR
4857	017152	062702	000002		ADD	#NAC, R2		
4858	017156	004767	003514		JSR	PC, SUER2A		:GO SET UP ERROR INFO
4859	017162	004567	003730		JSR	RS, SUNUM		:PUT LINE NO IN MESSAGE
4860	017166	025144			LINE			
4861	017170	027155			EM25+34			
4862	017172	104025			ERROR	25		:INCORRECT DATA RECEIVED
4863	017174	000167	177266		JMP	15		:GO TRY NEXT LINE
4864	017200	000240		99:	NOP			:EXIT THIS TEST
4865								

4874	017232	004767	003740		JSR	PC,CLCABC	:GO CLR THE "CAR" AND "BCR" MEMORIES
4875	017236	156711	005702		BISB	LINE,(R1)	:SELECT THE LINE
4876	017242	012761	033436	000006	MOV	STBUF,CAR(R1)	:POINT TO XMIT BUFFER
4877	017250	012761	177400	000010	MOV	8-400,BCR(R1)	:XMIT 256(10) CHARS
4878	017256	012761	073503	000004	MOV	873503,LPR(R1)	:SET UP THE LINE PARAMETERS
4879	017264	016761	005122	000012	MOV	LINMSK,BAR(R1)	:ACTIVATE THE SELECTED LINE
4880							
4881	017272	012767	000001	005672	MOV	81,TIMEA	:INIT TIME A
4882	017300	005067	005670		CLR	TIMEB	:INIT TIME B
4883	017304	005761	000012		35: TST	BAR(R1)	:WAIT FOR XMITTR TO FINISH
4884	017310	001423			BEG	48	:BR IF XMITTR FINISHED
4885	017312	004767	004634		JSR	PC,TIMEIT	:CALL TIMER
4886	017316	000772			BR	35	:TIMER WILL MOVE RETURN PC AROUND :THIS BRANCH IF TIMEOUT OCCURS
4887							
4888							
4889	017320	004767	005010		JSR	PC,SAPS	:SAVE THE ERROR PSM
4890	017324	016103	000012		MOV	BAR(R1),R3	:GET THE WAS DATA
4891	017330	010102			MOV	R1,R2	:SET UP REGADR
4892	017332	062702	000012		ADD	8BAR,R2	
4893	017336	005004			CLR	R4	:SET UP NEW S/B DATA
4894	017340	004767	003332		JSR	PC,SUER2A	:GO SET UP THE ERROR INFO
4895	017344	004567	003546		JSR	RS,SUNUM	:PUT LINE NO. IN MESSAGE
4896	017350	025144			LINE		
4897	017352	027220			EM26+37		
4898	017354	104026			ERROR	26	:BAR BIT FAILED TO CLEAR ON TIME
4899	017356	000720			BR	15	:GO TRY NEXT LINE
4900							
4901	017360	105711			45: TSTB	(R1)	:CHAR AVAIL SET ??
4902	017362	100316			BPL	15	:BR IF NOT IT SHOULDN'T BE
4903							
4904	017364	004767	004744		JSR	PC,SAPS	:SAVE THE ERROR PSM
4905	017370	010102			MOV	R1,R2	:SET UP REGADR
4906	017372	011103			MOV	(R1),R3	:GET WAS DATA
4907	017374	042703	100000		BIC	8BIT15,R3	:CLEAR JUNK BIT
4908	017400	116704	005540		NOVB	LINE,R4	:SET UP S/B DATA
4909	017404	004767	003266		JSR	PC,SUER2A	:GO SETUP ERROR INFO
4910	017410	004567	003502		JSR	RS,SUNUM	:PUT LINE NO. IN MSG
4911	017414	025144			LINE		
4912	017416	027220			EM26+37		
4913	017420	104026			ERROR	26	:HALF DUPLEX FAILED TO BLIND RECVR
4914	017422	000676			BR	15	:GO SELECT NEXT LINE

4917
(3)
(3)
(2) 017424 000004
(1) 017426 012767 000054 161600
4918

: #TEST 54 VERIFY THAT OVERRUN CAN SET PROPERLY - ALL LINES
: *****

↑ST54: SCOPE
MOV #STN-1,STESTN ;;SET TEST NUMBER IN MAIL BOX

REM X
TEST ABSTRACT:

THIS TEST VERIFIES THAT "OVERRUN" SETS PROPERLY FOR ALL LINES
THAT ARE SELECTED FOR TEST WHEN THE OVERRUN CONDITION IS FORCED BY THE
PROGRAM. THE TEST SEQUENCE IS AS FOLLOWS:

1. SET UP THE ERROR LOOP RETURN
2. SELECT A LINE NO. TO TEST - IF DONE ALL LINES GO TO
END OF PASS HANDLER.
3. PRIME THE SELECTED LINE TO XMIT 68. CHARS
4. ACTIVATE THE SELECTED LINE
5. WAIT FOR "XMIT DONE" TO SET - IF TIMEOUT REPORT ERROR
AND RESTART AT STEP 2
6. IF NO TIMEOUT READ 65. CHARS FROM THE SILO AND VERIFY THAT
"OVERRUN" IS SET ON THE LAST WORD READ
7. IF NOT REPORT ERROR AND RESTART AT STEP 2

ERRORS:

1. [ERROR 50] IS CALLED TO REPORT "XMIT DONE " TIMEOUTS
2. [ERROR 56] IS CALLED TO REPORT "OVERRUN" ERROR

SYNC: M7277 SH3 INIT A H EF2

DEBUG:

1. IF FAULT APPEARS ON ONLY ONE LINE SUSPECT UART MODULE
FOR THE APPROPRIATE LINE IN QUESTION.
2. IF FAULT APPEARS ON ALL LINES SUSPECT THE M7279 MODULE

KEY LOGIC:

M7279	SH1	MASTER OR H	E12-9
	SH2	MEMORY CHIP (3341)	E13-11
M7280	SH2	UC1 OR 2 MASTER OR	EN2
	SH2-5	UART PIN 15 (BUF OR LINE NN)	

%

4919 017434 012767 017450 161446
4920 017442 004767 003356

MOV #25,\$LPERR ;SET UP ERROR LOOP RETURN
15: JSR PC,\$ELINE ;GO SELECT A LINE # TO TEST

H03

MAINDEC-11-DZDMM-A
DZDMM.P11 T54

MACY11 27(663) 12-DEC-75 08:41 PAGE 91-1
VERIFY THAT OVERRUN CAN SET PROPERLY - ALL LINES

SEQ 0239

```

4921 017446 000512          BR      ENDA          ;; BR IF DONE ALL SELECTED LINES
4922 017450 012711 004000    2S:    MOV      #BIT11,(R1)  ;; CLEAR OUT THE DM11
4923 017454 116711 005464          MOVB     LINE,(R1)    ;; SELECT THE LINE TO TEST
4924 017460 012761 033436 000006    MOV      #TBUF,CAR(R1) ;; SET UP CURRENT ADDRESS
4925 017466 012761 177674 000010    MOV      #68,BCR(R1)  ;; SET UP BYTE COUNT REG
4926 017474 012761 033503 000004    MOV      #33503,LPR(R1) ;; DO IT AT 9600 BAUD - 8 BITS
4927 017502 016761 004704 000012    MOV      LINMSK,BAR(R1) ;; ACTIVATE THE SELECTED LINE
4928
4929 017510 012767 000001 005454    MOV      #1,TIMEA     ;; INIT TIMERS A AND B
4930 017516 005067 005452          CLR      TIMEB
4931 017522 005711          3S:    TST      (R1)        ;; TRANSMITTER DONE ??
4932 017524 100425          BMI     4S           ;; BR IF YES
4933 017526 004767 004420    JSR     PC,TIMEIT    ;; CALL TIMER
4934 017532 000773          BR      3S          ;; BR IF NO TIMEOUT
4935
4936 017534 004767 004574    JSR     PC,SAPS      ;; GO SAVE PSW
4937 017540 011103          MOV     (R1),R3     ;; GET THE WAS DATA
4938 017542 042703 077760    BIC     #77760,R3   ;; CLEAR UNINTERESTING BITS
4939 017546 116704 005372    MOVB     LINE,R4    ;; SET UP S/B DATA
4940 017552 052704 100000    BIS     #BIT15,R4
4941 017556 010102          MOV     R1,R2      ;; SET UP REGADR
4942 017560 004767 003112    JSR     PC,SUER2A   ;; GO SET UP ERROR INFO
4943 017564 004567 003326    JSR     RS,SUNUM    ;; PUT LINE NO. IN MESSAGE HEADER
4944 017570 025144          LINE
4945 017572 031076          EMS0+53
4946 017574 104050          ERROR  50
4947 017576 000721          BR      1S          ;; REPORT XMIT DONE TIME OUT
4948
4949 017600 012767 000101 161374  4S:    MOV      #65.,STMP1  ;; SET UP TO READ 65. WORDS FROM SILO
4950 017606 116704 005332          MOVB     LINE,R4    ;; SET UP S/B DATA
4951 017612 000304          SWAB    R4
4952 017614 152704 000101          BISB    #65.,R4
4953 017620 052704 140000          BIS     #BIT15+BIT14,R4 ;; PUT IN OVERRUN AND VALID DATA BITS
4954 017624 016103 000002          5S:    MOV      NRC(R1),R3  ;; GET WAS DATA FROM SILO
4955 017630 005367 161346          DEC     STMP1       ;; COUNT ONE WORD READ
4956 017634 001373          BNE     5S         ;; BR TIL 65. READ
4957 017636 020304          CMP     R3,R4      ;; WAS DATA AND OVERRUN CORRECT ??
4958 017640 001700          BEQ     1S         ;; BR IF YES TRY NEXT SELECTED LINE
4959
4960 017642 004767 004466    JSR     PC,SAPS      ;; GO SAVE PSW
4961 017646 010102          MOV     R1,R2      ;; SET UP REGADR
4962 017650 062702 000002          ADD     #NRC,R2
4963 017654 004767 003016    JSR     PC,SUER2A   ;; GO SET UP ERROR INFO
4964 017660 004567 003232    JSR     RS,SUNUM    ;; GO PUT LINE NO. IN MSG HDR
4965 017664 025144          LINE
4966 017666 031514          EMS6+42
4967 017670 104056          ERROR  56
4968 017672 000663          BR      1S          ;; OVERRUN OR DATA INCORRECT
4969

```

```

4972 017674 000004          ENDA:  SCOPE
4973 017676 012767 000240 000054  MOV      #240,SEOP      ;NOP THE SCOPE AT THE BEGINNING OF EOP
4974 017704 005267 005232          INC      DHNUM        ;GENERATE NEW DH11 NUMBER
4975 017710 062767 000002 005232  ADD      #2,ADPTR     ;UPDATE THE TABLE POINTERS
4976 017716 062767 000002 005226  ADD      #2,VCPTR
4977 017724 062767 000002 005222  ADD      #2,BRPTR
4978 017732 006367 004446          ASL      SELMSK      ;SHIFT MARKER TO TEST NEXT DH11
4979 017736 001410          BEQ      SEOP        ;BR IF TESTED ALL SELECTED DH11'S
4980 017740 036767 004440 004440  BIT      SELMSK,DHSEL ;IS THIS DH11 SELECTED ?
4981 017746 001752          BEQ      ENDA       ;BR IF NOT
4982 017750 105067 161126          CLRB    $STSTNM     ;INIT TEST NUMBER
4983 017754 000167 162632          JMP     RSTRTA     ;GO TEST THIS DH11
4984 (1) ;*****
(1) .SBTTL  END OF PASS ROUTINE
(1)
(1) ;*INCREMENT THE PASS NUMBER ($PASS)
(1) ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
(1) ;*IF THERES A MONITOR GO TO IT
(1) ;*IF THERE ISN'T JUMP TO START2
(1)
(1) SEOP:
(1) 017760          SCOPE
(1) 017760 000004          CLR      $STSTNM    ;: ZERO THE TEST NUMBER
(1) 017762 005067 161114          CLR      $TIMES    ;: ZERO THE NUMBER OF ITERATIONS
(1) 017766 005067 161226          INC      $PASS     ;: INCREMENT THE PASS NUMBER
(1) 017772 005267 161240          BIC      #100000,$PASS ;: DON'T ALLOW A NEG. NUMBER
(1) 017776 042767 100000 161232  DEC      (PC)+     ;: LOOP?
(1) 020004 005327          SEOPCT: .WORD 1
(1) 020006 000001          BGT      $DOAGN    ;: YES
(1) 020010 003022          MOV      (PC)+,$(PC)+ ;: RESTORE COUNTER
(1) 020012 012737          SENDCT: .WORD 1
(1) 020014 000001          TYPE    SENDMG     ;: TYPE "END PASS #"
(1) 020016 020006          MOV      $PASS,-($P) ;: SAVE $PASS FOR TYPEOUT
(1) 020020 104400 020062          TYPDS   $SENDMG   ;: GO TYPE--DECIMAL ASCII WITH SIGN
(1) 020024 016746 161206          TYPE    ,SENULL   ;: TYPE A NULL CHARACTER
(1) 020030 104404          SGET42:
(1) 020032 104400 020077          MOV      #42,R0    ;: GET MONITOR ADDRESS
(1) 020036          BEQ      $DOAGN    ;: BRANCH IF NO MONITOR
(1) 020042 001405          RESET   ;: CLEAR THE WORLD
(1) 020044 000005          JSR     PC,(R0)    ;: GO TO MONITOR
(1) 020046 004710          NOP     ;: SAVE ROOM
(1) 020050 000240          NOP     ;: FOR
(1) 020052 000240          NOP     ;: ACT11
(1) 020054 000240          SDOAGN:
(1) 020056          JMP     #START2    ;: RETURN
(1) 020058 000137 002506          SENDMG: .ASCIZ <15><12>/END PASS #/
(1) 020062 005015 047105 020104          .BYTE  -1,-1,0    ;: NULL CHARACTER STRING
(1) 020070 040520 051523 021440          ;*****
(1) 020076          SENULL: .BYTE  -1,-1,0
(1) 020077          377      377      000
4985 (1)

```

```

(1) .SBTTL SCOPE HANDLER ROUTINE
(1)
(1) ;#THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
(1) ;#AND LOAD THE TEST NUMBER(STSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
(1) ;#AND LOAD THE ERROR FLAG (SERFLG) INTO DISPLAY<15:08>
(1) ;#THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
(1) ;#SW14=1 LOOP ON TEST
(1) ;#SW11=1 INHIBIT ITERATIONS
(1) ;#SW09=1 LOOP ON ERROR
(1) ;#SW08=1 LOOP ON TEST IN SMR<7:0>
(1) ;#CALL
(1) ;# SCOPE ;:SCOPE=IOT

(1) $SCOPE:
(1) 020102 005067 005036 CLR LINE ;:INIT THE LINE NO. TO ZERO
(3) 020102 016701 004266 MOV DHADR,R1 ;:SET UP DEVAOR IN R1
(3) 020106 032777 040000 161016 1S: BIT #BIT14,2SMR ;:LOOP ON PRESENT TEST?
(1) 020112 001114 BNE SOVER ;:YES IF SW14=1
(1) ;#####START OF CODE FOR THE XOR TESTER#####
(1) 020122 000416 $XTSTR: BR 6S ;:IF RUNNING ON THE "XOR" TESTER CHANGE
(1) ;:THIS INSTRUCTION TO A "NOP" (NOP=240)
(1) 020124 013746 000004 MOV 2#ERRVEC, -(SP) ;:SAVE THE CONTENTS OF THE ERROR VECTOR
(1) 020130 012737 020150 000004 MOV #5S,2#ERRVEC ;:SET FOR TIMEOUT
(1) 020136 005737 177060 TST 2#177060 ;:TIME OUT ON XOR?
(1) 020142 012637 000004 MOV (SP)+,2#ERRVEC ;:RESTORE THE ERROR VECTOR
(1) 020146 000463 BR $SVLAD ;:GO TO THE NEXT TEST
(1) 020150 022626 5S: CMP (SP)+,(SP)+ ;:CLEAR THE STACK AFTER A TIME OUT
(1) 020152 012637 000004 MOV (SP)+,2#ERRVEC ;:RESTORE THE ERROR VECTOR
(1) 020156 000423 BR 7S ;:LOOP ON THE PRESENT TEST
(1) 020160 6S: ;#####END OF CODE FOR THE XOR TESTER#####
(1) 020160 032777 000400 160750 BIT #BIT08,2SMR ;:LOOP ON SPEC. TEST?
(1) 020166 001404 BEQ 2S ;:BR IF NO
(1) 020170 127767 160742 160704 CMPB 2SMR,STSTNM ;:ON THE RIGHT TEST? SMR<7:0>
(1) 020176 001465 BEQ SOVER ;:BR IF YES
(1) 020200 105767 160677 2S: TSTB SERFLG ;:HAS AN ERROR OCCURRED?
(1) 020204 001421 BEQ 3S ;:BR IF NO
(1) 020206 126767 160703 160667 CMPB $ERMAX,SERFLG ;:MAX. ERRORS FOR THIS TEST OCCURRED?
(1) 020214 101015 BHI 3S ;:BR IF NO
(1) 020216 032777 001000 160712 BIT #BIT09,2SMR ;:LOOP ON ERROR?
(1) 020224 001404 BEQ 4S ;:BR IF NO
(1) 020226 016767 160656 160652 7S: MOV $LPERR,$LPADR ;:SET LOOP ADDRESS TO LAST SCOPE
(1) 020234 000446 BR SOVER
(1) 020236 105067 160641 4S: CLRB SERFLG ;:ZERO THE ERROR FLAG
(1) 020242 005067 160752 CLR $TIMES ;:CLEAR THE NUMBER OF ITERATIONS TO MAKE
(1) 020246 000415 BR 1S ;:ESCAPE TO THE NEXT TEST
(1) 020250 032777 004000 160660 3S: BIT #BIT11,2SMR ;:INHIBIT ITERATIONS?
(1) 020256 001011 BNE 1S ;:BR IF YES
(1) 020260 005767 160752 TST $PASS ;:IF FIRST PASS OF PROGRAM
(1) 020264 001406 BEQ 1S ;:INHIBIT ITERATIONS
(1) 020266 005267 160612 INC $ICNT ;:INCREMENT ITERATION COUNT
(1) 020272 026767 160722 160604 CMP $TIMES,$ICNT ;:CHECK THE NUMBER OF ITERATIONS MADE
(1) 020300 002024 BGE SOVER ;:BR IF MORE ITERATION REQUIRED
(1) 020302 012767 000001 160574 1S: MOV #1,$ICNT ;:REINITIALIZE THE ITERATION COUNTER
(1) 020310 016767 000052 160702 MOV $MXCNT,$TIMES ;:SET NUMBER OF ITERATIONS TO DO

```

```

(1) 020316 105267 160560  SSVLAD: INCB STSTNM           ;; COUNT TEST NUMBERS
(1) 020322 116767 160554 160704  MOVB STSTNM,STSTN  ;; SET TEST NUMBER IN APT MAILBOX
(1) 020330 011667 160552   MOV (SP),SLPADR   ;; SAVE SCOPE LOOP ADDRESS
(1) 020334 011667 160550   MOV (SP),SLPERR  ;; SAVE ERROR LOOP ADDRESS
(1) 020340 005067 160656   CLR SESCPE      ;; CLEAR THE ESCAPE FROM ERROR ADDRESS
(1) 020344 112767 000001 160543  MOVB #1,SERMAX  ;; ONLY ALLOW ONE(1) ERROR ON NEXT TEST
(1) 020352 016777 160524 160560  SOVER: MOV STSTNM,DISP  ;; DISPLAY TEST NUMBER
(1) 020360 016716 160522   MOV SLPADR,(SP) ;; FUDGE RETURN ADDRESS
(1) 020364 000002   RTI           ;; FIXES PS
(1) 020366 000010  SMXCNT: 10      ;; MAX. NUMBER OF ITERATIONS
4986 ;*****

(1) .SBTTL ERROR HANDLER ROUTINE
(1) ;#THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
(1) ;#SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
(1) ;#AND GO TO SERRTYP ON ERROR
(1) ;#THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
(1) ;#SW15=1 HALT ON ERROR
(1) ;#SW13=1 INHIBIT ERROR TIMEOUTS
(1) ;#SM09=1 LOOP ON ERROR
(1) ;#CALL
(1) ;# ERROR N ;;ERROR=ENT AND N=ERROR ITEM NUMBER

(1) 020370 SERROR:
(1) 020370 105267 160507 7S: INCB SERFLG           ;; SET THE ERROR FLAG
(1) 020374 001775 BEQ 7S              ;; DON'T LET THE FLAG GO TO ZERO
(1) 020376 016777 160500 160534  MOV STSTNM,DISP  ;; DISPLAY TEST NUMBER AND ERROR FLAG
(1) 020404 005267 160502   INC SERTTL      ;; INC THE ERROR COUNT
(1) 020410 011667 160502   MOV (SP),SERPC  ;; GET ADDRESS OF ERROR INSTRUCTION
(1) 020414 162767 000002 160474  SUB #2,SERPC
(1) 020422 117767 160470 160464  MOVB #SERPC,$ITEMB ;; STRIP AND SAVE THE ERROR ITEM CODE
(1) 020430 032777 020000 160500  BIT #BIT13,$SWR  ;; SKIP TIMEOUT IF SET
(1) 020436 001004 BNE 20S           ;; SKIP TIMEOUTS
(1) 020440 004767 000072   JSR PC,SERTYP   ;; GO TO USER ERROR ROUTINE
(1) 020444 104400 001225   TYPE ,SCALF

(1) 020450 20S:
(1) 020450 122767 000001 160572  CMPB #APTENV,SENV ;; RUNNING IN APT MODE
(1) 020456 001007 BNE 2S           ;; NO SKIP APT ERROR REPORT
(1) 020460 116767 160430 000004  MOVB $ITEMB,21S  ;; SET ITEM NUMBER AS ERROR NUMBER
(1) 020466 004767 001146   JSR PC,SATY4    ;; REPORT FATAL ERROR TO APT
(1) 020472 000 21S: .BYTE 0
(1) 020473 000 .BYTE 0
(1) 020474 000777 22S: BR 22S           ;; APT ERROR LOOP
(1) 020476 005777 160434 2S: TST $SWR        ;; HALT ON ERROR
(1) 020502 100001 BPL 3S          ;; SKIP IF CONTINUE
(1) 020504 000000 HALT           ;; HALT ON ERROR!
(1) 020506 032777 001000 160422 3S: BIT #BIT09,$SWR ;; LOOP ON ERROR SWITCH SET?
(1) 020514 001402 BEQ 4S           ;; BR IF NO
(1) 020516 016716 160366   MOV SLPERR,(SP) ;; FUDGE RETURN FOR LOOPING
(1) 020522 005767 160474 4S: TST SESCPE      ;; CHECK FOR AN ESCAPE ADDRESS
(1) 020526 001402 BEQ 5S          ;; BR IF NONE
(1) 020530 016716 160466   MOV SESCPE,(SP) ;; FUDGE RETURN ADDRESS FOR ESCAPE
(1) 020534 5S:

```

(1) 020534 000002
4987
(1)
(1)
(1)
(1)
(1)
(1) 020536
(1) 020536 104400 001225
(1) 020542 010046
(1) 020544 005000
(1) 020546 153700 001114
(1) 020552 001004
(2) 020554 016746 160336
(2)
(2) 020560 104401
(1) 020562 000426
(1) 020564 005300
(1) 020566 006300
(1) 020570 006300
(1) 020572 006300
(1) 020574 062700 001354
(1) 020600 012067 000004
(1) 020604 001404
(1) 020606 104400
(1) 020610 000000
(1) 020612 104400 001225
(1) 020616 012067 000004
(1) 020622 001404
(1) 020624 104400
(1) 020626 000000
(1) 020630 104400 001225
(1) 020634 011000
(1) 020636 001004
(1) 020640 012600
(1) 020642 104400 001225
(1) 020646 000207
(2) 020650
(2) 020650 013046
(2) 020652 104401
(1) 020654 005710
(1) 020656 001770
(1) 020660 104400 020666
(1) 020664 000771
(1) 020666 020040 000
(1) 020672

```

RTI                                     ::RETURN
;*****

.SBTTL ERROR MESSAGE TIMEOUT ROUTINE

;#THIS ROUTINE USES THE "ITEM CONTROL BYTE" (SITEMB) TO DETERMINE WHICH
;#ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" (SERRTB),
;#AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.

SERRTYP:
TYPE      SCRLF                         :: "CARRIAGE RETURN" & "LINE FEED"
MOV       RD, -(SP)                       :: SAVE RD
CLR       RD                                :: PICKUP THE ITEM INDEX
BISB     @SITEMB, RD
BNE      IS

; IF ITEM NUMBER IS ZERO, JUST
; TYPE THE PC OF THE ERROR
; SAVE SERRPC FOR TYPEOUT
; ERROR ADDRESS
; GO TYPE--OCTAL ASCII(ALL DIGITS)
; GET OUT
; ADJUST THE INDEX SO THAT IT WILL
; WORK FOR THE ERROR TABLE
MOV       SERRPC, -(SP)
TPOC
BR       6S
IS:      DEC      RD
ASL     RD
ASL     RD
ASL     RD
ADD     #SERRTB, RD          :: FORM TABLE POINTER
MOV     (RD)+, 2S          :: PICKUP "ERROR MESSAGE" POINTER
BEQ     3S                :: SKIP TYPEOUT IF NO POINTER
TYPE   "ERROR MESSAGE"   :: TYPE THE "ERROR MESSAGE"
WORD  0                  :: "ERROR MESSAGE" POINTER GOES HERE
SCRLF                                     :: "CARRIAGE RETURN" & "LINE FEED"
MOV     (RD)+, 4S          :: PICKUP "DATA HEADER" POINTER
BEQ     5S                :: SKIP TYPEOUT IF 0
TYPE   "DATA HEADER"     :: TYPE THE "DATA HEADER"
WORD  0                  :: "DATA HEADER" POINTER GOES HERE
SCRLF                                     :: "CARRIAGE RETURN" & "LINE FEED"
MOV     (RD), RD          :: PICKUP "DATA TABLE" POINTER
BNE     7S                :: GO TYPE THE DATA
MOV     (SP)+, RD         :: RESTORE RD
TYPE   SCRLF              :: "CARRIAGE RETURN" & "LINE FEED"
RTS     PC                :: RETURN

7S:      MOV     @ (RD)+, -(SP)           :: SAVE @ (RD)+ FOR TYPEOUT
TPOC    :: GO TYPE--OCTAL ASCII(ALL DIGITS)
TST     (RD)              :: IS THERE ANOTHER NUMBER?
BEQ     6S                :: BR IF NO
TYPE   2S                 :: TYPE TWO(2) SPACES
BR      7S                :: LOOP
BS:      .ASCIZ / /           :: TWO(2) SPACES
.EVEN

;*****

```

4988
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)

```

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

;#THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT

```

```

(1)      ;#OCTAL (ASCII) NUMBER AND TYPE IT.
(1)      ;#STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
(1)      ;#CALL:
(1)      ;#      MOV      NUM,-(SP)      ;: NUMBER TO BE TYPED
(1)      ;#      TYPOS      ;: CALL FOR TYPEOUT
(1)      ;#      .BYTE  N      ;: N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
(1)      ;#      .BYTE  M      ;: M=1 OR 0
(1)      ;#      ;: 1=TYPE LEADING ZEROS
(1)      ;#      ;: 0=SUPPRESS LEADING ZEROS
(1)      ;#STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
(1)      ;#STYPOS OR STYPOC
(1)      ;#CALL:
(1)      ;#      MOV      NUM,-(SP)      ;: NUMBER TO BE TYPED
(1)      ;#      TYPON      ;: CALL FOR TYPEOUT
(1)      ;#STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
(1)      ;#CALL:
(1)      ;#      MOV      NUM,-(SP)      ;: NUMBER TO BE TYPED
(1)      ;#      TYPOC      ;: CALL FOR TYPEOUT
(1)      020672 017646 000000      STYPOS: MOV      2(SP),-(SP)      ;: PICKUP THE MODE
(1)      020676 116667 000001 000211  MOVB     1(SP),SOFILL      ;: LOAD ZERO FILL SWITCH
(1)      020704 112667 000207      MOVB     (SP)+,SOMODE+1    ;: NUMBER OF DIGITS TO TYPE
(1)      020710 062716 000002      ADD      #2,(SP)          ;: ADJUST RETURN ADDRESS
(1)      020714 000406      BR      STYPON
(1)      020716 112767 000001 000171  STYPOC: MOVB     #1,SOFILL      ;: SET THE ZERO FILL SWITCH
(1)      020724 112767 000006 000165  MOVB     #6,SOMODE+1      ;: SET FOR SIX(6) DIGITS
(1)      020732 112767 000005 000154  STYPON: MOVB     #5,SOCNT      ;: SET THE ITERATION COUNT
(1)      020740 010346      MOV      R3,-(SP)         ;: SAVE R3
(1)      020742 010446      MOV      R4,-(SP)         ;: SAVE R4
(1)      020744 010546      MOV      R5,-(SP)         ;: SAVE R5
(1)      020746 116704 000145      MOVB     SOMODE+1,R4      ;: GET THE NUMBER OF DIGITS TO TYPE
(1)      020752 005404      NEG      R4
(1)      020754 062704 000006      ADD      #6,R4            ;: SUBTRACT IT FOR MAX. ALLOWED
(1)      020760 110467 000132      MOVB     R4,SOMODE        ;: SAVE IT FOR USE
(1)      020764 116704 000125      MOVB     SOFILL,R4        ;: GET THE ZERO FILL SWITCH
(1)      020770 016605 000012      MOV      12(SP),R5        ;: PICKUP THE INPUT NUMBER
(1)      020774 005003      CLR      R3               ;: CLEAR THE OUTPUT WORD
(1)      020776 006105      1S:     ROL      R5        ;: ROTATE MSB INTO "C"
(1)      021000 000404      BR      3S                ;: GO DO MSB
(1)      021002 006105      2S:     ROL      R5        ;: FORM THIS DIGIT
(1)      021004 006105      ROL      R5
(1)      021006 006105      ROL      R5
(1)      021010 010503      MOV      R5,R3
(1)      021012 006103      3S:     ROL      R3        ;: GET LSB OF THIS DIGIT
(1)      021014 105367 000076      DECB     SOMODE           ;: TYPE THIS DIGIT?
(1)      021020 100016      BPL      7S               ;: BR IF NO
(1)      021022 042703 177770      BIC      #177770,R3       ;: GET RID OF JUNK
(1)      021026 001002      BNE      4S               ;: TEST FOR 0
(1)      021030 005704      TST     R4                ;: SUPPRESS THIS 0?
(1)      021032 001403      BEQ     5S                ;: BR IF YES
(1)      021034 005204      4S:     INC     R4          ;: DON'T SUPPRESS ANYMORE 0'S
(1)      021036 052703 000060      BIS     #'0,R3           ;: MAKE THIS DIGIT ASCII

```



```

(1) 021042 052703 000040 5S: BIS #',R3 ::MAKE ASCII IF NOT ALREADY
(1) 021046 110367 000040 MOV R3,B# ::SAVE FOR TYPING
(1) 021050 104400 021112 TYPE B# ::GO TYPE THIS DIGIT
(1) 021054 105367 000032 7S: DECB $OCNT ::COUNT BY 1
(1) 021058 003347 BGT 2# ::BR IF MORE TO DO
(1) 021062 002402 BLT 6# ::BR IF DONE
(1) 021066 005204 INC R4 ::INSURE LAST DIGIT ISN'T A BLANK
(1) 021070 000744 BR 2# ::GO DO THE LAST DIGIT
(1) 021072 012605 6S: MOV (SP)+,R5 ::RESTORE R5
(1) 021074 012604 MOV (SP)+,R4 ::RESTORE R4
(1) 021076 012603 MOV (SP)+,R3 ::RESTORE R3
(1) 021100 016666 000002 000004 MOV 2(SP),4(SP) ::SET THE STACK FOR RETURNING
(1) 021106 012616 MOV (SP)+,(SP)
(1) 021110 000002 RTI ::RETURN
(1) 021112 000 8S: .BYTE 0 ::STORAGE FOR ASCII DIGIT
(1) 021113 000 .BYTE 00 ::TERMINATOR FOR TYPE ROUTINE
(1) 021114 000 SOCNT: .BYTE 00 ::OCTAL DIGIT COUNTER
(1) 021115 000 SOFILL: .BYTE 00 ::ZERO FILL SWITCH
(1) 021116 000000 SOMODE: .WORD 0 ::NUMBER OF DIGITS TO TYPE
*****
4989 (1)
(1) .SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
(1)
(1) ::THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
(1) ::SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
(1) ::NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
(1) ::BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
(1) ::REPLACED WITH SPACES.
(1) ::CALL:
(1) * MOV NUM,-(SP) ::PUT THE BINARY NUMBER ON THE STACK
(1) * TYPDS ::GO TO THE ROUTINE
(1)
(2) 021120 010046 STYPDS: MOV R0,-(SP) ::PUSH R0 ON STACK
(3) 021122 010146 MOV R1,-(SP) ::PUSH R1 ON STACK
(3) 021124 010246 MOV R2,-(SP) ::PUSH R2 ON STACK
(3) 021126 010346 MOV R3,-(SP) ::PUSH R3 ON STACK
(3) 021128 010546 MOV R5,-(SP) ::PUSH R5 ON STACK
(1) 021130 012746 020200 MOV 020200,-(SP) ::SET BLANK SWITCH AND SIGN
(1) 021132 016605 000020 MOV 20(SP),R5 ::GET THE INPUT NUMBER
(1) 021134 100004 BPL 1# ::BR IF INPUT IS POS.
(1) 021136 005405 NEG R5 ::MAKE THE BINARY NUMBER POS.
(1) 021138 112766 000055 000001 MOV B',-1(SP) ::MAKE THE ASCII NUMBER NEG.
(1) 021140 005000 1S: CLR R0 ::ZERO THE CONSTANTS INDEX
(1) 021142 012703 021334 MOV #SDBLK,R3 ::SETUP THE OUTPUT POINTER
(1) 021144 112723 000040 MOV B',(R3)+ ::SET THE FIRST CHARACTER TO A BLANK
(1) 021146 005002 2S: CLR R2 ::CLEAR THE BCD NUMBER
(1) 021148 016001 021324 MOV SOTBL(R0),R1 ::GET THE CONSTANT
(1) 021150 160105 3S: SUB R1,R5 ::FORM THIS BCD DIGIT
(1) 021152 002402 BLT 4# ::BR IF DONE
(1) 021154 005202 INC R2 ::INCREASE THE BCD DIGIT BY 1
(1) 021156 000774 BR 3#
(1) 021158 060105 4S: ADD R1,R5 ::ADD BACK THE CONSTANT
(1) 021200 005702 TST R2 ::CHECK IF BCD DIGIT=0

```

```

(1) 021210 001002 BNE 55 ; FALL THROUGH IF 0
(1) 021212 105716 TSTB (SP) ; STILL DOING LEADING 0'S?
(1) 021214 100407 BMI 75 ; BR IF YES
(1) 021216 106316 55: ASLB (SP) ; MSD?
(1) 021220 103003 BCC 65 ; BR IF NO
(1) 021222 116663 000001 177777 MOVB 1(SP),-1(R3) ; YES--SET THE SIGN
(1) 021224 052702 000060 65: BIS 0'R2 ; MAKE THE BCD DIGIT ASCII
(1) 021226 052702 000040 75: BIS 0'R2 ; MAKE IT A SPACE IF NOT ALREADY A DIGIT
(1) 021228 110223 MOVB R2,(R3)+ ; PUT THIS CHARACTER IN THE OUTPUT BUFFER
(1) 021230 005720 TST (R0)+ ; JUST INCREMENTING
(1) 021232 020027 000010 CMP R0,#10 ; CHECK THE TABLE INDEX
(1) 021234 002746 BLT 25 ; GO DO THE NEXT DIGIT
(1) 021236 003002 BGT 85 ; GO TO EXIT
(1) 021238 010502 MOV R5,R2 ; GET THE LSD
(1) 021240 000764 BR 65 ; GO CHANGE TO ASCII
(1) 021242 105726 85: TSTB (SP)+ ; WAS THE LSD THE FIRST NON-ZERO?
(1) 021244 100003 BPL 95 ; BR IF NO
(1) 021246 116663 177777 177776 95: MOVB -1(SP),-2(R3) ; YES--SET THE SIGN FOR TYPING
(1) 021248 105013 CLR R3 ; SET THE TERMINATOR
(3) 021274 012605 MOV (SP)+,R5 ; POP STACK INTO R5
(3) 021276 012603 MOV (SP)+,R3 ; POP STACK INTO R3
(3) 021300 012602 MOV (SP)+,R2 ; POP STACK INTO R2
(3) 021302 012601 MOV (SP)+,R1 ; POP STACK INTO R1
(3) 021304 012600 MOV (SP)+,R0 ; POP STACK INTO R0
(1) 021306 104400 021334 TYPE SDBLK ; NOW TYPE THE NUMBER
(1) 021312 016666 000002 000004 MOV 2(SP),4(SP) ; ADJUST THE STACK
(1) 021320 012616 MOV (SP)+,(SP)
(1) 021322 000002 RTI ; RETURN TO USER
(1) 021324 023420 SOTBL: 10000.
(1) 021326 001750 1000.
(1) 021330 000144 100.
(1) 021332 000012 10.
(1) 021334 000004 SDBLK: .BLKN 4
;*****
(1) .SBTTL TYPE ROUTINE
(1) ;#ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
(1) ;#THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
(1) ;#NOTE1: SNULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
(1) ;#NOTE2: SFILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
(1) ;#NOTE3: SFILLC CONTAINS THE CHARACTER TO FILL AFTER.
(1) ;#CALL:
(1) ;#1) USING A TRAP INSTRUCTION
(1) ;# TYPE ,MESADR ;:MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
(1) ;#OR
(1) ;# TYPE
(1) ;# MESADR
(1) ;#
(1) ;#
(1) 021344 105767 157605 STYPE: TSTB STPFLG ;: IS THERE A TERMINAL?
(1) 021350 100002 BPL IS ;: BR IF YES
(1) 021352 000000 HALT ;: HALT HERE IF NO TERMINAL

```

```

(1) 021354 000430 BR 3S : LEAVE
(1) 021356 010046 (1) 021356 010046 1S: MOV RO,-(SP) : SAVE RO
(1) 021360 017600 000002 MOV 22(SP),RO : GET ADDRESS OF ASCIZ STRING
(1) 021364 122767 000001 157656 CMPB #APTENV,SENV : RUNNING IN APT MODE
(1) 021372 001011 BNE 62S : NO GO CHECK FOR APT CONSOLE
(1) 021374 132767 000100 157647 BITB #APTSPool,SENV : SPOOL MESSAGE TO APT
(1) 021402 001405 BEQ 62S : NO GO CHECK FOR CONSOLE
(1) 021404 010067 000004 MOV RO,61S : SETUP MESSAGE ADDRESS FOR APT
(1) 021410 004767 000214 JSR PC,SATY3 : SPOOL MESSAGE TO APT
(1) 021414 000000 (1) 021414 000000 61S: .WORD 0 : MESSAGE ADDRESS
(1) 021416 132767 000040 157625 62S: BITB #APTCSUP,SENV : APT CONSOLE SUPPRESSED
(1) 021424 001003 BNE 60S : YES, SKIP TYPE OUT
(1) 021426 112046 (1) 021426 112046 2S: MOVB (RO)+,-(SP) : PUSH CHARACTER TO BE TYPED ONTO STACK
(1) 021430 001005 BNE 4S : BR IF IT ISN'T THE TERMINATOR
(1) 021432 005726 TST (SP)+ : IF TERMINATOR POP IT OFF THE STACK
(1) 021434 012600 (1) 021434 012600 60S: MOV (SP)+,RO : RESTORE RO
(1) 021436 062716 000002 3S: ADD #2,(SP) : ADJUST RETURN PC
(1) 021442 000002 RTI : RETURN
(1) 021444 122716 000011 4S: CMPB #HT,(SP) : BRANCH IF <HT>
(1) 021450 001426 BEQ 8S :
(1) 021452 122716 000200 CMPB #CRLF,(SP) : ; BRANCH IF NOT <CRLF>
(1) 021456 001004 BNE 5S :
(1) 021460 005726 TST (SP)+ : ; POP <CR><LF> EQUIV
(1) 021462 104400 TYPE : ; TYPE A CR AND LF
(1) 021464 001225 SCRLF :
(1) 021466 000757 BR 2S : ; GET NEXT CHARACTER
(1) 021470 004767 000056 5S: JSR PC,STYPEC : GO TYPE THIS CHARACTER
(1) 021474 126726 157454 6S: CMPB #FILLC,(SP)+ : IS IT TIME FOR FILLER CHARS.?
(1) 021500 001352 BNE 2S : IF NO GO GET NEXT CHAR.
(1) 021502 016746 157444 MOV #NULL,-(SP) : GET # OF FILLER CHARS. NEEDED AND THE NULL CHAR.
(1) 021506 105366 000001 7S: DECB 1(SP) : DOES A NULL NEED TO BE TYPED?
(1) 021512 002770 BLT 6S : BR IF NO--GO POP THE NULL OFF OF STACK
(1) 021514 004767 000032 JSR PC,STYPEC : GO TYPE A NULL
(1) 021520 105367 000072 DECB #CHARCNT : DO NOT COUNT AS A COUNT
(1) 021524 000770 BR 7S : LOOP
(1) ;HORIZONTAL TAB PROCESSOR
(1) 021526 112716 000040 8S: MOVB #40,(SP) : REPLACE TAB WITH SPACE
(1) 021532 004767 000014 9S: JSR PC,STYPEC : TYPE A SPACE
(1) 021536 132767 000007 000052 BITB #7,#CHARCNT : BRANCH IF NOT AT
(1) 021544 001372 BNE 9S : TAB STOP
(1) 021546 005726 TST (SP)+ : POP SPACE OFF STACK
(1) 021550 000726 BR 2S : GET NEXT CHARACTER
(1) 021552 105777 157370 STYPEC: TSTB #STPS : WAIT UNTIL PRINTER IS READY
(1) 021556 100375 BPL STYPEC :
(1) 021560 116677 000002 157362 MOVB 2(SP),#STPB : ; LOAD CHAR TO BE TYPED INTO DATA REG.
(1) 021566 122766 000015 000002 CMPB #15,2(SP) : ; BRANCH IF
(1) 021574 001003 BNE 1S : ; NOT <CR>
(1) 021576 105067 000014 CLRB #CHARCNT :
(1) 021602 000406 BR STYPEX : ; EXIT
(1) 021604 122766 000012 000002 1S: CMPB #12,2(SP) : ; BRANCH IF
(1) 021612 002002 BGE STYPEX : ; <LF>

```

```

(1) 021614 105227          INCB      (PC)+      ;;INC SPACE
(1) 021616 000000          SCHARCNT: WORD 0      ;;COUNT
(1) 021620 000207          STYPEX: RTS      PC
(1)          ;;          EQUATES
(1)          000011          THT=11
(1)          000200          TCRLF=200
4991          ;*****
(1)          .SBTTL      APT COMMUNICATIONS ROUTINE
(1) 021622 112767 000001 000236 SATY1:  MOVB      #1,SFFLG      ;TO REPORT FATAL ERROR
(1) 021630 112767 000001 000226 SATY3:  MOVB      #1,SMFLG      ;TO TYPE A MESSAGE
(1) 021636 000403          BR          SATYC
(1) 021640 112767 000001 000220 SATY4:  MOVB      #1,SFFLG      ;TO ONLY REPORT FATAL ERROR
(2) 021646          SATYC:
(3) 021646 010046          MOV      RO,-(SP)      ;;PUSH RO ON STACK
(3) 021650 010146          MOV      R1,-(SP)      ;;PUSH R1 ON STACK
(1) 021652 105767 000206          TSTB     SMFLG      ;;SHOULD TYPE A MESSAGE?
(1) 021656 001450          BEQ      55          ;;IF NOT: BR
(1) 021660 122767 000001 157362          CMPSB   #APTENV,SENV   ;;OPERATING UNDER APT?
(1) 021666 001031          BNE     35          ;;IF NOT: BR
(1) 021670 132767 000100 157353          BITB    #APTSPool,SENVH ;;SHOULD SPOOL MESSAGES?
(1) 021676 001425          BEQ      35          ;;IF NOT: BR
(1) 021700 017600 000004          MOV      #4(SP),RO     ;GET MESSAGE ADDR.
(1) 021704 062766 000002 000004          ADD     #2,4(SP)      ;BUMP RETURN ADDR.
(1) 021712 005767 157312          15:    TST      SMSGTYPE   ;;SEE IF DONE W/ LAST XMISSION?
(1) 021716 001375          BNE     15          ;;IF NOT: WAIT
(1) 021720 010067 157320          MOV     RO,SMSGAD     ;PUT ADDR IN MAILBOX
(1) 021724 105720          25:    TSTB     (RO)+    ;FIND END OF MESSAGE
(1) 021726 001376          BNE     25
(1) 021730 166700 157310          SUB     SMSGAD,RO     ;SUB START OF MESSAGE
(1) 021734 006200          ASR     RO           ;GET MESSAGE LNGTH IN WORDS
(1) 021736 010067 157304          MOV     RO,SMSG LGT   ;PUT LENGTH IN MAILBOX
(1) 021742 012767 000004 157260          MOV     #4,SMSGTYPE   ;TELL APT TO TAKE MSG.
(1) 021750 000413          BR      55
(1) 021752 017667 000004 000016 35:    MOV     #4(SP),45     ;PUT MSG ADDR IN JSR LINKAGE
(1) 021760 062766 000002 000004          ADD     #2,4(SP)      ;BUMP RETURN ADDRESS
(3) 021766 016746 156004          MOV     177776,-(SP)  ;;PUSH 177776 ON STACK
(1) 021772 004767 177346          JSR     PC,STYPE     ;CALL TYPE MACRO
(1) 021776 000000          45:    .WORD    0
(1) 022000          55:
(1) 022000 105767 000062          105:   TSTB     SFFLG      ;SHOULD REPORT FATAL ERROR?
(1) 022004 001416          BEQ     125          ;;IF NOT: BR
(1) 022006 005767 157236          TST     SENV         ;RUNNING UNDER APT?
(1) 022012 001413          BEQ     125          ;;IF NOT: BR
(1) 022014 005767 157210          115:   TST     SMSGTYPE   ;FINISHED LAST MESSAGE?
(1) 022020 001375          BNE     115          ;;IF NOT: WAIT
(1) 022022 017667 000004 157202          MOV     #4(SP),SFATAL ;GET ERROR #
(1) 022030 062766 000002 000004          ADD     #2,4(SP)      ;BUMP RETURN ADDR.
(1) 022036 005267 157166          INC     SMSGTYPE     ;TELL APT TO TAKE ERROR
(1) 022042 105067 000020          125:   CLRB    SFFLG      ;CLEAR FATAL FLAG
(1) 022046 105067 000013          CLRB    SMFLG      ;CLEAR LOG FLAG
(1) 022052 105067 000006          CLRB    SMFLG      ;CLEAR MESSAGE FLAG
(3) 022056 012601          MOV     (SP)+,R1     ;;POP STACK INTO R1

```

```

(3) 022060 012600          MOV      (SP)+,R0          ;; POP STACK INTO R0
(1) 022062 0002C7          RTS      PC                ;; RETURN
(1)
(1) 022064      000          SMFLG: .BYTE 0           ;; MESSG. FLAG
(1) 022065      000          SLFLG: .BYTE 0           ;; LOG FLAG
(1) 022066      000          SFFLG: .BYTE 0           ;; FATAL FLAG
(1)      022070          .EVEN
(1)      000200          APTSIZE=200
(1)      000001          APTENV=001
(1)      000100          APTSPool=100
(1)      000040          APTCSUP=040
4992 ;*****
(1)
(1) .SBTTL TTY INPUT ROUTINE
(1)
(1) ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
(1) ;*CALL:
(1) ;*   ROCHR                ;; INPUT A SINGLE CHARACTER FROM THE TTY
(1) ;*   RETURN HERE          ;; CHARACTER IS ON THE STACK
(1)
(1) 022070 011646          SRDCHR: MOV      (SP),-(SP)      ;; PUSH DOWN THE PC
(1) 022072 016666 000004 000002  MOV      4(SP),2(SP)          ;; SAVE THE PS
(1) 022100 105777 157036          1S:   TSTB   @STKS             ;; WAIT FOR
(1) 022104 100375          BPL     1S                   ;; A CHARACTER
(1) 022106 117766 157032 000004  MOVB   @STKB,4(SP)          ;; READ THE TTY
(1) 022114 042766 177600 000004  BIC    @C<177>,4(SP)        ;; GET RID OF JUNK IF ANY
(1) 022122 000002          RTI                        ;; GO BACK TO USER
(2) ;*****
(1) ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
(1) ;*CALL:
(1) ;*   RDLIN                ;; INPUT A STRING FROM THE TTY
(1) ;*   RETURN HERE          ;; ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
(1) ;*   TERMINATOR WILL BE A BYTE OF ALL 0'S
(1)
(1) 022124 010346          SRDLIN: MOV      R3, -(SP)      ;; SAVE R3
(1) 022126 012703 022232 1S:   MOV      @STTYIN,R3          ;; GET ADDRESS
(1) 022132 022703 022242 2S:   CMP      @STTYIN+8.,R3      ;; BUFFER FULL?
(1) 022136 101405          BLOS   4S                   ;; BR IF YES
(1) 022140 104405          ROCHR   GO READ ONE CHARACTER FROM THE TTY
(1) 022142 112613          MOVB   (SP)+,(R3)          ;; GET CHARACTER
(1) 022144 122713 000177          CMPB   @177,(R3)          ;; IS IT A RUBOUT
(1) 022150 001003          BNE    3S                   ;; SKIP IF NOT
(1) 022152 104400 001224 4S:   TYPE   @QUES              ;; TYPE A '?'
(1) 022156 000763          BR     1S                   ;; CLEAR THE BUFFER AND LOOP
(1) 022160 111367 000044 3S:   MOVB   (R3),9S            ;; ECHO THE CHARACTER
(1) 022164 104400 022230          TYPE   9S
(1) 022170 122723 000015          CMPB   @15,(R3)+          ;; CHECK FOR RETURN
(1) 022174 001356          BNE    2S                   ;; LOOP IF NOT RETURN
(1) 022176 105063 177777          CLRB   -1(R3)              ;; CLEAR RETURN (THE 15)
(1) 022202 104400 001226          TYPE   $LF                 ;; TYPE A LINE FEED
(1) 022206 012603          MOV     (SP)+,R3           ;; RESTORE R3
(1) 022210 011646          MOV     (SP),-(SP)         ;; ADJUST THE STACK AND PUT ADDRESS OF THE
(1) 022212 016666 000004 000002  MOV     4(SP),2(SP)        ;; FIRST ASCII CHARACTER ON IT

```

```

(1) 022220 012766 022232 000004      MOV      #STTYIN,4(SP)
(1) 022226 000002                    RTI      ;: RETURN
(1) 022230      000                    9S: .BYTE 0 ;: STORAGE FOR ASCII CHAR. TO TYPE
(1) 022231      000                    .BYTE 0 ;: TERMINATOR
(1) 022232 000010                    STTYIN: .BLKB 8. ;: RESERVE 8 BYTES FOR TTY INPUT
4993 ;:*****
(1)                                     .SBTTL READ AN OCTAL NUMBER FROM THE TTY
(1)                                     ;: THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
(1)                                     ;: CHANGE IT TO BINARY.
(1)                                     ;: THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
(1)                                     ;: OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
(1)                                     ;: FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
(1)                                     ;: THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
(1)                                     ;: CALL:
(1)                                     ;: * RDOCT ;: READ AN OCTAL NUMBER
(1)                                     ;: * RETURN HERE ;: LOW ORDER BITS ARE ON TOP OF THE STACK
(1)                                     ;: * ;: HIGH ORDER BITS ARE IN SHIOCT
(1) 022242 011646                    SRDOCT: MOV      (SP) -(SP) ;: PROVIDE SPACE FOR THE
(1) 022244 016666 000004 000002      MOV      4(SP),2(SP) ;: INPUT NUMBER
(3) 022250 010046                    MOV      R0,-(SP) ;: PUSH R0 ON STACK
(3) 022254 010146                    MOV      R1,-(SP) ;: PUSH R1 ON STACK
(3) 022256 010246                    MOV      R2,-(SP) ;: PUSH R2 ON STACK
(1) 022260 104406                    1S: RDLIN ;: READ AN ASCII LINE
(1) 022262 012600                    MOV      (SP)+,R0 ;: GET ADDRESS OF 1ST CHARACTER
(1) 022264 010067 000100            MOV      R0,5S ;: AND SAVE IT
(1) 022270 005001                    CLR      R1 ;: CLEAR DATA WORD
(1) 022272 005002                    CLR      R2
(1) 022274 112046                    2S: MOVVB (R0)+,-(SP) ;: PICKUP THIS CHARACTER
(1) 022276 001420                    BEQ      3S ;: IF ZERO GET OUT
(1) 022300 122716 000060            CMPB    #'0,(SP) ;: MAKE SURE THIS CHARACTER
(1) 022304 003026                    BGT      4S ;: IS AN OCTAL DIGIT
(1) 022306 122716 000067            CMPB    #'7,(SP)
(1) 022312 002423                    BLT      4S
(1) 022314 006301                    ASL      R1 ;: #2
(1) 022316 006102                    ROL      R2
(1) 022320 006301                    ASL      R1 ;: #4
(1) 022322 006102                    ROL      R2
(1) 022324 006301                    ASL      R1 ;: #8
(1) 022326 006102                    ROL      R2
(1) 022330 042716 177770            BIC      #'C7,(SP) ;: STRIP THE ASCII JUNK
(1) 022334 062601                    ADD      (SP)+,R1 ;: ADD IN THIS DIGIT
(1) 022336 000756                    BR       2S ;: LOOP
(1) 022340 005726                    3S: TST      (SP)+ ;: CLEAN TERMINATOR FROM STACK
(1) 022342 010166 000012            MOV      R1,12(SP) ;: SAVE THE RESULT
(1) 022344 010267 000026            MOV      R2,$HIOCT
(3) 022352 012602                    MOV      (SP)+,R2 ;: POP STACK INTO R2
(3) 022354 012601                    MOV      (SP)+,R1 ;: POP STACK INTO R1
(3) 022356 012600                    MOV      (SP)+,R0 ;: POP STACK INTO R0
(1) 022360 000002                    RTI      ;: RETURN
(1) 022362 005726                    4S: TST      (SP)+ ;: CLEAN PARTIAL FROM STACK
(1) 022364 105010                    CLRB    (R0) ;: SET A TERMINATOR

```

(1) 022366 104400
(1) 022370 000000
(1) 022372 104400 001224
(1) 022376 000730
(1) 022400 000000
4994

```
SS:      TYPE          ;;TYPE UP THRU THE BAD CHAR.
        WORD          0
        TYPE          0 SQUES
        BR            1S
SHIOCT: .WORD          0  ;; "?" "CR" & "LF"
                          ;; TRY AGAIN
                          ;; HIGH ORDER BITS GO HERE
;*****
```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 022402 010046
(1) 022404 016600 000002
(1) 022410 005740
(1) 022412 111000
(1) 022414 006300
(1) 022416 016000 022424
(1) 022422 000200
(1)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3)
(3) 022424 021344
(3) 022426 020716
(3) 022430 020672
(3) 022432 020732
(3) 022434 021120
(3) 022436 022070
(3) 022440 022124
(3) 022442 022242
4995

.SBTTL TRAP DECODER

```
;;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
;;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
;;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
;;*GO TO THAT ROUTINE.
```

```
STRAP:  MOV      RO,-(SP)      ;;SAVE RO
        MOV      2(SP),RO     ;;GET TRAP ADDRESS
        TST      -(RO)        ;;BACKUP BY 2
        MOVB     (RO),RO      ;;GET RIGHT BYTE OF TRAP
        ASL      RO           ;;POSITION FOR INDEXING
        MOV      STRPAD(RO),RO ;;INDEX TO TABLE
        RTS      RO           ;;GO TO ROUTINE
```

.SBTTL TRAP TABLE

```
;;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
;;*BY THE "TRAP" INSTRUCTION.
```

```
ROUTINE
-----
STRPAL: $TYPE    ;;CALL=TYPE    TRAP+0(104400)  TTY TYPEOUT ROUTINE
        $TYPOC   ;;CALL=TYPOC   TRAP+1(104401)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
        $TYPOS   ;;CALL=TYPOS   TRAP+2(104402)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
        $TYPON   ;;CALL=TYPON   TRAP+3(104403)  TYPE OCTAL NUMBER (AS PER LAST CALL)
        $TYPDS   ;;CALL=TYPOD   TRAP+4(104404)  TYPE DECIMAL NUMBER (WITH SIGN)
        $RDCHR   ;;CALL=RDCHR   TRAP+5(104405)  TTY TYPEIN CHARACTER ROUTINE
        $RDLIN   ;;CALL=RDLIN   TRAP+6(104406)  TTY TYPEIN STRING ROUTINE
        $RDOCT   ;;CALL=RDOCT   TRAP+7(104407)  READ AN OCTAL NUMBER FROM TTY
;*****
```

.SBTTL POWER DOWN AND UP ROUTINES

:POWER DOWN ROUTINE

(1) 022444 012737 022572 000024
(1) 022452 012737 000340 000026
(3) 022460 010046
(3) 022462 010146
(3) 022464 010246
(3) 022466 010346
(3) 022470 010446
(3) 022472 010546
(1) 022474 010667 000076
(1) 022500 012737 022512 000024

```
$PWRDN: MOV      $SILLUP,$PWRVEC ;;SET FOR FAST UP
        MOV      $340,$PWRVEC+2 ;;PRIO:7
        MOV      RO,-(SP)      ;;PUSH RO ON STACK
        MOV      R1,-(SP)      ;;PUSH R1 ON STACK
        MOV      R2,-(SP)      ;;PUSH R2 ON STACK
        MOV      R3,-(SP)      ;;PUSH R3 ON STACK
        MOV      R4,-(SP)      ;;PUSH R4 ON STACK
        MOV      R5,-(SP)      ;;PUSH R5 ON STACK
        MOV      SP,$SAVR6     ;;SAVE SP
        MOV      $PWRUP,$PWRVEC ;;SET UP VECTOR
```

```

(1) 022506 000000          HALT
(1) 022510 000776          BR      .-2          ;;HANG UP
(1)
(1)
(1) 022512 016706 000060      :POWER UP ROUTINE
(1) 022516 005067 000054      SPWRUP: MOV     $$AVR6, SP      ;; GET SP
(1) 022520 005267 000050      15:   CLR     $$AVR6          ;; WAIT LOOP FOR THE TTY
(1) 022524 001375          INC     $$AVR6          ;; WAIT FOR THE INC
(1) 022528 012600          BNE    15              OF WORD
(1) 022532 012600          MOV     (SP)+, R5      POP STACK INTO R5
(1) 022536 012600          MOV     (SP)+, R4      POP STACK INTO R4
(1) 022540 012600          MOV     (SP)+, R3      POP STACK INTO R3
(1) 022544 012600          MOV     (SP)+, R2      POP STACK INTO R2
(1) 022548 012600          MOV     (SP)+, R1      POP STACK INTO R1
(1) 022552 012600          MOV     (SP)+, R0      POP STACK INTO R0
(1) 022556 012737 022444 000024  MOV     $SPWRON, $PMRVEC ;; SET UP THE POWER DOWN VECTOR
(1) 022560 012737 000340 000026  MOV     $340, $PMRVEC+2 ;; PRIO: 7
(1) 022564 104400          TYPE                                REPORT THE POWER FAILURE
(1) 022568 022600          SPWRMG: .WORD    SPOWER        POWER FAIL MESSAGE POINTER
(1) 022572 012716          MOV     (PC)+, (SP)      RESTART AT RSTRTA
(1) 022576 002612          SPWRAD: .WORD    RSTRTA        RESTART ADDRESS
(1) 022580 000002          RTI
(1) 022584 000000          SILLUP: HALT
(1) 022588 000776          BR      .-2          ;; THE POWER UP SEQUENCE WAS STARTED
(1) 022592 000000          $$AVR6: 0              BEFORE THE POWER DOWN WAS COMPLETE
(1) 022600 005015 047520 042527  SPPOWER: .ASCIZ  <15><12>"POWER" ;; PUT THE SP HERE
(1) 022606 000122          .EVEN

```



```

4998      ;*****
4999      ;COMMON DH11 SERVICE ROUTINES
5000      ;*****
5001
5002      ;THIS ROUTINE IS CALLED DURING START UP TO LOAD THE XMITTER
5003      ;OUTPUT BUFFER WITH A BINARY COUNT TEST PATTERN
5004
5005      LDTBF1: MOV      #TBUF,R1      ;POINT TO START OF BUFFER
5006      CLR        R2                ;INIT DATA BYTE GENERATOR
5007      1S:      MOVB     R2,(R1)+    ;LOAD ONE CHAR
5008      INC        R2                ;GENERATE NEXT CHAR
5009      CMP        #400,R2          ;LOADED 256(10) BYTES
5010      BNE        1S              ;BR IF NOT
5011      RTS         PC              ;RETURN TO START TESTING
5012
5013      ;THIS ROUTINE SETS UP THE ERROR INFORMATION REQUIRED BY ANY TEST
5014      ;USING A "DH1" HEADER
5015
5016      SUER1: JSR      PC,SAPS        ;SAVE THE ERROR PSW
5017      MOVB     $STNM,R0           ;SAVE THE TEST NO.
5018      MOV      R0,$REG0          ;SAVE THE TEST NO. FOR ERROR PRINT
5019      MOV      R1,$REG1          ;SAVE THE DH11 ADDR
5020      MOV      R2,$REG2          ;SAVE THE REG ADDRESS
5021      MOV      R6,$REG6          ;SAVE THE SP
5022      ADD      #2,$REG6          ;CORRECT FOR CALLING JSR
5023      RTS         PC              ;RETURN TO CALLING ROUTINE
5024
5025      ;THIS ROUTINE IS CALLED BY THOSE TESTS USING A "DH2" HEADER TO
5026      ;SAVE THE ERROR INFORMATION IN "DT2"
5027
5028      SUER2: JSR      PC,SAPS        ;SAVE THE ERROR PSW
5029      SUER2A: MOVB     $STNM,R0           ;GET THE TEST NO.
5030      MOV      R0,$REG0          ;SAVE THE REGISTERS-TEST#
5031      MOV      R1,$REG1          ;SAVE THE DH ADDRESS
5032      MOV      R2,$REG2          ;SAVE THE REGISTER ADDRESS
5033      MOV      R3,$REG3          ;SAVE THE WAS DATA
5034      MOV      R4,$REG4          ;SAVE THE S/B DATA
5035      MOV      R6,$REG6          ;SAVE THE STACK POINTER
5036      ADD      #2,$REG6          ;CORRECT FOR CALLING JSR
5037      RTS         PC              ;RETURN TO REPORT ERROR
5038
5039      ;THIS ROUTINE IS CALLED TO SET UP ERROR INFORMATION FOR THE
5040      ;BUS ERROR AND RSVD INSTR ERROR ROUTINES
5041
5042      SUER3: MOV      R0,$REG0          ;SAVE THE REGS
5043      MOV      R1,$REG1
5044      MOV      R2,$REG2
5045      RTS         PC              ;RETURN TO REPORT ERROR
5046
5047      ;THIS ROUTINE IS CALLED TO SET UP ERROR INFORMATION FOR THE
5048      ;CAR/BCR MEMORY PATTERNS TESTS
5049
5050      SUER4: CLR        $TMPD          ;SAVE THE LINE NO. WRITTEN
5051      MOVB     LINEA,$TMPD
    
```

```

5052 022772 116700 156104      MOV      $STNM,RO      ;SAVE THE TEST NUMBER
5053 022776 010067 156156      MOV      RO,$REGO     ;SAVE THE REGISTER INFORMATION
5054 023002 010167 156154      MOV      R1,$REG1
5055 023006 010267 156152      MOV      R2,$REG2
5056 023012 010367 156150      MOV      R3,$REG3
5057 023016 010467 156146      MOV      R4,$REG4
5058 023022 000207                RTS      PC            ;RETURN TO PATTERNS TEST
    
```

; THIS ROUTINE IS CALLED TO SELECT A NEW LINE NO. BASED ON THE
 ; VALUE OF THE LINE SELECTION PARAMETER

; CALLING SEQUENCE:

```

;JSR    PC,SELINE      ;CALL THE ROUTINE
;BR     1$             ;EXIT BRANCH-ROUTINE MOVES THE RETURN
                    ;PC AROUND THIS BR IF MORE LINES ARE
                    ;YET TO BE TESTED
    
```

```

5070 023024 105767 002115      SELINE: TSTB    LINE+1      ;FIRST TIME THROUGH FOR ANY TEST ?
5071 023030 001010                BNE     1$                ;BR IF NOT
5072 023032 105167 002107      COMB    LINE+1          ;SET ENTRY FLAG
5073 023036 012767 000001 001346  MOV     R1,LINMSK      ;INIT SELECT TEST MASK TO TEST LINE 00
5074 023044 105067 002074      CLRB   LINE            ;START WITH LINE #00
5075 023050 000405                BR     2$                ;GO TEST FOR LINE #00
5076 023052 105267 002066      1$:    INCB   LINE      ;GENERATE NEW LINE NO.
5077 023056 006367 001330      ASL    LINMSK         ;SHIFT SELECT MASK TO TEST NXT LINE
5078 023062 001407                BEQ    3$                ;RETURN TO EXIT BRANCH - ALL LINES DONE
5079 023064 036767 001322 001316  2$:    BIT    LINMSK,LINSEL ;IS THE LINE SELECTED FOR TEST ??
5080 023072 001767                BEQ    1$                ;BR IF NOT
5081 023074 062716 000002      ADD    R2,(SP)        ;MOVE RETURN PC AROUND EXIT BRANCH
5082 023100 000402                BR     4$                ;RETURN TO TEST SELECTED LINE
5083 023102 005067 002036      3$:    CLR    LINE      ;INIT ENTRY FLAG AND LINE NO. TO 000
5084 023106 142777 000017 001264  4$:    BICB   R17,$DHADR   ;INIT LINE SELECT BITS IN "SCR"
5085 023114 000207                RTS     PC              ;RETURN TO CALLING TEST
    
```

; THIS ROUTINE IS CALLED TO CONVERT EITHER THE "DH" NUMBER OR THE
 ; "LINE" NUMBER TO TWO ASCII CHARACTERS AND MOVE THEM INTO A
 ; PARTICULAR MESSAGE BUFFER FOR ERROR REPORTING

; CALLING SEQUENCE

```

;JSR    R5,SUNUM      ;CALL TO THIS ROUTINE
;ADDR1  ;ADDRESS OF THE NUMBER TO BE CONVERTED
;ADDR2  ;ADDRESS OF THE MSG BUFFER SLOT
    
```

```

5097 023116                SUNUM:
(2) 023116 010046      MOV     RO,-(SP)        ;PUSH RO ON STACK
(2) 023120 010146      MOV     R1,-(SP)        ;PUSH R1 ON STACK
(2) 023122 010246      MOV     R2,-(SP)        ;PUSH R2 ON STACK
5098 023124 012500      MOV     (R5)+,RO        ;GET ADDRESS OF NUMBER
5099 023126 012501      MOV     (R5)+,R1        ;GET MSG BUFFER ADDR
5100 023130 111000      MOV     (RO),RO         ;GET NO. TO BE CONVERTED
5101 023132 010002      MOV     R0,R2          ;SAVE IT IN R2
5102 023134 006202      ASR    R2              ;SHIFT MSD TO LSD POSITION
    
```

```

5103 023136 006202 ASR R2
5104 023140 006202 ASR R2
5105 023142 042702 177770 BIC #177770,R2 ;CLR JUNK BITS
5106 023146 062702 000060 ADD #60,R2 ;MAKE IT ASCII
5107 023152 110221 MOV# R2,(R1)+ ;PUT IT IN MSG BUFFER
5108 023154 042700 177770 BIC #177770,R0 ;CLR JUNK FROM LSD
5109 023160 062700 000060 ADD #60,R0 ;MAKE IT ASCII
5110 023164 110011 MOV# R0,(R1) ;PUT LSD IN THE BUFFER
5111 023166 012602 MOV (SP)+,R2 ;POP STACK INTO R2
(2) 023170 012601 MOV (SP)+,R1 ;POP STACK INTO R1
(2) 023172 012600 MOV (SP)+,R0 ;POP STACK INTO R0
5112 023174 000205 RTS R5 ;RETURN TO CALLER

;THIS ROUTINE IS CALLED TO CLEAR THE "CAR" AND "BCR" MEMORIES
;IT ASSUMES THAT THE ADDRESS OF THE "SCR" IS IN R1
5114
5115
5116
5117 023176 005067 156014 CLCABC: CLR STMP7 ;INIT A COUNTER
5118 023202 116711 156010 IS: MOV# STMP7,(R1) ;SELECT A LINE
5119 023206 005061 000006 CLR CAR(R1) ;CLEAR A CAR LOCATION
5120 023212 005061 000010 CLR BCR(R1) ;CLEAR A BCR LOCATION
5121 023216 005267 155774 INC STMP7 ;GENERATE NEW LINE NO.
5122 023222 022767 000020 155766 CMP #20,STMP7 ;DONE ALL LINES?
5123 023230 001364 BNE IS ;BR IF NOT
5124 023232 142711 000017 BICB #17,(R1) ;SET "SCR" TO SELECT LINE 00
5125 023236 000207 RTS PC ;RETURN TO CALLER

;THIS ROUTINE IS CALLED TO LOAD THE "BCR" MEMORY WITH ALL ONES
;IT ASSUMES THAT THE ADDRESS OF THE SCR IS IN R1
5127
5128
5129
5130 023240 005067 155752 LDABC: CLR STMP7 ;INIT A COUNTER
5131 023244 116711 155746 IS: MOV# STMP7,(R1) ;SELECT A LINE
5132 023250 012761 177777 000010 MOV #-1,BCR(R1) ;LOAD BCR LOC. WITH 177777
5133 023256 005267 155734 INC STMP7 ;GENERATE NEXT LINE NO.
5134 023262 022767 000020 155726 CMP #20,STMP7 ;DONE ALL LINES?
5135 023270 001365 BNE IS ;BR IF NOT
5136 023272 142711 000017 BICB #17,(R1) ;SET "SCR" TO SELECT LINE 00
5137 023276 000207 RTS PC ;RETURN TO CALLER

;THIS ROUTINE CALLED TO SET UP FOR PARITY TESTS
5139
5140
5141 023300 012767 000020 155702 SUPPAR: MOV #20,STMP4 ;SET UP FOR 16. LINES
5142 023306 105011 CLR# (R1) ;INIT SCR TO START AT LINE 00
5143 023310 005002 CLR R2 ;INIT INDEX REGISTER FOR RBUF (EVEN)
5144 023312 012703 000200 MOV #200,R3 ;SET UP CONSTANT
5145 023316 012704 000001 MOV #1,R4 ;INIT INDEX REG FOR RBUF (ODD)
5146 023322 012761 033436 000006 IS: MOV #STBUF,CAR(R1) ;LOAD BUS ADDRESS REWG
5147 023330 016761 155660 000010 MOV STMP6,BCR(R1) ;LOAD BYTE COUNT REG
5148 023336 016761 155650 000004 MOV STMP5,LPR(R1) ;LOAD LINE PARAMETERS
5149 023344 105062 032436 CLR# RBUF(R2) ;INIT DATA BYTE IN RBUF TO START AT 000
5150 023350 110364 032436 MOV# R3,RBUF(R4) ;SET CONSTANT IN HIGH BYTE
5151 023354 005211 INC (R1) ;SELECT NEXT LINE
5152 023356 005203 INC R3 ;GENERATE NEW CONSTANT
5153 023360 062702 000002 ADD #2,R2 ;UPDATE POINTERS TO RBUF (EVEN/ODD)
5154 023364 062704 000002 ADD #2,R4

```

```

5155 023370 005367 155614          DEC      STMP4          ;COUNT ONE LINE SETUP
5156 023374 001352                BNE      15          ;BR TILL ALL 16. SET UP
5157 023376 012704 024660          MOV      @MULPTB,R4  ;SET UP TABLE POINTER
5158 023402 016724 155606          2S:    MOV      STMP6,(R4)+ ;SET UP BYTE COUNT ENTRY
5159 023406 022704 024720          CMP      @MULPTB+40,R4 ;SET UP ALL COUNTS ?
5160 023412 001373                BNE      25          ;BR IF NOT
5161 023414 105011                CLRB    (R1)         ;INIT SCR TO SELECT LINE 00
5162 023416 000207                RTS     PC           ;RETURN TO PARITY TEST

;THIS ROUTINE IS USED TO ACCEPT INPUT PARAMETERS FROM THE CONSOLE
;TELETYPE

5164
5165
5166
5167 023420 104400          INPARA: TYPE          ;"ASK FOR NO. WORDS BETWEEN VECTORS"
5168 023422 032306          VCMC          ;READ OCTAL NO. FM TTY
5169 023424 104407          RDOCT        ;GET THE NO. HE TYPED
5170 023426 012600          MOV      (SP)+,RO   ;BR IF HE TYPED <CR>
5171 023430 001412          BEQ      35          ;FOUR WORDS BETWEEN VECTORS ?
5172 023432 022700 000004          CMP      #4,RO     ;BR IF YES
5173 023436 001404          BEQ      25          ;8. WORDS BETWEEN VECTORS ??
5174 023440 022700 000010          CMP      #8,RO     ;BR IF YES
5175 023444 001404          BEQ      35          ;ASK ALL OVER AGAIN
5176 023446 000764          BR      INPARA     ;SET UP CONSTANT IN RO FOUR 4 WORDS
5177 023450 012700 000010          2S:    MOV      #10,RO ;CONTINUE
5178 023454 000402          BR      45          ;SET UP CONSTANT FOR 8. WORDS
5179 023456 012700 000020          3S:    MOV      #20,RO ;RETURN TO CALLER
5180 023462 000207          4S:    RTS     PC

5181
5182
5183
5184 023464 012700 177777          INPARC: MOV      #-1,RO ;SET FLAG IN RO
5185 023470 000167 156452          JMP      BEGINA     ;GO ASK FOR SELECT PARAMETER

5186
5187 023474 012767 177777 001374 INPARX: MOV      #-1,VCFLG ;SET SETUP FLAG
5188 023502 000167 156440          JMP      BEGINA     ;GO START UP

5189
5190
5191 023506 104400          INPAR:  TYPE          ;ASK FOR DEVICE ADDRESS
5192 023510 031663          INMSG1       ;READ IN WHAT IS TYPED
5193 023512 104407          RDOCT        ;GET THE NO. HE TYPED
5194 023514 012601          MOV      (SP)+,R1  ;BR IF DEFAULT
5195 023516 001403          BEQ      INPAR1     ;GO CHECK VALIDITY OF THE ADDR
5196 023520 004767 000106          JSR      PC,CHKADR ;ERROR BRANCH
5197 023524 000770          BR      INPAR

5198
5199 023526 104400          INPAR1: TYPE          ;ASK FOR VECTOR ADDRESS
5200 023530 031727          INMSG2       ;READ IN WHAT HE TYPES
5201 023532 104407          RDOCT        ;GET THE ADDRESS
5202 023534 012601          MOV      (SP)+,R1  ;BR IF DEFAULT
5203 023536 001403          BEQ      INPAR3     ;GO CHECK VALIDITY OF VECTOR
5204 023540 004767 000150          JSR      PC,CHKVCT ;ERROR BRANCH
5205 023544 000770          BR      INPAR1

5206 023546 104400          INPAR3: TYPE          ;ASK FOR DEVICE SELECTION PARAMETER
5207 023550 031776          INMSG3       ;READ IN WHAT HE TYPES
5208 023552 104407          RDOCT

```

```

0209 023554 012601          MOV      (SP)+,R1      ;GET THE SELECT PARAMETER
0210 023556 001402          BEQ      INPAR4      ;BR IF DEFAULT
0211 023560 010167 000622      MOV      R1,DHSEL    ;SET UP DH11 SELECTION PARAMETER

0212 023564 104400          INPAR4: TYPE        ;ASK FOR LINE SELECT PARAMETER
0213 023566 032174          INMSG6
0214 023570 104407          RDOCT
0215 023572 012601          MOV      (SP)+,R1    ;GET WHAT HE TYPES
0216 023574 001403          BEQ      1$         ;GET PARAMETER
0217 023576 010167 000606      MOV      R1,LINSEL  ;BR IF DEFAULT
0218 023602 000403          BR      2$         ;SET UP LINE SELECT PARAMETER
0219 023604 012767 177777 000576 1$: MOV      #-1,LINSEL ;CONTINUE
0220 023612 005777 155320 2$: TST     2$WR      ;SET UP DEFAULT (ALL LINES)
0221 023616 100003          BPL     EXPAR      ;HALT AFTER SET UP ??
0222 023620 104400          TYPE        ;BR IF NOT
0223 023622 032236          INMSG7          ;TYPE CONTINUE MESSAGE PRIOR TO HALTING
0224 023624 000000          HALT
0225 023626 000167 156654      EXPAR: JMP      START2 ;DEPRESS CONTINUE TO RESUME TESTING
                                           ;GO START UP THE PROGRAM

0226 023632 020127 160020      CHKADR: CMP      R1,#160020 ;IS ADDRESS ABOVE OR EQUAL TO LOW LIMIT
0227 023636 002001          BGE     1$         ;BR IF YES
0228 023640 000422          BR      4$         ;BR IF NOT
0229 023642 020127 160420      1$: CMP      R1,#160420 ;IS IT BELOW THE HIGH LIMIT?
0230 023646 002401          BLT     2$         ;BR IF YES
0231 023650 000416          BR      4$         ;BR IF NOT
0232 023652 032701 000017      2$: BIT      #17,R1    ;CORRECT BOUNDARY ?
0233 023656 001013          BNE     4$         ;BR IF NOT
0234 023660 062716 000002      ADD     #2,(SP)    ;MOVE RETURN PC AROUND ERROR BRANCH
0235 023664 012702 024776      MOV     #DHADTB,R2 ;POINT TO BEGIN OF ADDR TABLE
0236 023670 010122      3$: MOV     R1,(R2)+ ;SETR UP A TABLE ENTRY
0237 023672 062701 000020      ADD     #20,R1    ;GENERATE NEXT DH11 ADDR
0238 023676 022702 025036      CMP     #DHADTB+40,R2 ;END OF TABLE ?
0239 023702 001372          BNE     3$        ;BR IF NOT
0240 023704 000402          BR      5$        ;RETURN TO INPUT ROUTINES
0241 023706 104400      4$: TYPE        ;TELL HIM HE GOOFED
0242 023710 032047          INMSG4
0243 023712 000207      5$: RTS     PC      ;RETURN TO INPUT ROUTINES

0244 023714 020127 000300      CHKVCT: CMP      R1,#300 ;IS ADDRESS ABOVE OR EQUAL TO LOW LIMIT
0245 023720 002001          BGE     1$         ;BR IF YES
0246 023722 000421          BR      4$         ;BR IF NOT
0247 023724 020127 001000      1$: CMP      R1,#1000 ;IS IT BELOW THE HIGH LIMIT?
0248 023730 002401          BLT     2$         ;BR IF YES
0249 023732 000415          BR      4$         ;BR IF NOT
0250 023734 032701 000007      2$: BIT      #7,R1    ;CORRECT BOUNDARY ?
0251 023740 001012          BNE     4$         ;BR IF NOT
0252 023742 062716 000002      ADD     #2,(SP)    ;MOVE RETURN PC AROUND ERROR BRANCH
0253 023746 012702 025036      MOV     #DHVCTB,R2 ;POINT TO BEGIN OF VECTOR TABLE
0254 023752 010122      3$: MOV     R1,(R2)+ ;SETR UP A TABLE ENTRY
0255 023754 060001      ADD     #20,R1    ;GENERATE NEXT DH11 ADDR
0256 023756 022702 025076      CMP     #DHVCTB+40,R2 ;END OF TABLE ?
0257 023762 001373          BNE     3$        ;BR IF NOT
0258 023764 000402          BR      5$        ;RETURN TO INPUT ROUTINES

```

```

5263 023766 104400 4S: TYPE ;TELL HIM HE GOOFED
5264 023770 032120 INMSG5
5265 023772 000207 5S: RTS PC ;RETURN TO INPUT ROUTINES
;THESE TWO ROUTINES SERVICE UNEXPECTED BUS ERROR AND RSVD INSTR TRAPS
5269 023774 012767 000340 155176 BUSER: MOV #340,STMP0 ;SAVE THE PSW
5270 024002 010667 155166 MOV SP,SREG6 ;SAVE THE SP
5271 024006 012601 MOV (SP)+,R1 ;GET THE TRAP PC
5272 024010 012602 MOV (SP)+,R2 ;GET THE TRAP PSW
5273 024012 116700 155064 MOV# STSTN,R0 ;GET TEST NO.
5274 024016 012706 001100 MOV #STACK,SP ;RESET THE STACK POINTER
5275 024022 004767 176714 JSR PC,SUER3 ;GO SET UP ERROR INFO
5276 024026 012767 024036 155054 MOV #1$,SLPERR ;ALWAYS COME BACK TO 1$
5277 024034 104027 ERROR 27 ;UNEXPECTED BUS ERROR TRAP
5278 024036 000005 1S: RESET ;PREPARE TO RESTART
5279 024040 004767 000240 JSR PC,CHPS1 ;GO CLEAR PSW
5280 024044 000167 156532 JMP REST1 ;GO RESTART THE PROGRAM
5282 024050 012767 000340 155122 RESERR: MOV #340,STMP0 ;SAVE THE PSW
5283 024056 010667 155112 MOV SP,SREG6 ;SAVE THE SP
5284 024062 012601 MOV (SP)+,R1 ;GET THE TRAP PC
5285 024064 012602 MOV (SP)+,R2 ;GET THE TRAP PSW
5286 024066 116700 155010 MOV# STSTN,R0 ;GET TEST NO.
5287 024072 012706 001100 MOV #STACK,SP ;RESET THE STACK POINTER
5288 024076 004767 176640 JSR PC,SUER3 ;GO SET UP ERROR INFO
5289 024102 012767 024112 155000 MOV #1$,SLPERR ;ALWAYS COME BACK TO 1$
5290 024110 104030 ERROR 30 ;UNEXPECTED RSVD INSTR ERROR TRAP
5291 024112 000005 1S: RESET ;PREPARE TO RESTART
5292 024114 004767 000164 JSR PC,CHPS1 ;GO CLEAR PSW
5293 024120 000167 156456 JMP REST1 ;GO RESTART THE PROGRAM
;THIS ROUTINE IS CALLED WHEN A TEST NEEDS TO RESTORE THE TRAP
;CATCHER IN THE DH11 VECTOR
5298 024124 016703 000252 RESTRP: MOV DHVCT,R3 ;GET VECTOR ADDRESS
5299 024130 010313 MOV R3,(R3) ;RESTORE THE TRAP CATCHER
5300 024132 062723 000002 ADD #2,(R3)+
5301 024136 005023 CLR (R3)+
5302 024140 010313 MOV R3,(R3)
5303 024142 062723 000002 ADD #2,(R3)+
5304 024146 005023 CLR (R3)+
5305 024150 000207 RTS PC ;RETURN TO CALLING TEST
;THIS ROUTINE CALLED BY ANY TEST THAT NEEDS A TIMING WAIT LOOP
; "TIMEA" IS INITIALIZED BY THE CALLING ROUTINE TO THE MINIMUM REQUIRED
; VALUE AND "TIMEB" IS CLEARED TO 000000. IF A TIME OUT OCCURS THIS
; ROUTINE WILL MOVE THE RETURN PC AROUND THE "LOOP" BRANCH BACK IN
; THE ROUTINE THAT CALLED IT TO ALLOW REPORTING AN ERROR MESSAGE
5313 024152 005267 001016 TIMEIT: INC TIMEB ;COUNT B
5314 024156 001005 BNE 1$ ;BR IF NOT ZERO
5315 024160 005367 001006 DEC TIMEA ;COUNT TIME A
5316 024164 001002 BNE 1$ ;BR IF NO TIMEOUT
    
```

```

5317 024166 062716 000002          ADD      #2,(SP)      ;MOVE RETURN PC TO ALLOW ERROR REPORT
5318 024172 000207          IS:      RTS        PC      ;RETURN TO THE CALLING TEST
5319
5320          ;THIS ROUTINE CALLED BY THE AUTO ECHO TEST TO SET UP FOR TRANSFERRING
5321          ;A BINARY COUNT TEST PATTERN ON ALL LINES
5322
5323 024174 012767 000020 155012 SETALL: MOV      #20,STMP6      ;SET UP SIXTEEN LINES
5324 024202 005002          CLR      R2          ;INIT A TABLE INDEX REG
5325 024204 012703 000200          MOV      #200,R3     ;SET UP TO GENERATE HI BYTE OF EXPECTED DATA
5326 024210 012704 000001          MOV      #1,R4      ;SET UP INDEX REG TO ODD BYTES
5327 024214 005011          CLR      (R1)        ;START WITH LINE 00
5328 024216 012761 033436 000006 IS:      MOV      #TBUF,CAR(R1) ;SET UP BUS ADDR REG
5329 024224 012761 177400 000010          MOV      #400,BCR(R1) ;SET UP BYTE COUNT REG
5330 024232 012761 031403 000004          MOV      #31403,LPR(R1) ;SET UP FOR 4800 BAUD/8 BIT CHARS
5331 024240 105062 032436          CLR     RBUF(R2)    ;START WITH DATA CHAR OF 000
5332 024244 110364 032436          MOV     R3,RBUF(R4) ;SET UP HIGH BYTE OF EXPECTED DATA
5333 024250 005211          INC     (R1)        ;GEN NEW LINE NO. IN SCR
5334 024254 005203          INC     R3          ;UPDATE THE POINTERS AND DATA
5335 024254 062702 000002          ADD     #2,R2
5336 024260 062704 000002          ADD     #2,R4
5337 024264 005367 154724          DEC     STMP6       ;COUNT ONE LINE DONE
5338 024270 001352          BNE     IS          ;BR TIL ALL 16 SET UP
5339 024272 016767 000112 000356          MOV     LINSSEL,LINACT ;SET SOFTWARE FLAG FOR ALL LINES ACTIVE
5340 024300 005011          CLR     (R1)        ;PUT SCR REG BACK TO LINE 00
5341 024302 000207          RTS        PC      ;RETURN TO AUTO ECHO TEST

```

```

;THIS ROUTINE IS CALLED TO SET PSW PRIORITY TO 000 IN ORDER
;TO BE LSIII COMPATIBLE

```

```

5347 024304 012746 000000          CHPS1: MOV     #0,-(SP)      ;NEW PSW
5348 024310 012746 024316          MOV     #1$,-(SP)      ;NEW PC
5349 024314 000002          RTI
5350 024316 000207          IS:      RTS        PC      ;CHANGE PSW
;RETURN TO CALLING TEST

```

```

;THIS ROUTINE DOES THE SAME THING EXCEPT IT SET THE PSW
;PRIORITY TO 340 (LEVEL 7 ) TO LOCK OUT INTRs

```

```

5355 024320 012746 000340          CHPS2: MOV     #340,-(SP)  ;NEW PSW
5356 024324 012746 024332          MOV     #1$,-(SP)      ;NEW PC
5357 024330 000002          RTI
5358 024332 000207          IS:      RTS        PC      ;CHANGE THE PSW
;RETURN TO CALLING TEST

```

```

;THIS ROUTINE IS ALSO FOR LSIII COMPATIBILITY AND IT IS CALLED
;TO SAVE THE PSW IN "STMP0"

```

```

5363 024334 005046          SAPS:  CLR     -(SP)      ;TEMP STORAGE TO SAVE PSW
5364 024336 016746 153472          MOV     #34,-(SP)      ;SAVE TRAP VECTOR POINTER
5365 024342 012767 024352 153464          MOV     #1$,34        ;GO TO IS ON TRAP
5366 024350 104400          TRAP
5367 024352 016666 000002 000006 IS:      MOV     2(SP),6(SP)    ;GO TO IT
5368 024350 012716 024366          MOV     #2$,(SP)      ;GET PSW SAVED
5369 024364 000002          RTI
5370 024366 012667 153442          2$:    MOV     (SP)+,34   ;GO TO 2$ ON RTI
;RESTORE VECTOR

```

```
5371 024372 012667 154602      MOV      (SP)+,STMPD      ;FINALLY SAVE PSW IN STMPD
5372 024376 000207              RTS      PC
5373
```


5375
5376
5377
5378
5379
5380
5381
5382
5383
5384
5385
5386
5387
5388
5389
5390
5391
5392
5393
5394
5395
5396
5397
5398
5399
5400
5401
5402
5403
5404
5405
5406
5407
5408
5409
5410
5411
5412
5413
5414
5415
5416
5417
5418
5419
5420
5421
5422
5423
5424
5425
5426
5427
5428

: ADDITIONAL PROGRAM CONSTANTS AND VARIABLES

000002
000004
000006
000010
000012
000014
000016

NRC=2
LPR=4
CAR=6
BCR=10
BAR=12
BKR=14
SSR=16

: INDEX CONST. TO ACCESS NEXT RCVD CHAR REG
: INDEX CONST. TO ACCESS LINE PARAMETER REG.
: INDEX CONST. TO ACCESS CURRENT ADDRESS REG.
: INDEX CONST. TO ACCESS BYTE COUNT REG.
: INDEX CONST. TO ACCESS BUFFER ACTIVE REG.
: INDEX CONST. TO ACCESS BREAK CONTROL REG.
: INDEX CONST. TO ACCESS SILO STATUS REG.

024400 000000
024402 000000
024404 000000
024406 000003
024410 177777
024412 000000
024414 000000

DHADR: 0
DHVCT: 0
SELMSK: 0
DHSSEL: 3
LINSEL: 177777
LINMSK: 0
LMSK1: 0

: HOLDS THE "SCR" ADDRESS OF THE DH11 UNDER TEST
: HOLDS THE 1ST VECTOR ADDRESS OF THE DH11 UNDER TEST
: BIT TST MARKER FOR SELECTING DH11'S
: SPECIFIES DH11'S SELECTED FOR TEST
: SPECIFIES LINES TO TEST
: MARKER USED TO TEST FOR LINES TO TEST
: ALTERNATE MARKER TO SUPPORT THE
: SELECT LINES FEATURE

024416 000004

MSTCLR: .BLKW 4

: FOUR WORD ADDRESS TABLE USED BY THE TEST THAT
: CHECKS OPERATION OF "MASTER CLR"

024426 177777
024430 125252
024434 052525
024438 000000

PATRNA: 177777
125252
052525
000000

: BIT PATTERNS USED WITH "CAR" AND "BCR" TESTS

: TABLE TERMINATOR

024436 000060
024440 000300
024444 000020
024448 000100
024452 000040
024456 000200
024460 000000
024464 000000

PATRNB: 60
300
20
100
40
200
0
0

: BIT PATTERNS USED IN "CAR" MEM EXT BIT TEST

: TABLE TERMINATOR
: TABLE TERMINATOR

: THIS TABLE STORES THE BYTE COUNT AND LINE PARAMETERS FOR THE
: 8 SUBTESTS IN THE MULTILINE PARITY/DATA TEST

024456 177400
024460 027363
024464 177400
024468 027323
024472 177600
024476 027362
024480 177600
024484 027322
024488 177700
024492 027361
024496 177700
024500 027321
024504 177740
024508 027360

PRTYTB: -400
27363
-400
27323
-200
27362
-200
27322
-100
27361
-100
27321
-40
27360

: 256 CHARS
: 2400 BAUD - ODD PARITY - 8 BITS
: 256 CHARS
: 2400 BAUD - EVEN PARITY - 8 BITS
: 128 CHARS
: 2400 BAUD - ODD PARITY - 7 BITS
: 128 CHARS
: 2400 BAUD - EVEN PARITY - 7 BITS
: 64 CHARS
: 2400 BAUD - ODD PARITY - 6 BITS
: 64 CHARS
: 2400 BAUD - EVEN PARITY - 6 BITS
: 32 CHARS
: 2400 BAUD - ODD PARITY - 5 BITS

024512 177740
024514 027320

-40
27320

:32 CHARS
:2400 BAUD - EVEN PARITY - 5 BITS

:THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE AUTO ECHO
:TEST (ALL 1'S DATA TABLE)

024516 100377
024518 100777
024520 101377
024522 101777
024524 102377
024526 102777
024528 103377
024530 103777
024532 104377
024534 104777
024536 105377
024538 105777
024540 106377
024542 106777
024544 107377
024546 107777

RETAB: 100377 :TEST DATA FOR LINE 00
100777 :TEST DATA FOR LINE 01
101377
101777
102377
102777
103377
103777
104377
104777
105377
105777
106377
106777
107377
107777

:TEST DATA FOR LINE 17

:THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE AUTO ECHO
:TEST (ALL 0'S DATA TABLE)

024556 100000
024558 100400
024560 101000
024562 101400
024564 102000
024566 102400
024568 103000
024570 103400
024572 104000
024574 104400
024576 105000
024578 105400
024580 106000
024582 106400
024584 107000
024586 107400

RETAB0: 100000 :TEST DATA FOR LINE 00
100400 :TEST DATA FOR LINE 01
101000
101400
102000
102400
103000
103400
104000
104400
105000
105400
106000
106400
107000
107400

:TEST DATA FOR LINE 17

:THIS TABLE USED BY THE AUTO ECHO TEST 2 TO RESET ACTIVE BIT WHEN A
:LINE IS DONE

024616 000001
024618 000002
024620 000004
024622 000010
024624 000020
024626 000040
024628 000100
024630 000200

LINBIT: BIT00 :DEACTIVATE LINE 00
BIT01 :DEACTIVATE LINE 01
BIT02
BIT03
BIT04
BIT05
BIT06
BIT07

5483 024636 000400
5484 024640 001000
5485 024642 002000
5486 024644 004000
5487 024646 010000
5488 024648 020000
5489 024650 040000
5490 024654 100000
5491
5492 024656 000000
5493
5494
5495
5496
5497
5498
5499 024660 000020
5500
5501
5502
5503 024720 120000
5504 024722 120400
5505 024724 121000
5506 024726 121400
5507 024728 122000
5508 024730 122400
5509 024732 122800
5510 024734 123000
5511 024736 123400
5512 024740 124000
5513 024742 124400
5514 024744 125000
5515 024746 125400
5516 024750 126000
5517 024752 126400
5518 024754 127000
5519 024756 127400
5520
5521 024760 131177
5522 024762 046600
5523 024764 177767
5524 024766 177777
5525 024770 100077
5526 024772 042200
5527 024774 030100
5528
5529
5530
5531
5532 024776 160020
5533 025000 160040
5534 025002 160060
5535 025004 160100
5536 025006 160120

BIT08
BIT09
BIT10
BIT11
BIT12
BIT13
BIT14
BIT15 ;DEACTIVATE LINE 17

LINACT: 0 ;MAINTAINS STATUS OF ACTIVE LINES
;DURING AUTO ECHO TEST 2

;THIS TABLE CONTAINS 16. COUNTERS USED BYN THE MULTI-LINE
;PARITY TEST TO KEEP TRACK OF TOTAL CHARS RECEIVED

MULPTB: .BLKW 16. ;SIXTEEN WORD COUNTERS TABLE

;THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE BREAK BIT
;TEST
BRKTAB: 120000 ;TEST DATA FOR LINE 00
120400 ;TEST DATA FOR LINE 01
121000
121400
122000
122400
123000
123400
124000
124400
125000
125400
126000
126400
127000
127400 ;TEST DATA FOR LINE 17

RGMSK1: 131177 ;MASK TO SPECIFY R/W BITS FOR NORMAL "SCR" REG TEST
RGMSK2: 46600 ;MASK TO SPECIFY READ ONLY BITS IN "SCR" FOR NORMAL MODE TEST
RGMSK3: 177767 ;MASK TO SPECIFY R/W BITS IN "LPR"
RGMSK4: 177777 ;MASK TO SPECIFY R/W BITS IN "BKR"
RGMSK5: 100077 ;MASK TO SPECIFY R/W BITS IN "SSR"
RGMSK6: 42200 ;MASK TO SPECIFY READ ONLY BITS IN "SCR" FOR MAINT. MODE TEST
INTMSK: 30100 ;MASK USED TO SELECT INTR BITS TO TEST

;DH11 ADDRESS TABLE - THIS TABLE CONTAINS THE "SCR" ADDRESS FOR UP TO
;SIXTEEN DH11'S

DHAOTB: 160020 ;ADDRESS OF FIRST DH11
160040 ;ADDRESS OF SECOND DH11
160060
160100
160120

5537 025010 160140
5538 025010 160160
5539 025010 160160
5540 025010 160200
5541 025010 160200
5542 025016 160220
5543 025016 160220
5544 025016 160240
5545 025016 160240
5546 025016 160260
5547 025016 160260
5548 025030 160300
5549 025030 160300
5550 025030 160320
5551 025030 160320
5552 025030 160340
5553 025030 160340
5554 025030 160360
5555 025030 160360
5556 025034 160400

160140
160160
160160
160200
160200
160220
160220
160240
160240
160260
160260
160300
160300
160320
160320
160340
160340
160360
160360
160400

;ADDRESS OF THE LAST DH11

;DH11 VECTOR TABLE - THIS TABLE CONTAINS THE VECTOR ADDRESSES FOR UP
;TO SIXTEEN DH11'S

5557 025036 000330
5558 025036 000350
5559 025040 000370
5560 025040 000410
5561 025040 000430
5562 025040 000430
5563 025040 000450
5564 025040 000450
5565 025040 000470
5566 025040 000470
5567 025040 000510
5568 025040 000510
5569 025040 000530
5570 025040 000530
5571 025040 000550
5572 025040 000550
5573 025040 000570
5574 025040 000570
5575 025040 000610
5576 025040 000610
5577 025040 000630
5578 025040 000630
5579 025040 000650
5580 025040 000650
5581 025070 000670
5582 025070 000670
5583 025074 000710

DHVCTB: 330
350
370
410
430
450
470
510
530
550
570
610
630
650
670
710

;ADDRESS OF VECTOR FOR FIRST DH11
;ADDRESS OF VECTOR FOR SECOND DH11

;ADDRESS OF VECTOR FOR LAST DH11

025076 000000

VCFLG: 0

;VECTOR SET UP FLAGG

;BR PRIORITY LEVEL TABLE - THIS TABLE CONTAINS THE PRIORITY LEVELS
;FOR UP TO SIXTEEN DH11'S - THE RCVR LEVEL IS STORED IN THE LOW BYTE
;AND THE XMTTR LEVEL IN THE HIGH BYTE

5576 025100 120240
5577 025100 120240
5578 025100 120240
5579 025100 120240
5580 025100 120240
5581 025100 120240
5582 025100 120240
5583 025100 120240
5584 025100 120240
5585 025100 120240
5586 025100 120240
5587 025100 120240
5588 025100 120240
5589 025100 120240
5590 025100 120240

BRVL: 120240
120240
120240
120240
120240
120240
120240
120240
120240
120240
120240
120240
120240
120240
120240

;BRLEVELS FOR FIRST DH11
;BR LEVELS FOR SECOND DH11

```

5591 025136 120240 ;BR LEVELS FOR LAST DH11
5592 025140 000 ;BR LEVEL FOR RCVR
5593 025141 000 ;BR LEVEL FOR XMITTER
5594 025142 000000 ;CONTAINS NUMBER OF THE DH11 UNDER TEST
5595 025144 000000 ;CONTAINS NUMBER OF THE LINE UNDER TEST
5596 025146 000000 ;LOCATION TO SAVE LINE NUMBER
5597 ;ADDRESS POINTERS TO SET UP TABLES WHEN INPUTTING PARAMETERS
5598
5599 025150 000000 ;POINTS TO ADDRESS TABLE
5600 025152 000000 ;POINTS TO VECTOR TABLE
5601 025154 000000 ;POINTS TO BR LEVEL TABLE
5602
5603 025156 000000 ;DATA BUFFER FOR BASIC DATA TEST
5604 025160 000037 ;TEST DATA FOR FIVE BIT CHAR
5605 025162 000077 ;TEST DATA FOR SIX BIT CHAR
5606 025164 000177 ;TEST DATA FOR SEVEN BIT CHAR
5607 025166 000377 ;TEST DATA FOR EIGHT BIT CHAR
5608 025170 000000 ;FLAG TO ALLOW PRINTING TITLE ONLY ONCE
5609 025172 000000 ;GENERAL PURPOSE TIMERS
5610 025174 000000
5611 025176 000000 ;TIMER FOR TIMING TESTS
5612 025200 000000 ;CONTAINS TWO NULL CHARS USED BY BREAK TEST
5613
5614
5615

```

5617
5618
5619
5620
5621
5622
5623

;ERROR MESSAGE INFORMATION - MESSAGE BUFFERS AND POINTERS

;INFORMATION FOR MESSAGE 1

025202 044104 030461 051040
025202 043505 051511 042524
025202 020122 042522 042506
025202 042522 041516 020105
025202 040503 051523 042105
025202 052040 046511 047505
025202 021222 000000

EM1: .ASCIZ 'DH11 REGISTER REFERENCE CAUSED TIMEOUT'

025202 040000 050050 024503
025202 020040 020040 050050
025202 024523 020040 020040
025202 051450 024520 020040
025202 020040 042524 052123
025202 020040 042040 053105
025314 042101 020122 051040
025322 043505 042101 000122

DH1: .ASCIZ ' (PC) (PS) (SP) TEST DEVADR REGADR'

5625
5626
025330 001116 001200 001174
025336 001160 001162 001164
025344 000000

.EVEN
DT1: .WORD SERRPC, STMPO, SREG6, SREG0, SREG1, SREG2, 0

5627
5628
5629
5630

;INFORMATION FOR MESSAGE 2

025346 054523 052123 046505
025354 041440 047117 051124
025362 046117 051040 043505
025370 051511 042524 020122
025376 051105 047522 000122
5631 025404 024040 041520 020051
025412 020040 024040 051520
025420 020051 020040 024040
025426 050123 020051 020040
025434 052040 051505 020124
025442 020040 042504 040526
025450 051104 020040 042522
025456 040507 051104 020040
025464 053440 051501 020040
025472 020040 051440 041057
025500 000000

EM2: .ASCIZ 'SYSTEM CONTROL REGISTER ERROR'

DH2: .ASCIZ ' (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B'

5632
5633
025502 001116 001200 001174
025510 001160 001162 001164
025516 001166 001170 000000

.EVEN
DT2: .WORD SERRPC, STMPO, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4, 0

5634
5635
5636
5637

;INFORMATION FOR MESSAGE 3

025524 044104 030461 046440
025532 051501 042524 020122
025540 046103 040505 020122

EM3: .ASCIZ 'DH11 MASTER CLEAR FAILED TO CLR SPECIFIED REG'

025546 040506 046111 042105
025554 052040 020117 046103
025562 020122 050123 041505
025570 043111 042511 020104
025576 042522 000107

5638
5639
5640
5641

; INFORMATION FOR MESSAGE 4

025602 044514 042516 050040
025610 051101 046501 052105
025616 051105 051040 043505
025624 051511 042524 020122
025632 051105 047522 000122

EM4: .ASCIZ 'LINE PARAMETER REGISTER ERROR'

5642
5643
5644
5645

; INFORMATION FOR MESSAGE 5

025640 051102 040505 020113
025646 047503 052116 047522
025654 020114 042522 044507
025662 052123 051105 042440
025670 051122 051117 000

EM5: .ASCIZ 'BREAK CONTROL REGISTER ERROR'

5646
5647
5648
5649

; INFORMATION FOR MESSAGE 6

025675 123 046111 020117
025702 052123 052101 051525
025710 051040 043505 051511
025716 042524 020122 051105
025724 047522 000122

EM6: .ASCIZ 'SILO STATUS REGISTER ERROR'

5650
5651
5652
5653

; INFORMATION FOR MESSAGE 7

025730 052503 051122 047105
025736 020124 042101 051104
025744 051505 020123 042522
025752 044507 052123 051105
025760 042440 051122 051117
025766 026440 046040 047111
025774 020105 054043 000130

EM7: .ASCIZ 'CURRENT ADDRESS REGISTER ERROR - LINE #XX'

5654
5655
5656
5657

; INFORMATION FOR MESSAGE 10

026002 054502 042524 041440
026010 052517 052116 051105
026016 051040 043505 051511
026024 042524 020122 051105
026032 047522 020122 020055
026040 044514 042516 021440
026046 054130 000

EM10: .ASCIZ 'BYTE COUNTER REGISTER ERROR - LINE #XX'

5658
5659
5660
5661

; INFORMATION FOR MESSAGE 11

026051 125 042516 050130
026056 041505 042524 020104

EM11: .ASCIZ 'UNEXPECTED DH11 RCVR INTERRUPT'

026064 044104 030461 051040
026072 053103 020122 047111
026100 042524 051122 050125
026106 000124

5662
5663
5664
5665

;INFORMATION FOR MESSAGE 12

026110 047125 054105 042520
026116 052103 042105 042040
026124 030510 020061 046530
026132 052111 051124 044440
026140 052116 051105 052522
026146 052120 000

EM12: .ASCIZ 'UNEXPECTED DH11 XMITTR INTERRUPT'

5666
5667
5668
5669

;INFORMATION FOR MESSAGE 13

026151 103 040510 020122
026156 053101 044501 040514
026164 046102 020105 040506
026172 046111 042105 052040
026200 020117 042507 042516
026206 040522 042524 051040
026214 053103 020122 047111
026222 042524 051122 050125
026230 000124

EM13: .ASCIZ 'CHAR AVAILABLE FAILED TO GENERATE RCVR INTERRUPT'

5670
5671
5672
5673

;INFORMATION FOR MESSAGE 14

026232 051124 047101 046523
026240 052111 042524 020122
026246 050116 020122 047514
026254 044507 020103 051105
026262 047522 020122 020055
026270 044514 042516 021440
026276 020040 000

EM14: .ASCIZ 'TRANSMITTER NPR LOGIC ERROR - LINE # '

5674
5675
5676
5677

;INFORMATION FOR MESSAGE 15

026301 130 044515 052124
026306 020122 040506 046111
026314 042105 052040 020117
026322 047111 042524 051122
026330 050125 020124 020055
026336 044514 042516 021440
026344 020040 000

EM15: .ASCIZ 'XMITTR FAILED TO INTERRUPT - LINE # '

5678
5679
5680
5681

;INFORMATION FOR MESSAGE 16

026347 122 053103 020122
026354 040506 046111 042105
026362 052040 020117 047111
026370 042524 051122 050125
026376 000124

EM16: .ASCIZ 'RCVR FAILED TO INTERRUPT'

5682


```

5683 ;INFORMATION FOR MESSAGE 17
5684
5685 026400 051124 047101 046523 EM17: .ASCIZ 'TRANSMITTER TIMING ERROR - LINE # '
      026406 052111 042524 020122
      026414 044524 044515 043516
      026422 042440 051122 051117
      026430 026440 046040 047111
5686 026436 020105 020043 000040
      026444 024040 041520 020051 DM6: .ASCIZ ' (PC) (PS) (SP) TEST DEVRDR SPEED TIMEB TIMEC'
      026452 020040 024040 051520
      026460 020051 020040 024040
      026466 050123 020051 020040
      026474 052040 051505 020124
      026502 020040 042504 040526
      026510 051104 020040 050123
      026516 042505 020104 020040
      026524 044524 042515 020102
      026532 020040 044524 042515
      026540 000103

5687 ;INFORMATION FOR MESSAGE 20
5688
5689 026542 042522 042503 053111 EM20: .ASCIZ 'RECEIVER TIMING ERROR - LINE # '
      026550 051105 052040 046511
      026556 047111 020107 051105
      026564 047522 020122 020055
      026572 044514 042516 021440
      026600 020040 000

5691 ;INFORMATION FOR MESSAGE 21
5692
5693 026603 122 053103 020122 EM21: .ASCIZ 'RCVR FAILED TO INTERRUPT - LINE # '
      026610 040506 046111 042105
      026616 052040 020117 047111
      026624 042524 051122 050125
      026632 020124 020055 044514
      026640 042516 021440 020040
      026646 000

5695 ;INFORMATION FOR MESSAGE 22
5696
5697 026647 103 040510 020122 EM22: .ASCIZ 'CHAR AVAIL FAILED TO SET ON TIME - LINE # '
      026654 053101 044501 020114
      026662 040506 046111 042105
      026670 052040 020117 042523
      026676 020124 047117 052040
      026704 046511 020105 020055
      026712 044514 042516 021440
      026720 020040 000

5699 ;INFORMATION FOR MESSAGE 23
5700
5701 026723 102 051501 041511 EM23: .ASCIZ 'BASIC DATA TEST ERROR - LINE # '
      026730 042040 052101 020101

```


5721	027304	020040	042524	052123	
	027312	020040	052040	050122	
	027320	041520	020040	052040	
	027326	050122	051520	000040	
5722	027334	001116	001200	001174	.EVEN
	027342	001160	001162	001164	DT3: .WORD SERRPC,STMP0,SREG6,SREG0,SREG1,SREG2,0
	027350	000000			
5723					
5724					;INFORMATION FOR MESSAGE 30
5725					
5726	027352	047125	054105	042520	EM30: .ASCIZ 'UNEXPECTED RSVD INSTR TRAP'
	027360	052103	042105	051040	
	027366	053123	020104	047111	
	027374	052123	020122	051124	
	027402	050101	000		
5727					
5728					;INFORMATION FOR MESSAGE 31
5729					
5730	027405	101	052125	020117	EM31: .ASCIZ 'AUTO ECHO DATA COMPARE ERROR - LINE # '
	027412	041505	047510	042040	
	027420	052101	020101	047503	
	027426	050115	051101	020105	
	027434	051105	047522	020122	
	027442	020055	044514	042516	
	027450	021440	020040	000	
5731	027455	040	050050	024503	DH4: .ASCIZ ' (PC) (PS) (SP) TEST WASADR SBADR WAS S/B'
	027462	020040	020040	050050	
	027470	024523	020040	020040	
	027476	051450	024520	020040	
	027504	020040	042524	052123	
	027512	020040	053440	051501	
	027520	042101	020122	051440	
	027526	040502	051104	020040	
	027534	020040	040527	020123	
	027542	020040	020040	027523	
	027550	000102			
5732					
5733					;INFORMATION FOR MESSAGE 32
5734					
5735	027552	052501	047524	042440	EM32: .ASCIZ 'AUTO ECHO TEST TIMEOUT - LINE # '
	027560	044103	020117	042524	
	027566	052123	052040	046511	
	027574	047505	052125	026440	
	027602	046040	047111	020105	
	027610	020043	000040		
5736	027614	024040	041520	020051	DH5: .ASCIZ ' (PC) (LPRG) TEST'
	027622	020040	046050	051120	
	027630	024507	020040	052040	
	027636	051505	000124		
5737					
5738	027642	001116	001200	001204	.EVEN
	027650	000000			DT4: .WORD SERRPC,STMP0,STMP2,0
5739					

5740
5741
5742 027652 040520 044522 054524
027660 046040 043517 041511
027666 052040 051505 020124
027674 051105 047522 020122
027702 020055 044514 042516
027710 021440 020040 000

; INFORMATION FOR MESSAGE 33
EM33: .ASCIZ 'PARITY LOGIC TEST ERROR - LINE # '

5743
5744
5745
5746 027715 115 046125 044524
027722 046055 047111 020105
027730 040520 044522 054524
027736 042040 052101 020101
027744 042524 052123 042440
027752 051122 051117 025440
027760 046040 047111 020105
027766 020043 020040 020055
027774 052523 052102 051505
030002 020124 020043 000040

; INFORMATION FOR MESSAGE 34
EM34: .ASCIZ 'MULTI-LINE PARITY DATA TEST ERROR - LINE # - SUBTEST # '

5747
5748
5749
5750 030010 052515 052114 026511
030016 044514 042516 050040
030024 051101 052111 020131
030032 040504 040524 052040
030040 051505 020124 044524
030046 042515 052517 000124
5751 030054 024040 041520 020051
030062 020040 046050 051120
030070 024507 020040 041501
030076 046124 047111 000

; INFORMATION FOR MESSAGE 35
EM35: .ASCIZ 'MULTI-LINE PARITY DATA TEST TIMEOUT'

5752
5753 030104 001116 001200 001206
030112 000000

.EVEN
DT6: .WORD SERRPC, STMPD, STMP3, 0

5754
5755
5756
5757 030114 044103 051101 040440
030122 040526 046111 041101
030130 042514 052040 046511
030136 047505 052125 000

; INFORMATION FOR MESSAGE 36
EM36: .ASCIZ 'CHAR AVAILABLE TIMEOUT'

5758
5759
5760
5761 030143 104 052101 020101
030150 047503 050115 051101
030156 020105 051105 047522
030164 020122 020055 044514
030172 042516 021440 020040
030200 000

; INFORMATION FOR MESSAGE 37
EM37: .ASCIZ 'DATA COMPARE ERROR - LINE # '

5762

```

5763 ;INFORMATION FOR MESSAGE 40
5764
5765 030201 102 043125 042506 EM40: .ASCIZ 'BUFFER ACTIVE REG ERROR - LINE # '
      030206 020122 041501 044524
      030214 042526 051040 043505
      030222 042440 051122 051117
      030230 026440 046040 047111
      030236 020105 020043 000040

5766 ;INFORMATION FOR MESSAGE 41
5767
5768
5769 030244 041522 051126 043040 EM41: .ASCIZ 'RCVR FALSE INTERRUPT'
      030252 046101 042523 044440
      030260 052116 051105 052522
      030266 052120 000

5770 ;INFORMATION FOR MESSAGE 42
5771
5772
5773 030271 123 046111 020117 EM42: .ASCIZ 'SILO OVERFLOW ERROR'
      030276 053117 051105 046106
      030304 053517 042440 051122
      030312 051117 000

5774 ;INFORMATION FOR MESSAGE 43
5775
5776
5777 030315 123 046111 020117 EM43: .ASCIZ 'SILO OVERFLOW FAILED TO GENERATE RCVR INTERRUPT'
      030322 053117 051105 046106
      030330 053517 043040 044501
      030336 042514 020104 047524
      030344 043440 047105 051105
      030352 052101 020105 041522
      030360 051126 044440 052116
      030366 051105 052522 052120
      030374 000

5778 ;INFORMATION FOR MESSAGE 44
5779
5780
5781 030375 116 047117 042440 EM44: .ASCIZ 'NON EX MEMORY FAILED TO GENERATE XMITTR INTERRUPT'
      030402 020130 042515 047515
      030410 054522 043040 044501
      030416 042514 020104 047524
      030424 043440 047105 051105
      030432 052101 020105 046530
      030440 052111 051124 044440
      030446 052116 051105 052522
      030454 052120 000

5782 ;INFORMATION FOR MESSAGE 45
5783
5784
5785 030457 130 044515 020124 EM45: .ASCIZ 'XMIT DONE FAILED TO GENERATE XMITTR INTERRUPT'
      030464 047504 042516 043040
      030472 044501 042514 020104
      030500 047524 043440 047105
      030506 051105 052101 020105
    
```

030514 046530 052111 051124
030522 044440 052116 051105
030530 052522 052120 000

5786
5787
5788
5789

;INFORMATION FOR MESSAGE 46

EM46: .ASCIZ 'CURRENT ADDRESS MEMORY PATTERNS TEST ERROR - LINE # '

030535 103 051125 042522
030542 052116 040440 042104
030550 042522 051523 046440
030556 046505 051117 020131
030564 040520 052124 051105
030572 051516 052040 051505
030600 020124 051105 047522
030606 020122 020055 044514
030614 042516 021440 020040
030622 000

5790

DH10: .ASCIZ ' (PC) LINEAR PATTRN TEST DEVAOR REGAOR WAS S/B'

030623 040 050050 024503
030630 020040 046440 047111
030636 053505 020123 050040
030644 052101 051124 020116
030652 020040 042524 052123
030660 020040 042040 053105
030666 042101 020122 051040
030674 043505 042101 020122
030702 020040 040527 020123
030710 020040 020040 027523
030716 000102

5791
5792

.EVEN
DTS: .WORD SERRPC, STMP0, STMP1, SREG0, SREG1, SREG2, SREG3, SREG4, 0

030720 001116 001200 001202
030726 001160 001162 001164
030734 001166 001170 000000

5793
5794
5795
5796

;INFORMATION FOR MESSAGE 47

EM47: .ASCIZ 'BYTE COUNT MEMORY PATTERNS TEST ERROR - LINE # '

030742 054502 042524 041440
030750 052517 052116 046440
030756 046505 051117 020131
030764 040520 052124 051105
030772 051516 052040 051505
031000 020124 051105 047522
031006 020122 020055 044514
031014 042516 021440 020040
031022 000

5797
5798
5799

;INFORMATION FOR MESSAGE 50

EM50: .ASCIZ 'TEST TIMEOUT WAITING FOR XMIT DONE - LINE # '

031023 124 051505 020124
031030 044524 042515 052517
031036 020124 040527 052111
031044 047111 020107 047506
031052 020122 046530 052111
031060 042040 047117 020105
031066 020055 044514 042516
031074 021440 020040 000

5800

```

5801 ;INFORMATION FOR MESSAGE 51
5802
5803 031101 116 051120 046040 EMS1: .ASCIZ 'NPR LOGIC TEST 2 ERROR'
      031106 043517 041511 052040
      031111 050517 020124 020063
      031115 050517 047503 000123
5804 031116 040440 041500 020051 DH11: .ASCIZ ' (PC) LINACT LINCHK TEST DEVADR REGADR WAS S/B'
      031118 020040 044514 040516
      031119 020040 020040 044514
      031119 041516 045510 020040
      031119 052040 051500 020124
      031119 020040 042504 040526
      031119 051104 020040 042522
      031202 040507 051104 020040
      031210 053440 051501 020040
      031216 020040 051440 041057
      031224 000

```

```

5805 ;INFORMATION FOR MESSAGE 52
5806
5807
5808 031225 102 051501 041511 EMS2: .ASCIZ 'BASIC DATA COMPARE ERROR'
      031232 042040 052101 020101
      031240 047503 050115 051101
      031246 020105 051105 047522
      031254 000122

```

```

5809 ;INFORMATION FOR MESSAGE 53
5810
5811
5812 031256 024040 041520 020051 DH12: .ASCIZ ' (PC) SPEED (SP) TEST DEVADR REGADR WAS S/B'
      031264 020040 050123 042505
      031272 020104 020040 024040
      031300 050123 020051 020040
      031306 052040 051505 020124
      031314 020040 042504 040526
      031322 051104 020040 042522
      031330 040507 051104 020040
      031336 053440 051501 020040
      031344 020040 051440 041057
      031352 000

```

```

5813 ;INFORMATION FOR MESSAGE 55
5814
5815
5816 031353 040 050050 024503 DH13: .ASCIZ ' (PC) (PS) (SP) TEST DEVADR CHRLNG SCRNAS SCRS/B'
      031360 020040 020040 050050
      031366 024523 020040 020040
      031374 051450 024520 020040
      031402 020040 042524 052123
      031410 020040 042040 053105
      031416 042101 020122 041440
      031424 051110 047114 020107
      031432 051440 051103 040527
      031440 020123 051440 051103
      031446 027523 000102

```

5817

```

5818 ;INFORMATION FOR MESSAGE 56
5819
5820 031452 053117 051105 052522 EMS6: .ASCIZ 'OVERRUN BIT FAILED TO SET - LINE # '
      031460 020116 044502 020124
      031466 040506 046111 042105
      031474 052040 020117 042523
      031502 020124 020055 044514
      031510 042516 021440 020040
      031516      000

5821 ;INFORMATION FOR MESSAGE 57
5822
5823
5824 031517      123 047524 040522 EMS7: .ASCIZ 'STORAGE OVERFLOW BIT FAILED - LINE # '
      031524 042507 047440 042526
      031532 043122 047514 020127
      031540 044502 020124 040506
      031546 046111 042105 026440
      031554 046040 047111 020105
      031562 020043 000040

5825 .EVEN
5826 ;MISCELLANEOUS MESSAGES
5827
5828 031566 005015 040515 047111 TITLE: .ASCIZ <15><12>'MAINDEC-11-DZDMM-A DH11 DIAGNOSTIC'<15><12>
      031574 042504 026503 030461
      031602 042055 042132 046510
      031610 040455 020040 041104
      031616 030461 042040 040511
      031624 047107 051517 044524
      031632 006503 000012
5829 031636 005015 042524 052123 TITLE2: .ASCIZ <15><12>'TESTING DH11 # ' <15><12>
      031644 047111 020107 044104
      031652 030461 021440 020040
      031660 005015      000
5830 031663      015 052012 050131 INMSG1: .ASCIZ <15><12>'TYPE SCR ADDRESS FOR FIRST DH11'<15><12>
      031670 020105 041523 020122
      031676 042101 051104 051505
      031704 020123 047506 020122
      031712 044506 051522 020124
      031720 044104 030461 005015
      031726      000
5831 031727      015 052012 050131 INMSG2: .ASCIZ <15><12>'TYPE VECTOR ADDRESS FOR FIRST DH11'<15><12>
      031734 020105 042526 052103
      031742 051117 040440 042104
      031750 042522 051523 043040
      031756 051117 043040 051111
      031764 052123 042040 030510
      031772 006461 000012
5832 031776 005015 054524 042520 INMSG3: .ASCIZ <15><12>'TYPE DH11 DEVICE SELECTION PARAMETER'<15><12>
      032004 042040 030510 020061
      032012 042504 044526 042503
      032020 051440 046105 041505
      032026 044524 047117 050040
      032034 051101 046501 052105
      032042 051105 005015      000

```



```

5833 032047 015 044412 053116 INMSG4: .ASCIZ <15><12>'INVALID DH11 SCR ADDRESS - TRY AGAIN'<15><12>
      032054 046101 042111 042040
      032062 030510 020061 041523
      032070 020123 042101 051104
      032076 051505 020123 020055
      032104 051124 020131 043501
      032112 044501 006516 000012
5834 032120 005015 047111 040526 INMSG5: .ASCIZ <15><12>'INVALID DH11 VECTOR ADDRESS - TRY AGAIN'<15><12>
      032128 044514 020104 044104
      032136 030461 053040 041505
      032144 047524 020122 042101
      032152 051104 051505 020123
      032160 020055 051124 020131
      032168 043501 044501 006516
      032172 000012
5835 032174 005015 054524 042520 INMSG6: .ASCIZ <15><12>'TYPE LINE SELECTION PARAMETER'<15><12>
      032202 046040 047111 020105
      032210 042523 042514 052103
      032216 047511 020116 040520
      032224 040522 042515 042524
      032232 006522 000012
5836 032236 005015 042504 051120 INMSG7: .ASCIZ <15><12>'DEPRESS "CONTINUE" TO START TESTING'<15><12>
      032244 051505 020123 041442
      032252 047117 044524 052516
      032260 021105 052040 020117
      032266 052123 051101 020124
      032274 042524 052123 047111
      032302 006507 000012
5837
5838 032306 005015 054524 042520 VCNC: .ASCIZ <15><12>'TYPE NO. OF WORDS (OCTAL) BETWEEN VECTORS (4 OR 10)'<15><12>
      032314 047040 027117 047440
      032322 020106 047527 042122
      032330 020123 047450 052103
      032336 046101 020051 042502
      032344 053524 042505 020116
      032352 042526 052103 051117
      032360 020123 032050 047440
      032366 020122 030061 006451
      032374 000012
5839 .EVEN
5840 ;SIXTEEN CHAR COUNTERS USED BY THE AUTO ECHO TEST #3
5841
5842 032376 000020 RCNT: .BLKW 16.
5843
5844 ;256. WORD RECEIVER INPUT BUFFER
5845
5846 032436 000400 RBUF: .BLKW 256.
5847
5848 ;256(10) BYTE TRANSMITTER OUTPUT DATA BUFFER
5849
5850
5851 .EVEN
5852 033436 000400 TBUF: .BLKB 256.
5853
    
```

H06

MAINDEC-11-DZDMM-A MACY11 27(663) 12-DEC-75 08:41 PAGE 94-17
DZDMM.A.P11 POWER DOWN AND UP ROUTINES

SEQ 0278

5854

000001

.END

CROSS REFERENCE TABLE

AVECT2= 000000	2259												
BAR = 000012	3644*	3675	3679	3735*	3996*	4065*	4130*	4196*	4253*	4306	4368*	4429*	4490
	4496	4540*	4622*	4707*	4774*	4806*	4810	4819	4879*	4883	4890	4892	4927*
	5383*												
BCR = 000010	3075	3155	3235	3326	3642*	3698	3703	3760	3771	3994*	4062*	4127*	4192*
	4249*	4320	4365*	4534*	4617*	4697*	4772*	4803*	4877*	4925*	5120*	5132*	5147*
	5329*	5382*											
BEGIN 002144	2259												
BEGINA 002146	2259												
BIT0 = 000001	2259												
BIT00 = 000001	2259												
BIT01 = 000002	2259	5475											
BIT02 = 000004	2259	5476											
BIT03 = 000010	2259	5477											
BIT04 = 000020	2259	5478											
BIT05 = 000040	2259	5479											
BIT06 = 000100	2259	5480											
BIT07 = 000200	2259	3478	3807	5481									
BIT08 = 000400	2259	3479	3657	4081	4147	5482							
BIT09 = 001000	2259	4985	5483										
	5484	2767	2769	2780	2781	2792	3477	3517	3557	3597	4252	4985	4986
BIT1 = 000002	2259												
BIT10 = 002000	2259	3559	5485										
BIT11 = 004000	2259	2688	3439	3474	3514	3554	3594	3640	3709	3730	3805	3858	3891
	3935	3990	4060	4125	4190	4246	4358	4419	4445	4452	4466	4497	4511
	4519	4611	4692	4767	4800	4873	4922	4985	5486				
BIT12 = 010000	2259	3519	5487										
BIT13 = 020000	2259	3558	3598	3645	4986	5488							
BIT14 = 040000	2259	3518	4000	4006	4022	4449	4953	4985	5489				
BIT15 = 100000	2259	3599	3747	3806	3861	3865	3896	3899	3939	3943	4083	4147	4554
	4583	4682	4745	4907	4940	4953	5490						
BIT2 = 000004	2259												
BIT3 = 000010	2259												
BIT4 = 000020	2259												
BIT5 = 000040	2259												
BIT6 = 000100	2259												
BIT7 = 000200	2259	3981											
BIT8 = 000400	2259												
BIT9 = 001000	2259												
BKR = 000014	2679	2843	2898*	2952	3014*	3397*	4367*	4805*	5384*				
BPTVEC= 000014	2259												
BRKTAB 024720	4753	5504*											
BRLVL 025100	2621	5576*											
BRPTR 025154	2621*	2628*	2636	4977*	5601*								
BUSER 023774	2599	5269*											
CAR = 000006	3029	3121	3189	3281	3381*	3391*	3398	3686	3692	3776	3787	3993*	4063*
	4128*	4193*	4248*	4334	4366*	4535*	4536*	4615*	4616*	4696*	4771*	4802*	4876*
	4924*	5119*	5146*	5328*	5381*								
CHKACR 023632	5195	5229*											
CHKVCT 023714	5203	5248*											
CHPS1 024304	3442	3480	3520	3560	3600	3646	3715	3808	3844	4432	4436	5279	5292
	5347*												
CHPS2 024320	3438	3476	3516	3556	3596	3639	3803	4423	5355*				
CLCABC 023176	3638	3731	4532	4768	4874	5117*							

EN30	027352	2424	5728#																	
EN31	027409	4131	4658#	4673	4738	5730#														
EN32	027552	4138	4635#	5735#																
EN33	027652	4146	4400#	5742#																
EN34	027715	4153	4473#	4476	5746#															
EN35	030010	4160	5750#																	
EN36	030114	4167	5757#																	
EN37	030143	4174	4295#	5761#																
EN4	025502	4181	5641#																	
EN40	030201	4188	4316#	5765#																
EN41	030244	4195	5769#																	
EN42	030271	4202	5773#																	
EN43	030315	4209	5777#																	
EN44	030379	4216	5781#																	
EN45	030457	4223	5785#																	
EN46	030535	4230	3223#	3313	5789#															
EN47	030743	4237	3268#	3358	5796#															
EN48	025640	4244	5645#																	
EN50	031023	4251	5558#	3753	4088	4945	5799#													
EN51	031101	4258	5803#																	
EN52	031223	4265	5808#																	
EN53	031453	4272	4966#	5820#																
EN57	031517	4279	4013#	4033	4047	5824#														
EN6	025675	4286	5649#																	
EN7	025730	4293	3059#	3128	3410	4344	5653#													
ENDA	017674	4300	4972#	4981																
ERRVEC#	000004	4307	2599#	2600#	2652	2653#	2668#	4985#												
EXPAR	023626	4314	5226#																	
GNS	#####	4321	4994#																	
INMSC1	031663	4328	5830#																	
INMSC2	031727	4335	5831#																	
INMSC3	031776	4342	5832#																	
INMSC4	032047	4349	5833#																	
INMSC5	032120	4356	5834#																	
INMSC6	032174	4363	5835#																	
INMSC7	032236	4370	5836#																	
INPAR	023506	4377	5190#	5196																
INPARA	023420	4384	5167#	5176																
INPARC	023464	4391	5184#																	
INPARX	023474	4398	5187#																	
INPAR1	023526	4405	5198#	5204																
INPAR3	023546	4412	5202#	5206#																
INPAR4	023564	4419	5213#																	
INTMSK	024774	4426	5227#																	
IOTVEC#	000020	4433	5236#																	
LDSCR	023240	4440	5130#																	
LDTBF1	022610	4447	5005#																	
LINACT	024656	4454	4679#	4504	4510	4621*	4679#	4680	4706*	4746*	5339*	5492#								
LINBIT	024616	4461	4679#	4746	5475#															
LINE	025144	3034	3038	3046	3050	3058	3066*	3080	3084	3092	3096	3104	3112#	3125						
		3127	3159	3161	3195	3213	3241	3259	3286	3304	3331	3349	3388#	3401						
		3637	3641	3660	3720	3733	3748	3752	3761	3768	3777	3784	3992	4007						
		4012	4027	4032	4046	4061	4084	4087	4106	4126	4148	4151	4170	4191						
		4210	4213	4220	4228	4247	4268	4271	4285	4294	4315	4329	4343	4360						

SBOOAT	001126	2259#	2596							
SCDM1	001310	2259#								
SCDM2	001312	2259#								
SCHARC	021616	4980#								
SCHTAG	001100	2259#	2596							
SCHI	000010	2259#								
SCH2	000020	2259#								
SCH3	000010	2259#								
SCH4	000010	2259#								
SCPUOP	001256	2259#								
SCRLF	001225	2259#	4986	4987	4990	4992	4993			
SDBLK	021334	4980#								
SDDMD	001314	2259#								
SDDM1	001316	2259#								
SDDM10	001340	2259#								
SDDM11	001340	2259#								
SDDM12	001344	2259#								
SDDM13	001346	2259#								
SDDM14	001350	2259#								
SDDM15	001350	2259#								
SDDM16	001350	2259#								
SDDM17	001350	2259#								
SDDM18	001350	2259#								
SDDM19	001350	2259#								
SDDM20	001350	2259#								
SDDM21	001350	2259#								
SDDM22	001350	2259#								
SDDM23	001350	2259#								
SDDM24	001350	2259#								
SDDM25	001350	2259#								
SDDM26	001350	2259#								
SDDM27	001350	2259#								
SDDM28	001350	2259#								
SDDM29	001350	2259#								
SDDM30	001350	2259#								
SDDM31	001350	2259#								
SDDM32	001350	2259#								
SDDM33	001350	2259#								
SDDM34	001350	2259#								
SDDM35	001350	2259#								
SDDM36	001350	2259#								
SDDM37	001350	2259#								
SDDM38	001350	2259#								
SDDM39	001350	2259#								
SDDM40	001350	2259#								
SDDM41	001350	2259#								
SDDM42	001350	2259#								
SDDM43	001350	2259#								
SDDM44	001350	2259#								
SDDM45	001350	2259#								
SDDM46	001350	2259#								
SDDM47	001350	2259#								
SDDM48	001350	2259#								
SDDM49	001350	2259#								
SDDM50	001350	2259#								
SDEVCT	001240	2259#								
SDEVN	001306	2259#								
SDOAGN	020056	4984#								
SOTBL	021324	4989#								
SENDAD	020046	2251#	4984#							
SENDCT	020014	4984#								
SENDMG	020062	4984#								
SENULL	020077	4984#								
SENV	001250	2259#	4986	4990	4991					
SENVH	001251	2259#	2596	4990	4991					
SEOP	017760	4973#	4979	4984#						
SEOPCT	020006	4984#								
SERFLG	001103	2259#	4985#	4986#						
SERMAX	001115	2259#	2596#	4985#						
SERROR	020370	2259#	4986#							
SERRPC	001116	2259#	4986#	4987	5626	5633	5722	5738	5753	5792
SERRTB	001354	2259#	4987							
SERTY	020536	4986	4987#							
SERTTL	001112	2259#	4986#							
SESCAP	001222	2259#	2596#	4985#	4986					
SETABL	001250	2259#								
SETEND	001354	2259#								
SFATAL	001232	2259#	4991#							
SFFLG	022066	4991#								
SFILLC	001154	2259#	4990							
SFILLS	001153	2259#	4990							

L07

MAINDEC-11-DZDMM-A
DZDMM.P11

CROSS REFERENCE TABLE

MACY11 27(663) 12-DEC-75 08:41 PAGE 94-34

SEQ 0295

.SERRT	22408	4987
.SPOWE	22408	4995
.SROOC	22418	4993
.SREAO	22418	4992
.SSCOP	22408	4985
.STRAP	22408	4994
.STYPD	22408	4989
.STYPE	22408	4990
.STYPO	22408	4988

NOV

NOVB

4855	4858	4859	4871	4874	4885	4889	4894	4895	4904	4909	4910	4920	4933	4936
4943	4960	4963	4964	4984	4986	4990	4991	5016	5028	5195	5203	5275	5279	
2600	2601	2602	2619	2620	2621	2622	2623	2634	2635	2636	2644	2647		
2700	2701	2702	2703	2704	2705	2706	2707	2708	2709	2710	2711	2712	2713	2714
2715	2716	2717	2718	2719	2720	2721	2722	2723	2724	2725	2726	2727	2728	2729
2730	2731	2732	2733	2734	2735	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	2795	2796	2797	2798	2799	2800	2801	2802	2803	2804
2805	2806	2807	2808	2809	2810	2811	2812	2813	2814	2815	2816	2817	2818	2819
2820	2821	2822	2823	2824	2825	2826	2827	2828	2829	2830	2831	2832	2833	2834
2835	2836	2837	2838	2839	2840	2841	2842	2843	2844	2845	2846	2847	2848	2849
2850	2851	2852	2853	2854	2855	2856	2857	2858	2859	2860	2861	2862	2863	2864
2865	2866	2867	2868	2869	2870	2871	2872	2873	2874	2875	2876	2877	2878	2879
2880	2881	2882	2883	2884	2885	2886	2887	2888	2889	2890	2891	2892	2893	2894
2895	2896	2897	2898	2899	2900	2901	2902	2903	2904	2905	2906	2907	2908	2909
2910	2911	2912	2913	2914	2915	2916	2917	2918	2919	2920	2921	2922	2923	2924
2925	2926	2927	2928	2929	2930	2931	2932	2933	2934	2935	2936	2937	2938	2939
2940	2941	2942	2943	2944	2945	2946	2947	2948	2949	2950	2951	2952	2953	2954
2955	2956	2957	2958	2959	2960	2961	2962	2963	2964	2965	2966	2967	2968	2969
2970	2971	2972	2973	2974	2975	2976	2977	2978	2979	2980	2981	2982	2983	2984
2985	2986	2987	2988	2989	2990	2991	2992	2993	2994	2995	2996	2997	2998	2999
3000	3001	3002	3003	3004	3005	3006	3007	3008	3009	3010	3011	3012	3013	3014
3015	3016	3017	3018	3019	3020	3021	3022	3023	3024	3025	3026	3027	3028	3029
3030	3031	3032	3033	3034	3035	3036	3037	3038	3039	3040	3041	3042	3043	3044
3045	3046	3047	3048	3049	3050	3051	3052	3053	3054	3055	3056	3057	3058	3059
3060	3061	3062	3063	3064	3065	3066	3067	3068	3069	3070	3071	3072	3073	3074
3075	3076	3077	3078	3079	3080	3081	3082	3083	3084	3085	3086	3087	3088	3089
3090	3091	3092	3093	3094	3095	3096	3097	3098	3099	3100	3101	3102	3103	3104

	4868	4872	4917	4921	4984	4985	4986	4987	4988	4989	4990	4991	4992	4993	4994
.IFT	4995	4998	5000	5375	5377	5617	5619								
.IFTF	4985	4986	4992	4993											
.IIF	2244	2245	2246	2247	2259	2596	4984	4985	4986	4987	4990	4992	4993	4994	
.IRP	2596	2647	2671	2698	2732	2755	2805	2806	2804	2838	2872	2909	2947	2985	3025
	3152	3186	3232	3278	3323	3368	3420	3464	3504	3544	3584	3624	3725	3797	3853
	3886	3929	3985	4054	4118	4182	4238	4352	4405	4515	4594	4689	4750	4868	4917
.LIST	4985	4989	4991	4992	4993	5097	5111	2596	2647	2671	2698	2732	2755	2805	2839
	2233	2236	2244	2247	2259	2259	2596	2647	2671	2698	2732	2755	2805	2839	
	2873	2910	2948	2986	3026	3072	3118	3152	3186	3232	3278	3323	3368	3420	3464
	3504	3544	3584	3624	3725	3797	3886	3929	3985	4053	4118	4182	4238	4353	4405
.MACRO	4405	4515	4594	4689	4750	4868	4917	4985	4986	4989	4992	4994	4995	5626	5633
	814	859	902	946	988	1052	1100	1154	1199	1244	1287	1381	1432	1493	1537
	1572	1622	1675	1732	1783	1817	1883	1917	1950	2015	2047	2078	2136	2184	2245
.MCALL	2253	2259	2240	2241	2242	2258	2259	2596	2647	2671	2698	2732	2755	2805	2839
.MLIST	2238	2239	2240	2241	2242	2258	2259	2596	2647	2671	2698	2732	2755	2805	2839
	2873	2910	2948	2986	3026	3072	3118	3152	3186	3232	3278	3323	3368	3420	3464
	3504	3544	3584	3624	3725	3797	3886	3929	3985	4053	4118	4182	4238	4353	4405
.PAGE	4405	4515	4594	4689	4750	4868	4917	4985	4986	4989	4992	4994	4995	5626	5633
	2259	2596	2647	2671	2698	2732	2755	2804	2838	2872	2909	2947	2985	3025	
	3071	3117	3151	3185	3231	3277	3322	3367	3419	3463	3503	3543	3583	3623	3724
	3796	3852	3885	3928	3984	4052	4117	4181	4237	4352	4404	4514	4593	4688	4749
.REM	4867	4916	4971	4997	5616	2806	2840	2874	2911	2949	2987	3027	3073	3119	3153
	2648	2672	2699	2733	2756	2806	2840	2874	2911	2949	2987	3027	3073	3119	3153
	3187	3233	3279	3324	3369	3421	3465	3505	3545	3585	3625	3726	3798	3854	3887
	3930	3986	4054	4119	4183	4239	4354	4406	4516	4595	4690	4751	4869	4918	
.REPT	2247	2259	2251	2252	2258	2259	2647	2671	2698	2732	2755	2805	2839	2873	2910
.SBTTL	2245	2247	2251	2252	2258	2259	2647	2671	2698	2732	2755	2805	2839	2873	2910
	2948	2986	3026	3072	3118	3152	3186	3232	3278	3323	3368	3420	3464	3504	3544
	3584	3624	3725	3797	3853	3886	3929	3985	4053	4118	4182	4238	4353	4405	4515
	4594	4689	4750	4868	4917	4984	4985	4986	4987	4988	4989	4990	4991	4992	4993
.TITLE	4994	4995													
.WORD	2244	2251	2252	2259	4984	4987	4988	4990	4991	4993	4995	5626	5633	5722	5738
	2247	5792													
	5753														

ERRORS DETECTED: 0
 *DZDHMA, DZDHMA/CRF=DHMMAC, DZDHMA
 RUN-TIME: 40 30 4 SECONDS
 CORE USED: 37K

ND-11-DZDMM-A DH11 DIAGNOSTIC
TEST 45 BASIC PARITY LOGIC TEST - ALL LINES - ODD PARITY

#TST45 *

SET UP ERROR *
LOOP RETURN *

#A-----> I
*** 1S I SELINE(60)

**GO SELECT A LINE NO. **
** TO TEST **
** *****

/ TESTED ALL SEL- \ YES
ECTED LINES ? \-----> I

#D-----> I NO
*** 2S I

#SCOPE(52)*

#ACTIVATE DH11 TO XFER*
ONE CHAR WITH "ODD"
PARITY SELECTED *

INIT TIMERS *
"A" AND "B" *

*** YES
#D<----- / LOOP ON ERROR? \
*** SRO9=1

I NO

#A*

/ CHAR AVAIL SET \ YES ***
YET ? \-----> #C*

I NO TIMEIT(59)

** CALL TIMER **
** *****

/ CHAR AVAILABLE \
TIMEOUT ? \-----> I

I YES

#B*

#B*

SAVE ERROR PSW SET *
UP WAS, S/B DATA AND *
REGADR *

I SUER2A(55)

**GO SETUP INFORMATION **
** TO BE PRINTED **
** *****

I SUNUM(56)

** GO SET UP LINE NO. **
** IN ERROR MESSAGE **
** *****

I SERROR(53)

** GO REPORT DATA **
** AVAILABLE TIMEOUT **
** EM22,DH2,DT2,0 **

#C*

4S

/ CORRECT DATA \ YES ***
RECEIVED ? \-----> #A*

I NO

SAVE ERROR PSW *
SET UP REGADR *

I SUER2A(55)

**GO SETUP INFORMATION **
** TO BE PRINTED **
** *****

I SUNUM(56)

** GO SET UP LINE NO. **
** IN ERROR MESSAGE **
** EM33,DH2,DT2,0 **

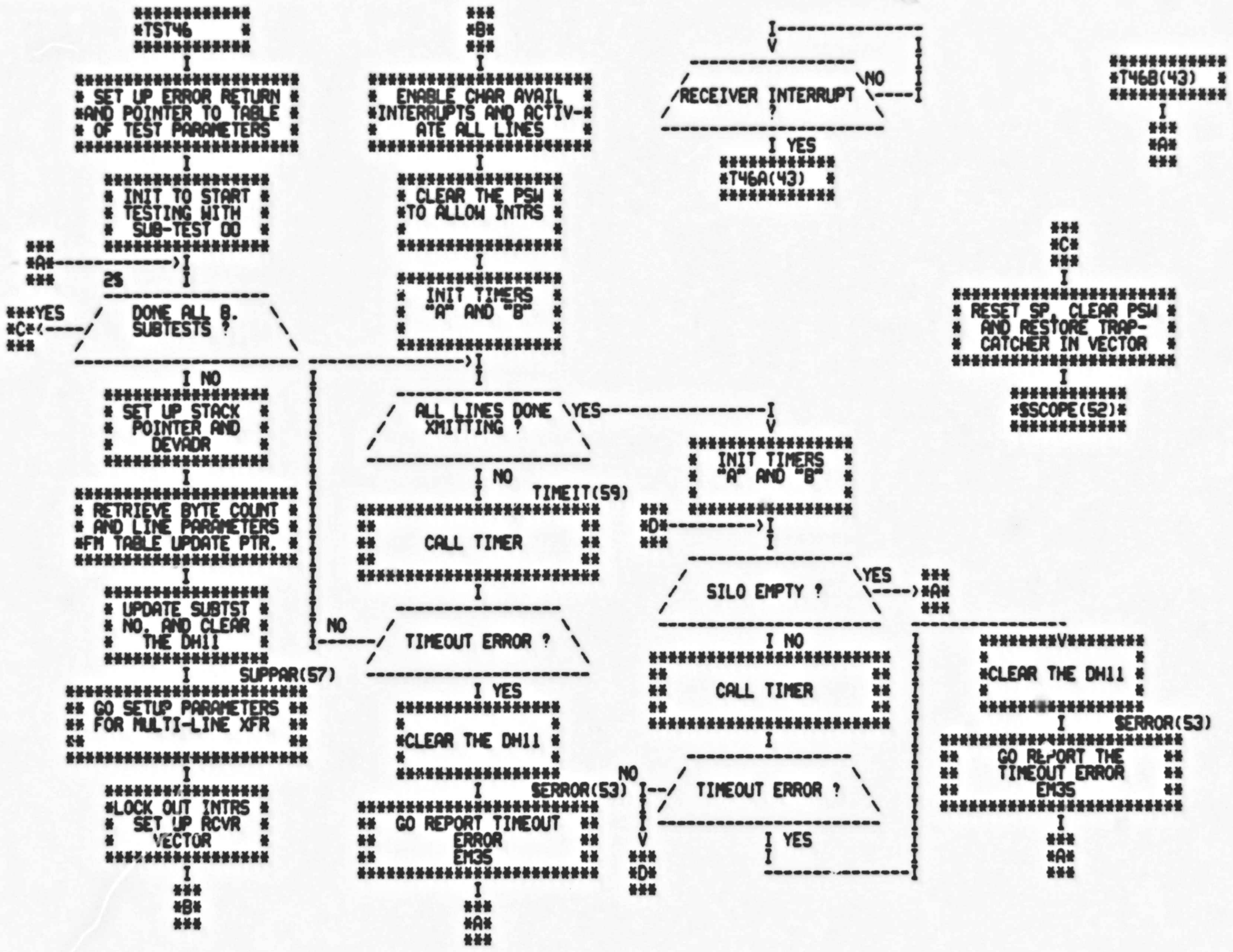
I SERROR(53)

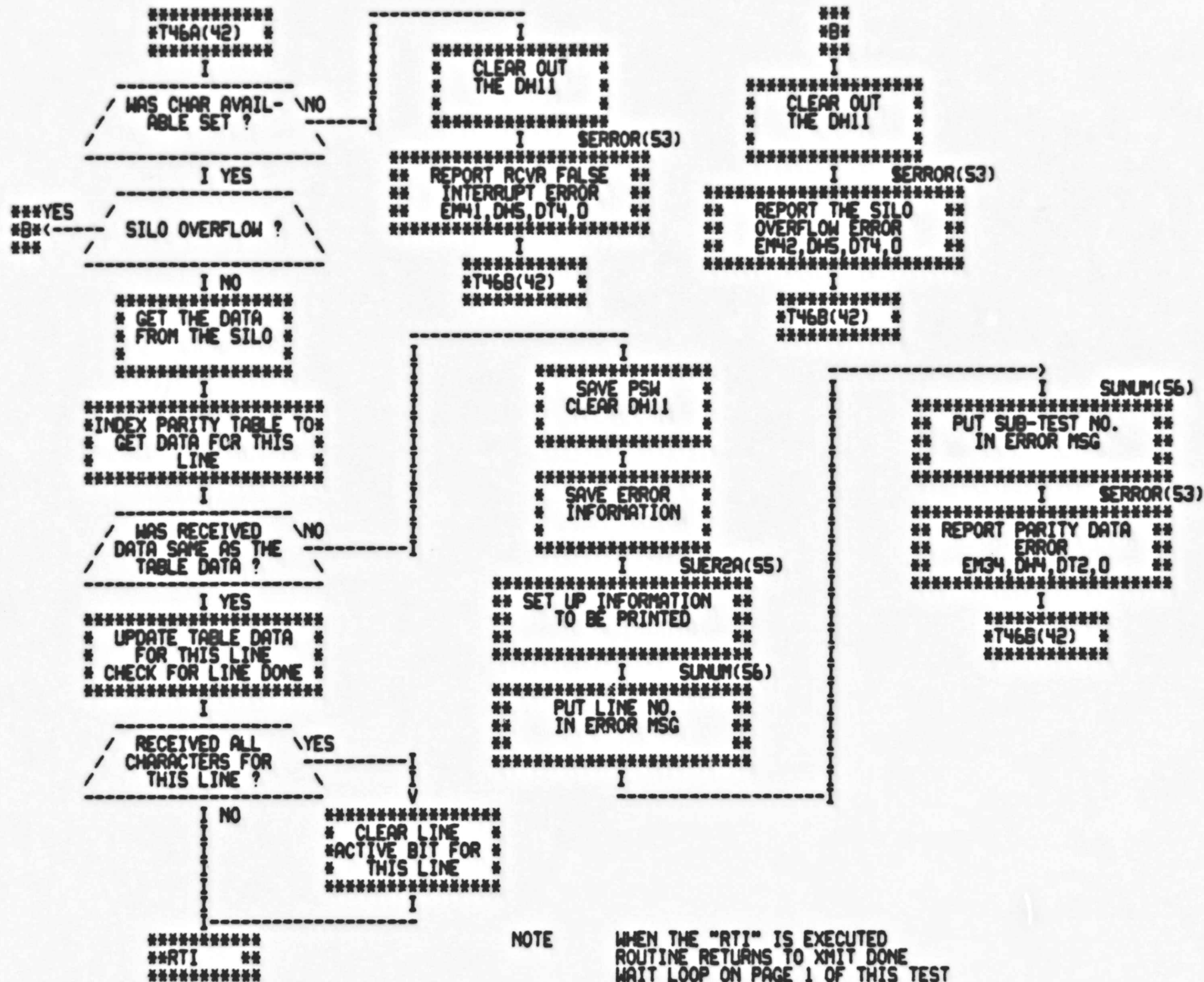
** GO REPORT PARITY **
** DATA ERROR **

*** YES
#D<----- / LOOP ON ERROR ? \
*** SRO9=1

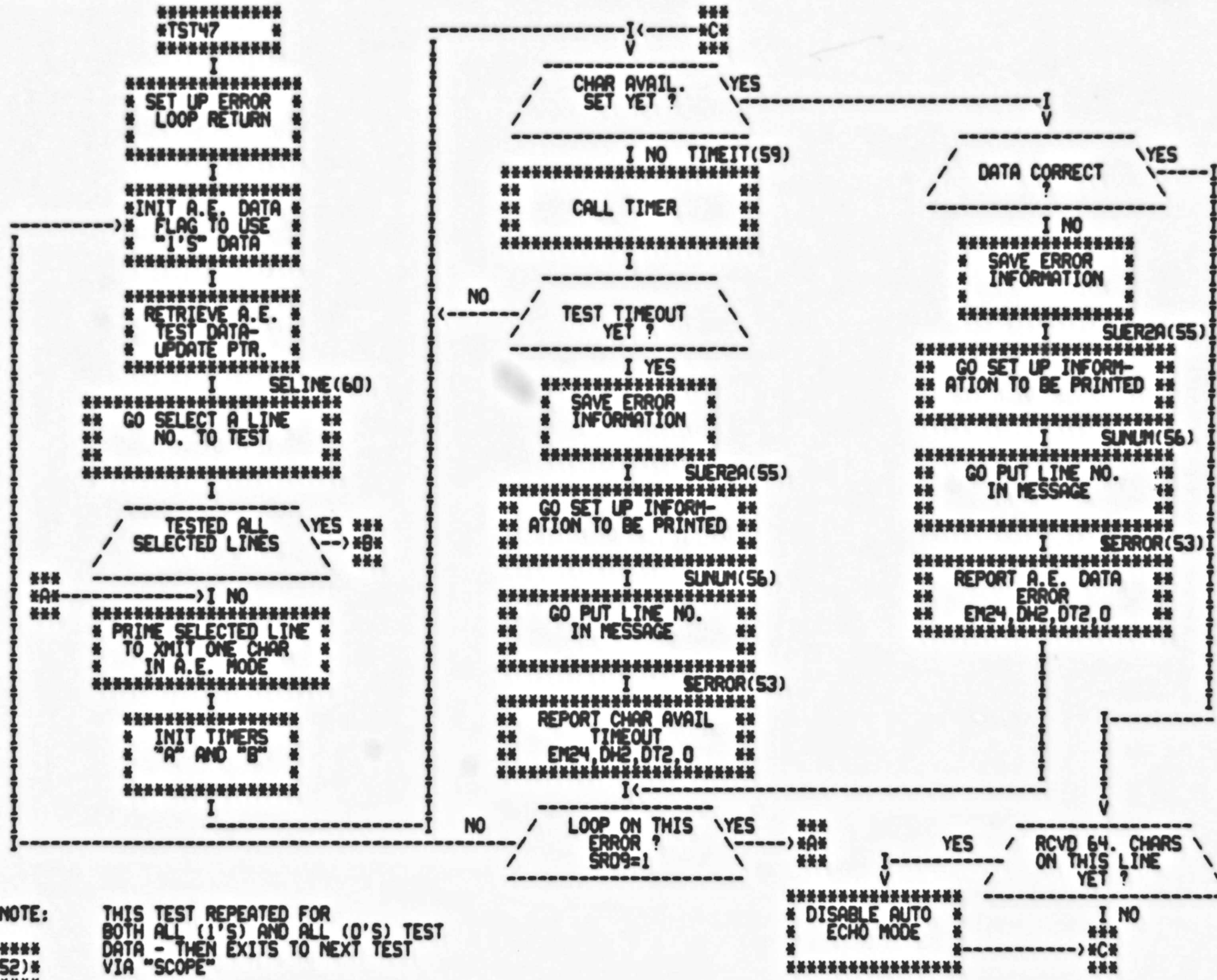
I NO

#A*





NOTE WHEN THE "RTI" IS EXECUTED
ROUTINE RETURNS TO XMIT DONE
WAIT LOOP ON PAGE 1 OF THIS TEST



#B#
*** NOTE:
I

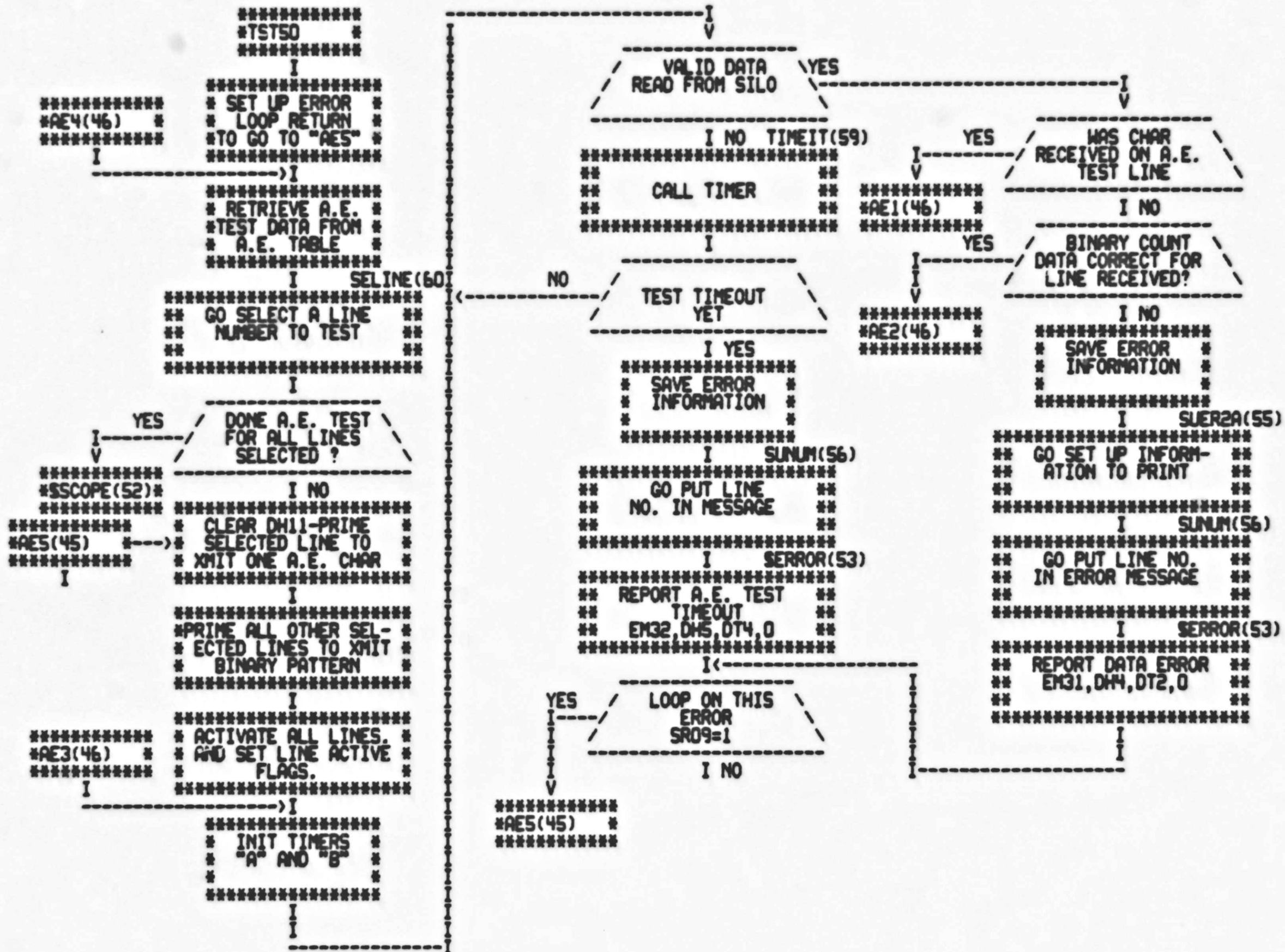
#SCOPE(52)*

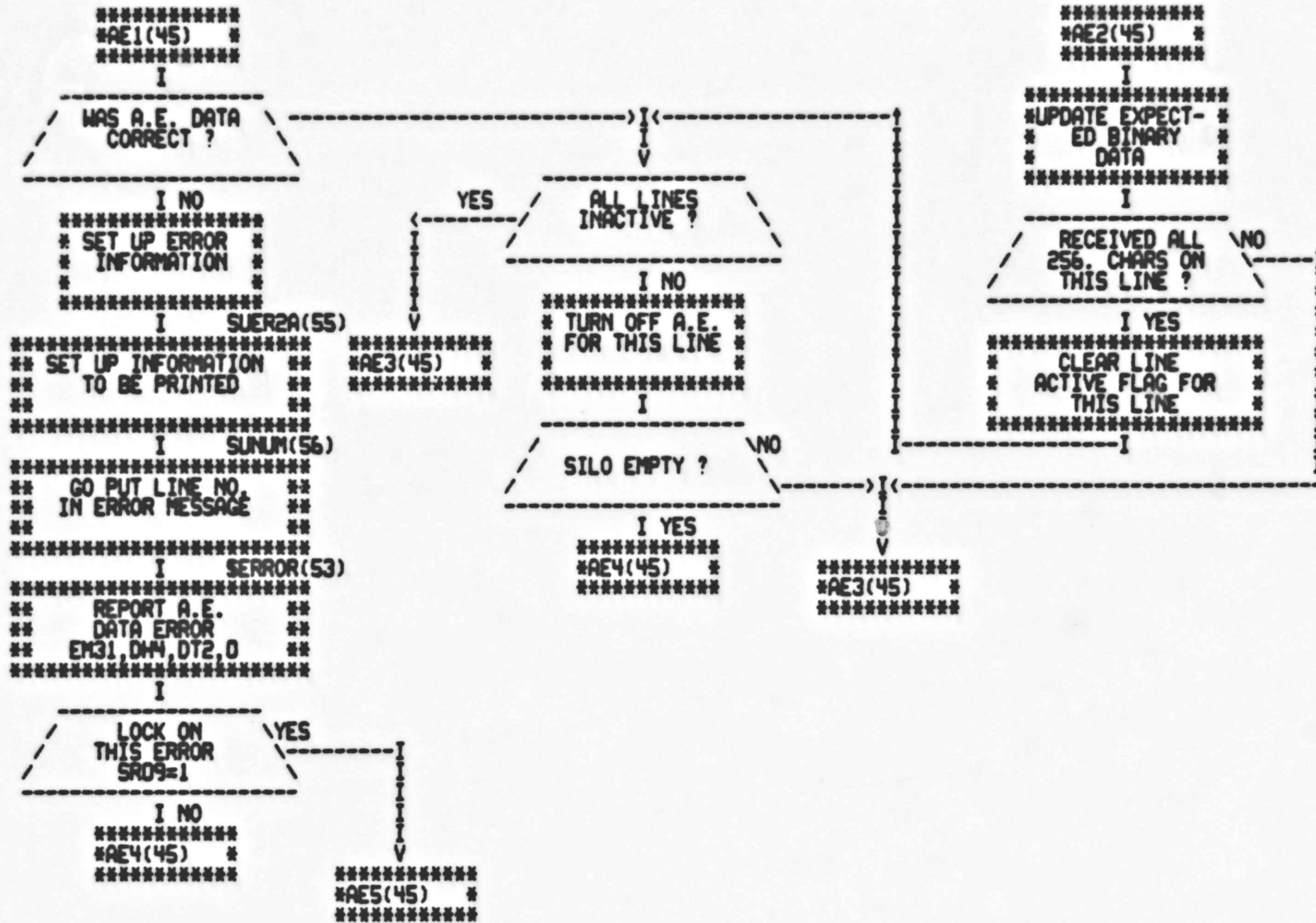
THIS TEST REPEATED FOR
BOTH ALL (1'S) AND ALL (0'S) TEST
DATA - THEN EXITS TO NEXT TEST
VIA "SCOPE"

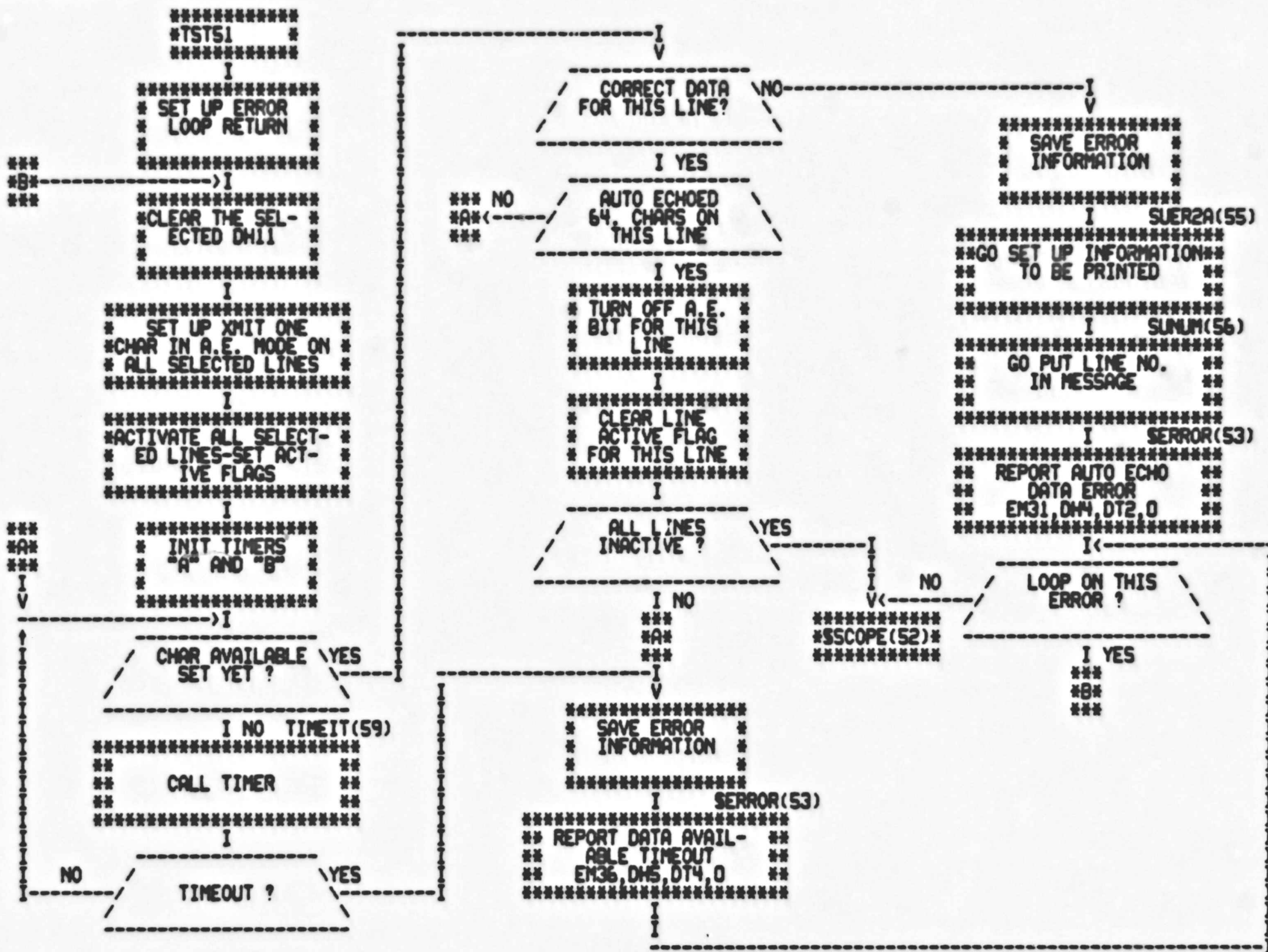
DISABLE AUTO
ECHO MODE

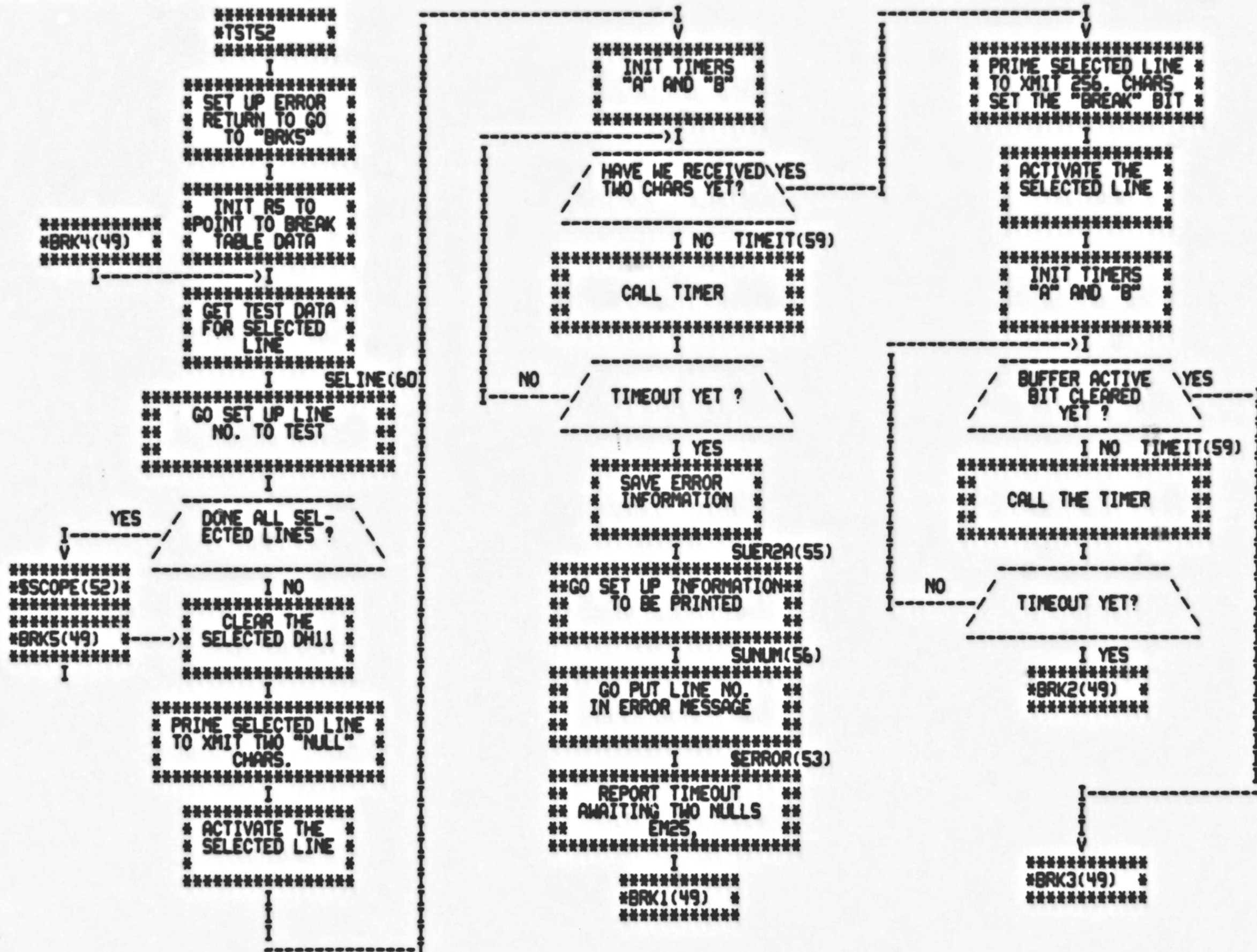
I NO

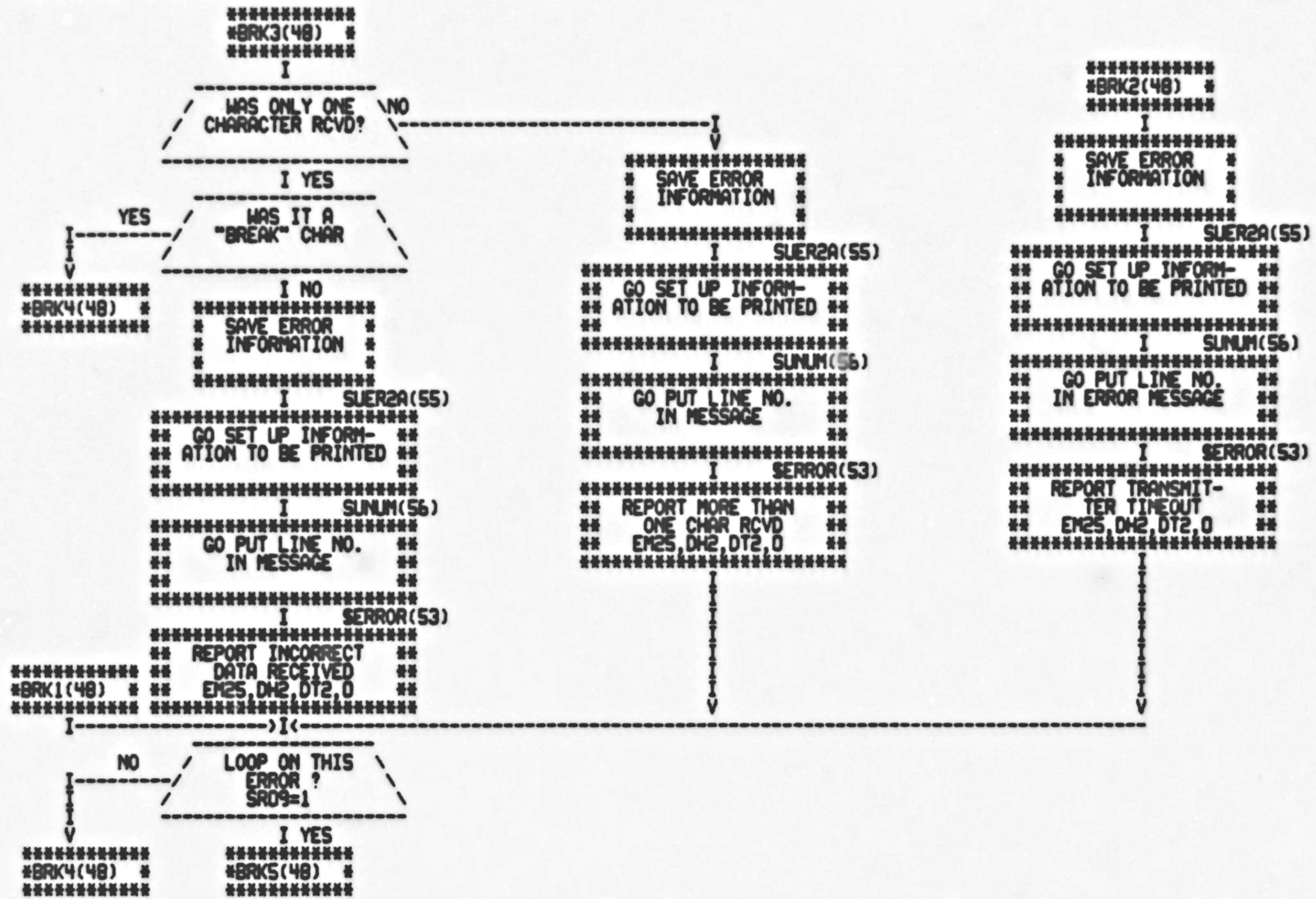
#C#

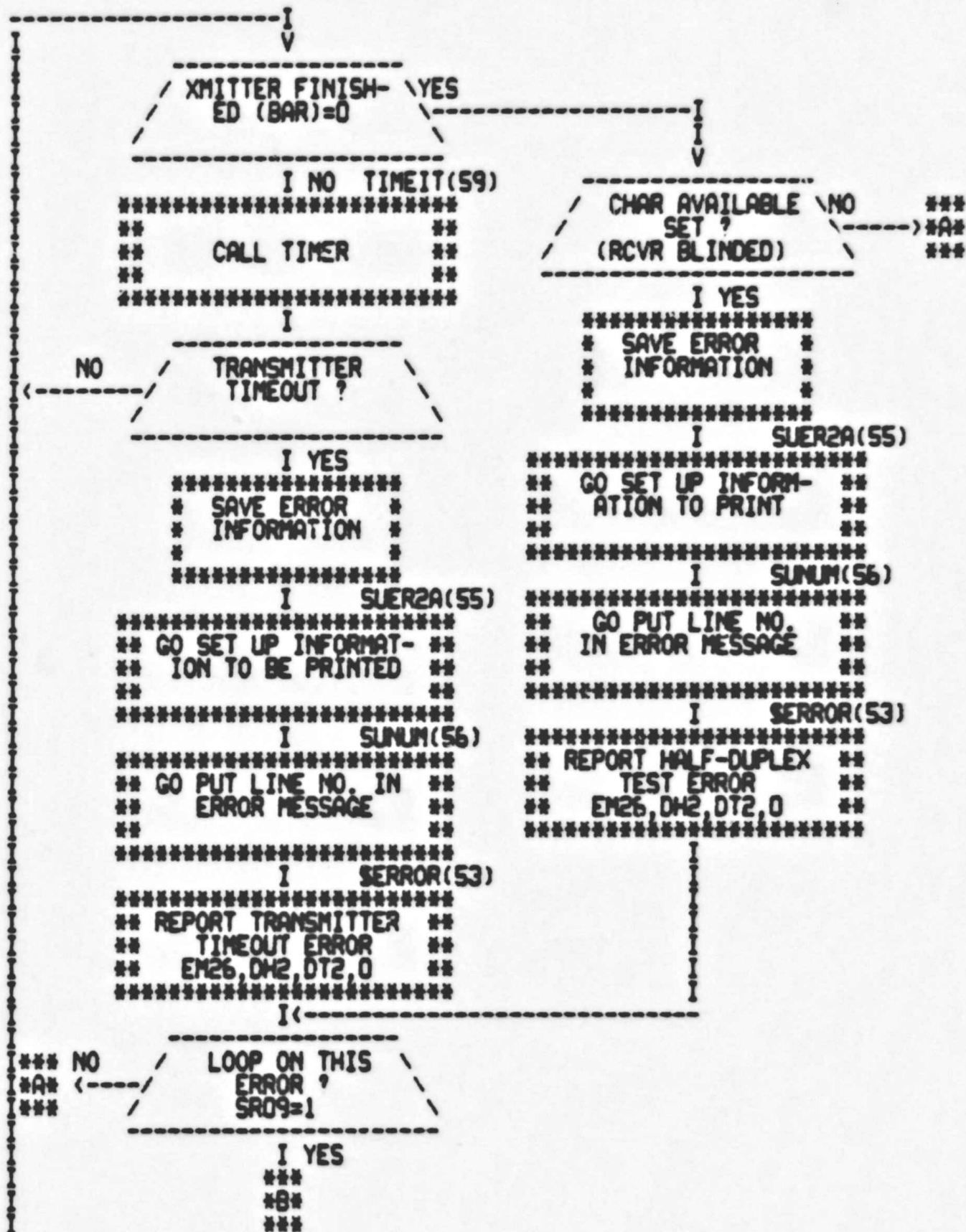
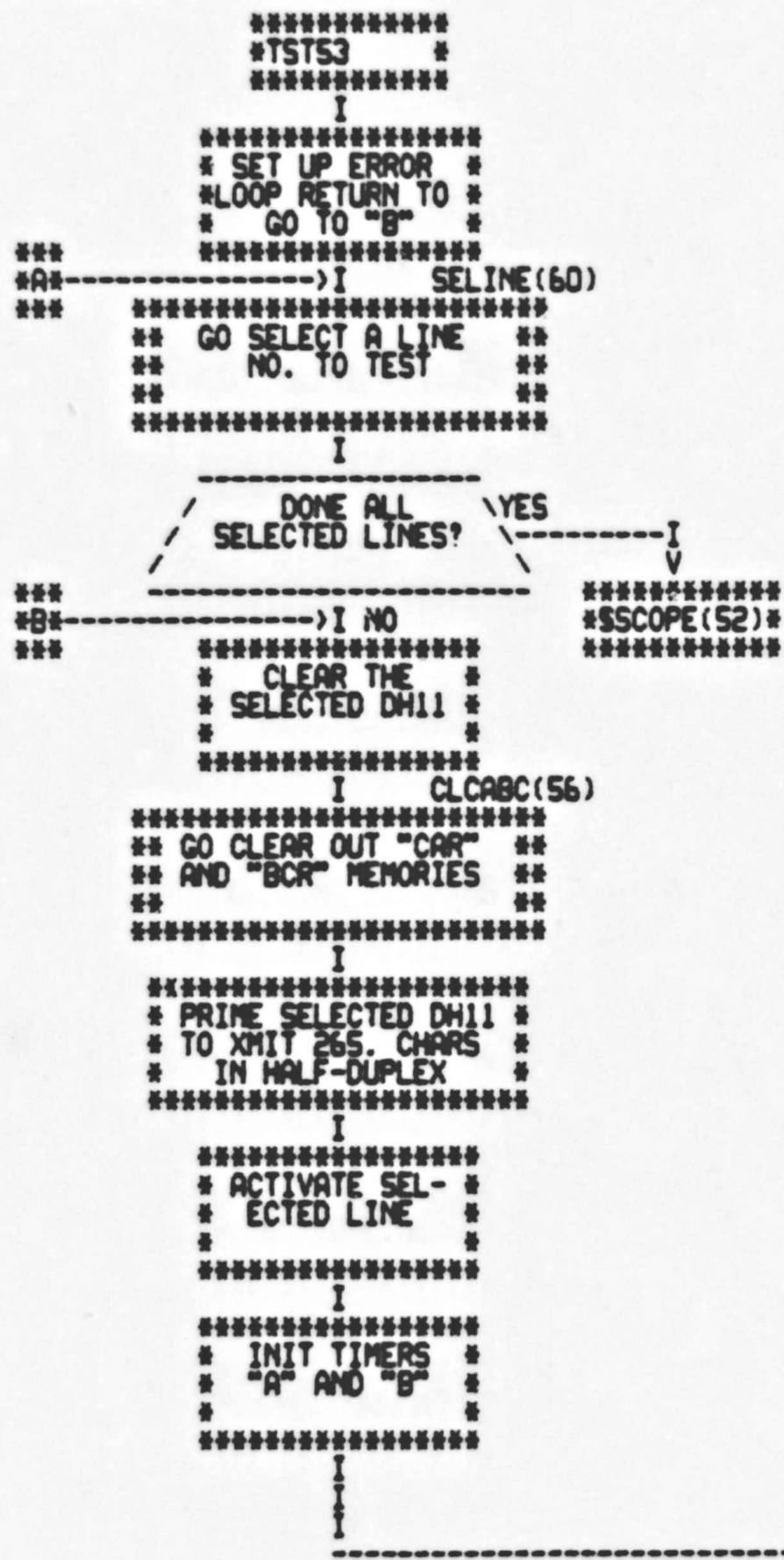


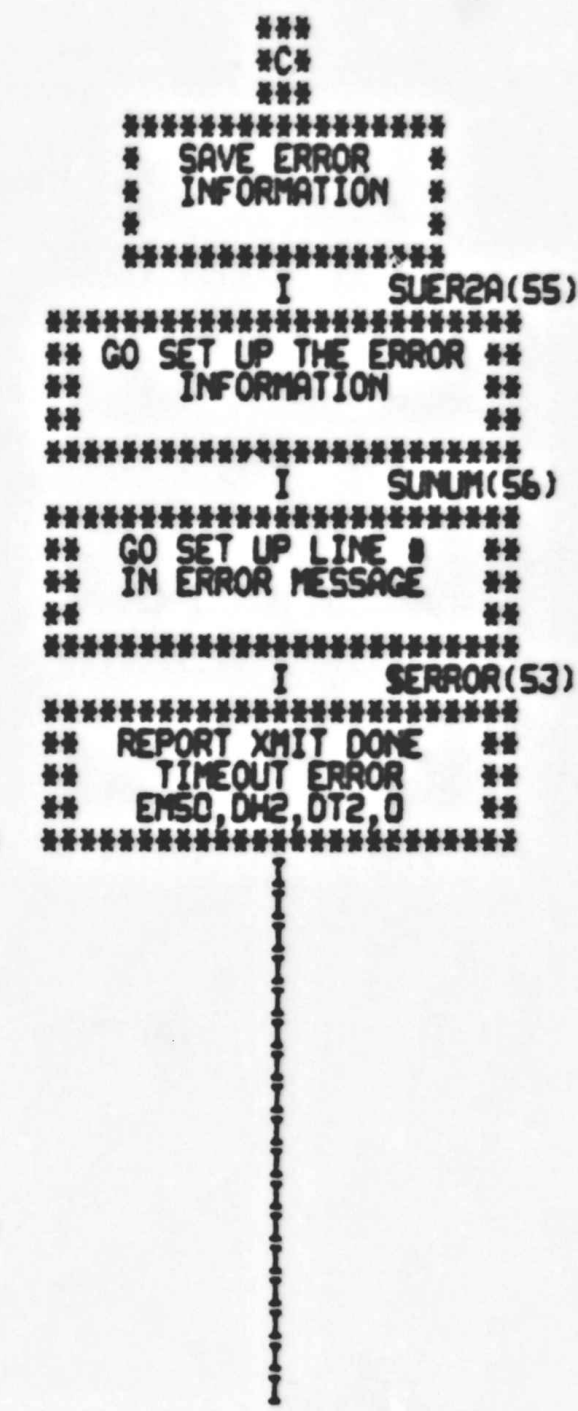
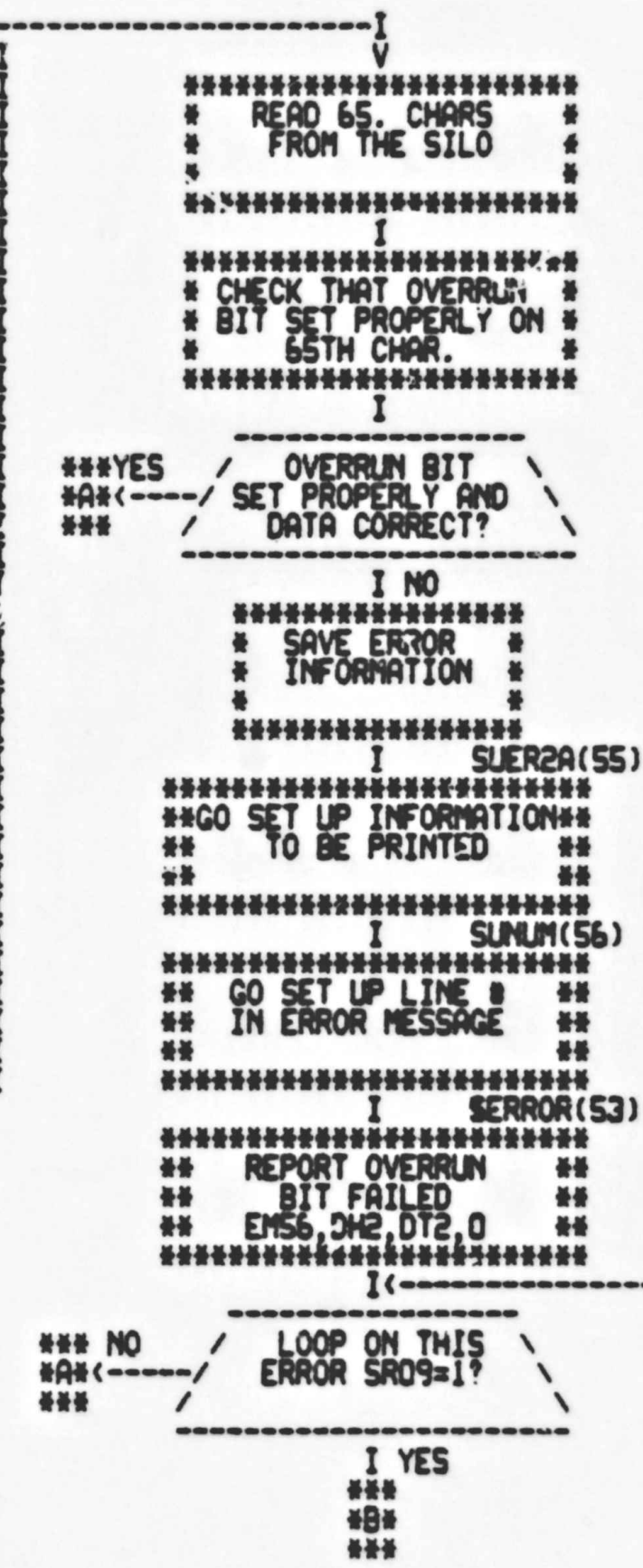
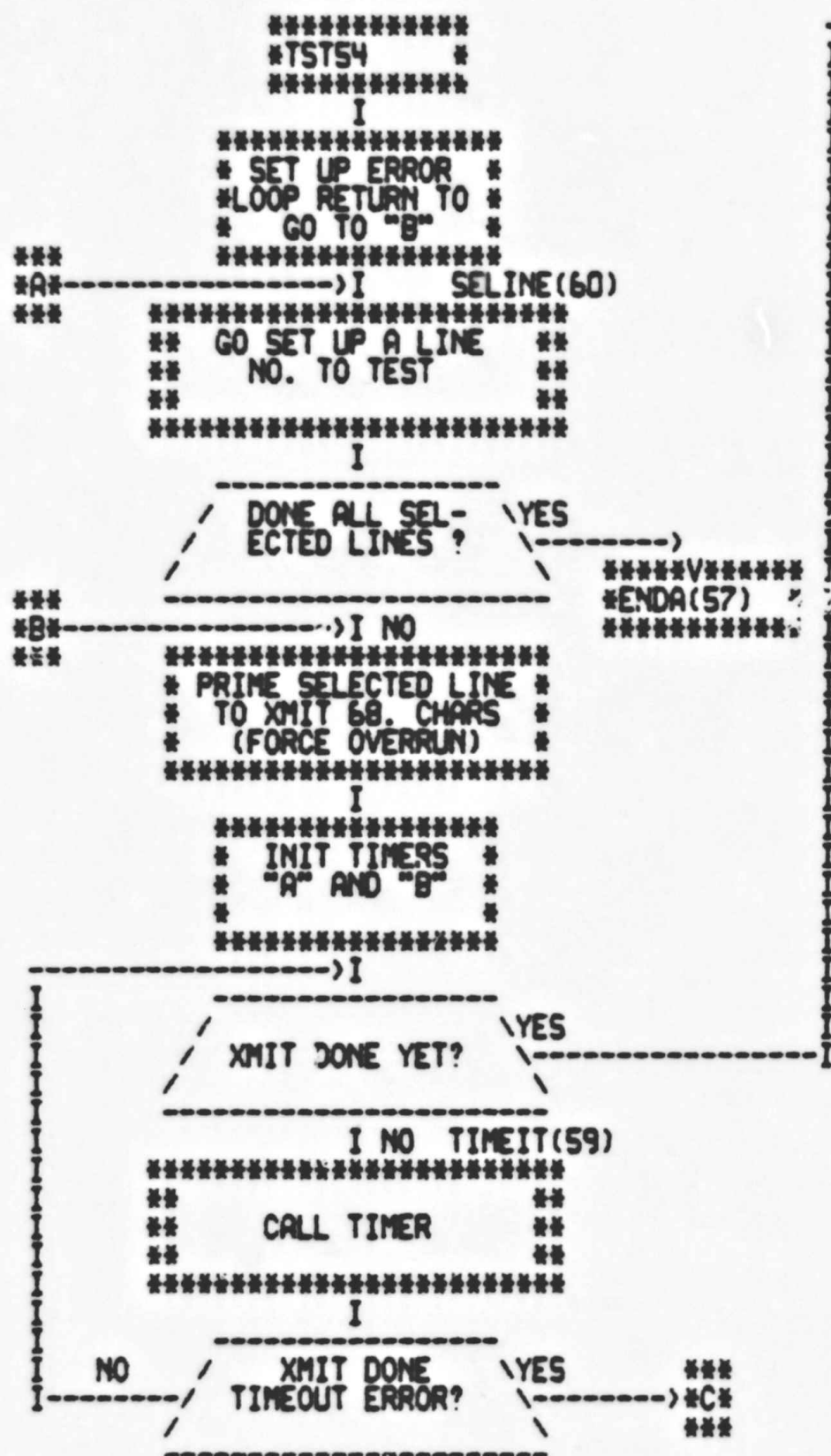


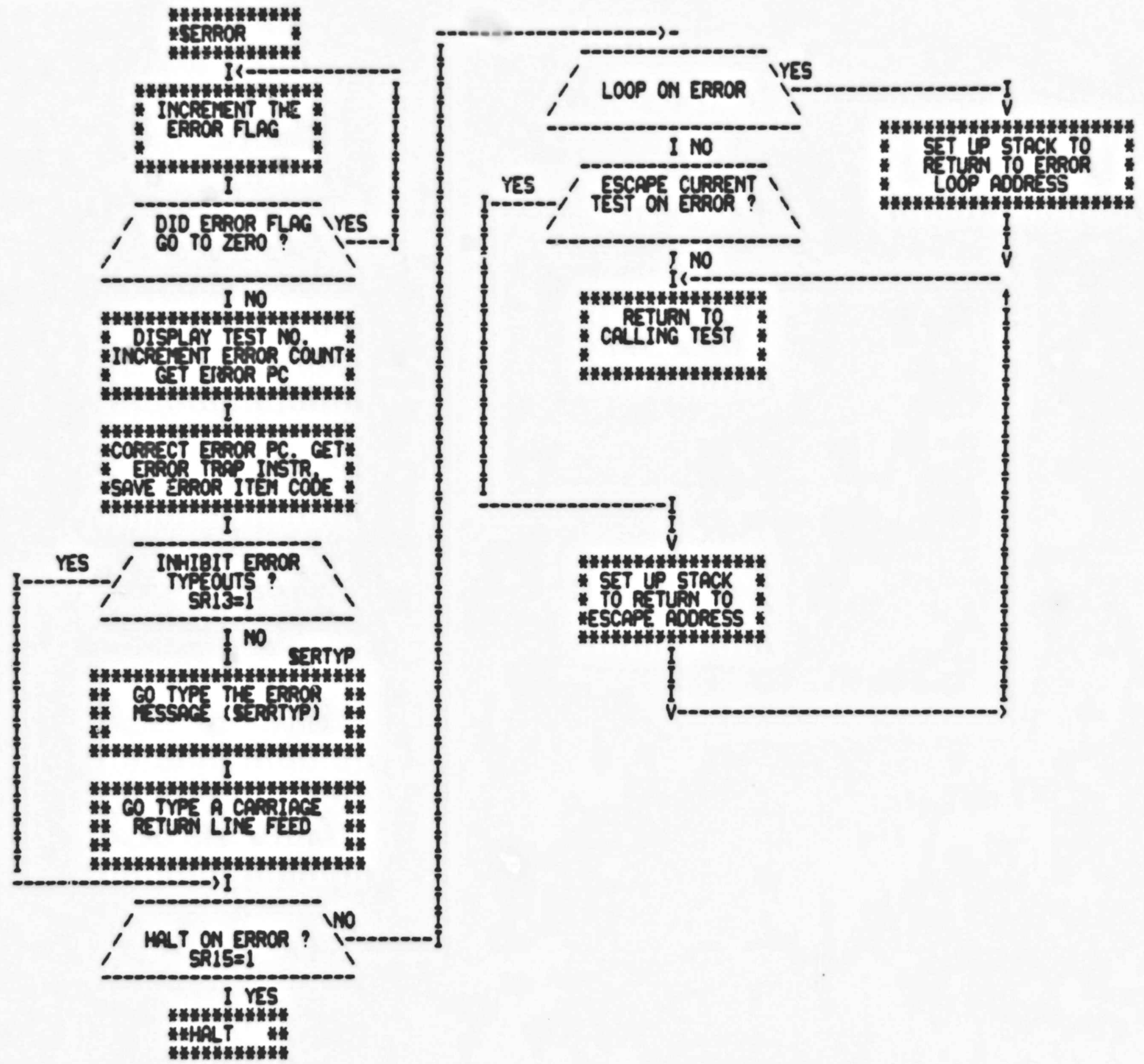












```

*****
#SETALL *
*****
I
*****
* SET UP STMP6 *
* TO A 20(8) TO *
* COUNT 16. LINES*
*****
I
*****
* INIT R2 TO INDEX *
* READ BUFFER TEST *
* DATA TABLE *
*****
I
*****
* INIT R3 TO SET UP *
* VALID DATA BIT + *
* LINE NO. IN TABLE *
*****
I
*****
* INIT R4 TO SET UP *
* FOR INDEXING HIGH *
* BYTE IN TABLE *
*****
I
*****
* INIT "SCR" TO *
* START WITH *
* LINE 00 *
*****
I<-----
*****
* LOAD "CAR", "BCR", *
* AND "LPR" REGS *
* FOR SELECTED LINE *
*****
I
*****
* CLEAR LO BYTE TABLE *
* ENTRY AND LOAD (R3) *
* INTO HI BYTE *
*****
I
***
#A*
***

```

```

***
#A*
***
I
*****
* INCREMENT (SCR) *
* UPDATE R3, R2, AND *
* R4 FOR NEXT ENTRY *
*****
I
*****
* COUNT STMP6 *
* TO INDICATE THE *
* NEXT LINE TO DO *
*****
I
-----
/ HAVE WE SET \
/ UP ALL 16. LINES \
-----
I YES
*****
**RTS **
*****
I
*****
*CHPS1 *
*****
I
*****
* PUSH NEW PSW=000 *
* AND NEW PC=1S *
* ON TO STACK *
*****
I
*****
* DO AN RTI TO *
* LOAD NEW PSW *
* GO TO 1S *
*****
I
*****
* DO RTS TO *
* RETURN TO *
* CALLER *
*****
I
*****
**RTS **
*****

```

```

*****
*CHPS2 *
*****
I
*****
* PUSH PSW=340 *
* AND PC=1S *
* ON THE STACK *
*****
I
*****
* DO AN RTI TO *
* LOAD NEW PSW *
* GO TO 1S *
*****
I
*****
* DO AN RTS *
* TO RETURN TO *
* CALLER *
*****
I
*****
**RTS **
*****

```

```

*****
*SAPS *
*****
I
*****
* PUSH STACK TO *
* SAVE SPACE *
* FOR SAVPSW *
*****
I
*****
* PUSH THE CURRENT *
* CONTENTS OF THE *
* TRAP VECTOR *
*****
I
*****
* SET UP TRAP VECT- *
* OR TO GO TO 1S *
*****
I
*****
* DO A "TRAP" *
* INSTRUCTION *
*
*****
I
*****
* SAVE PSW, PUSH *
* 2S ON STACK *
* DO RTI *
*****
I
*****
* RESTORE TRAP VECTOR *
* POP STACK (SAVPS) *
* INTO STMP0 *
*****
I
*****
**RTS **
*****

```

NOTE: I"CHPS1" "CHPS2", AND "SAPS" ARE CALLED WHENEVER
THE MAINLINE CODE HAS TO CLEAR THE PSW, LOCK OUT INTRs,
AND SAVE THE PSW RESPECTIVELY

THESE ROUTINES ARE REQUIRED FOR LS111 COMPATIBILITY

```

*****
#LDTBF1 #
*****
I
*****
# SET R1 TO #
# POINT TO XMIT #
# BUFFER #
*****

```

```

*****
# INIT R2=0 #
# AS A CHAR #
# GENERATOR #
*****

```

```

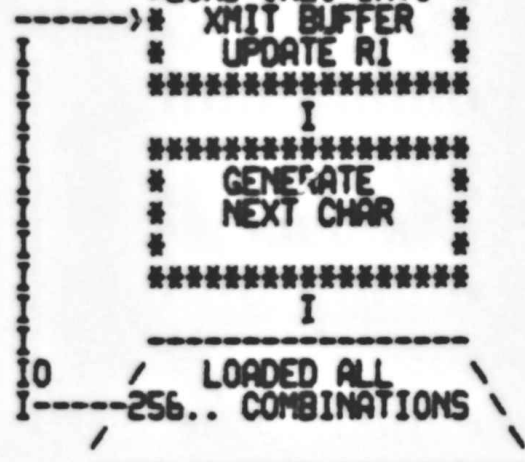
*****
#LOAD [R2] INTO #
# XMIT BUFFER #
# UPDATE R1 #
*****

```

```

*****
# GENERATE #
# NEXT CHAR #
*****

```



```

I YES
*****
**RTS **
*****

```

```

*****
#SUER1 #
*****
I SAPS(54)
*****
** GO SAVE PSW **
** IN STMP0 **
*****

```

```

*****
# SAVE TEST NO. #
# IN R0 #
*****

```

```

*****
# SAVE R0,R1,R2,R6 #
# IN SREG <0,1,2,6> #
*****

```

```

*****
# CORRECT SREG6 #
*****

```

```

I
*****
**RTS **
*****

```

```

*****
#SUER2 #
*****
I SAPS(54)
*****
** GO SAVE PSW **
** IN STMP0 **
*****

```

```

*****
# SAVE TEST NO. #
# IN R0 #
*****

```

```

*****
#SAVE R0,R1,R2,R3,R4, #
# AND R6 IN #
# SREG <0,1,2,3,4,6> #
*****

```

```

*****
# CORRECT SREG6 #
*****

```

```

I
*****
**RTS **
*****

```

```

*****
#SUER2A #
*****
I

```

```

*****
#SUER3 #
*****
I
*****
# SAVE R0,R1,R2 #
# IN SREG<0,1,2> #
*****

```

```

*****
**RTS **
*****

```

```

*****
#SUER4 #
*****
I
*****
# SAVE LINE NO. #
# IN STMP0 #
*****

```

```

*****
# SAVE R0,R1,R2,R3,R4 #
# IN SREG<0,1,2,3,4> #
*****

```

```

I
*****
**RTS **
*****

```



```

*****
#SUPPAR *
*****
I
*****
# INIT LINE COUNTER *
# AND "SCR" TO *
# START AT LINE 00 *
*****
I
*****
# INIT R3 AND R4 *
# TO LOAD TEST DATA *
# TABLE *
*****
I
*****
# LOAD "CAR", "BCR", *
# AND "LPR" FOR *
# SELECTED LINE *
*****
I
*****
# GENERATE NEW *
# LINE NO. *
*****
I
-----
/ \
DONE SET UP FOR 16. LINES \ / NO
-----
I YES
*****
# SET UP MULTI-LINE *
# PARITY TEST DATA *
# TABLE *
*****
I
*****
#RTS **
*****

```

```

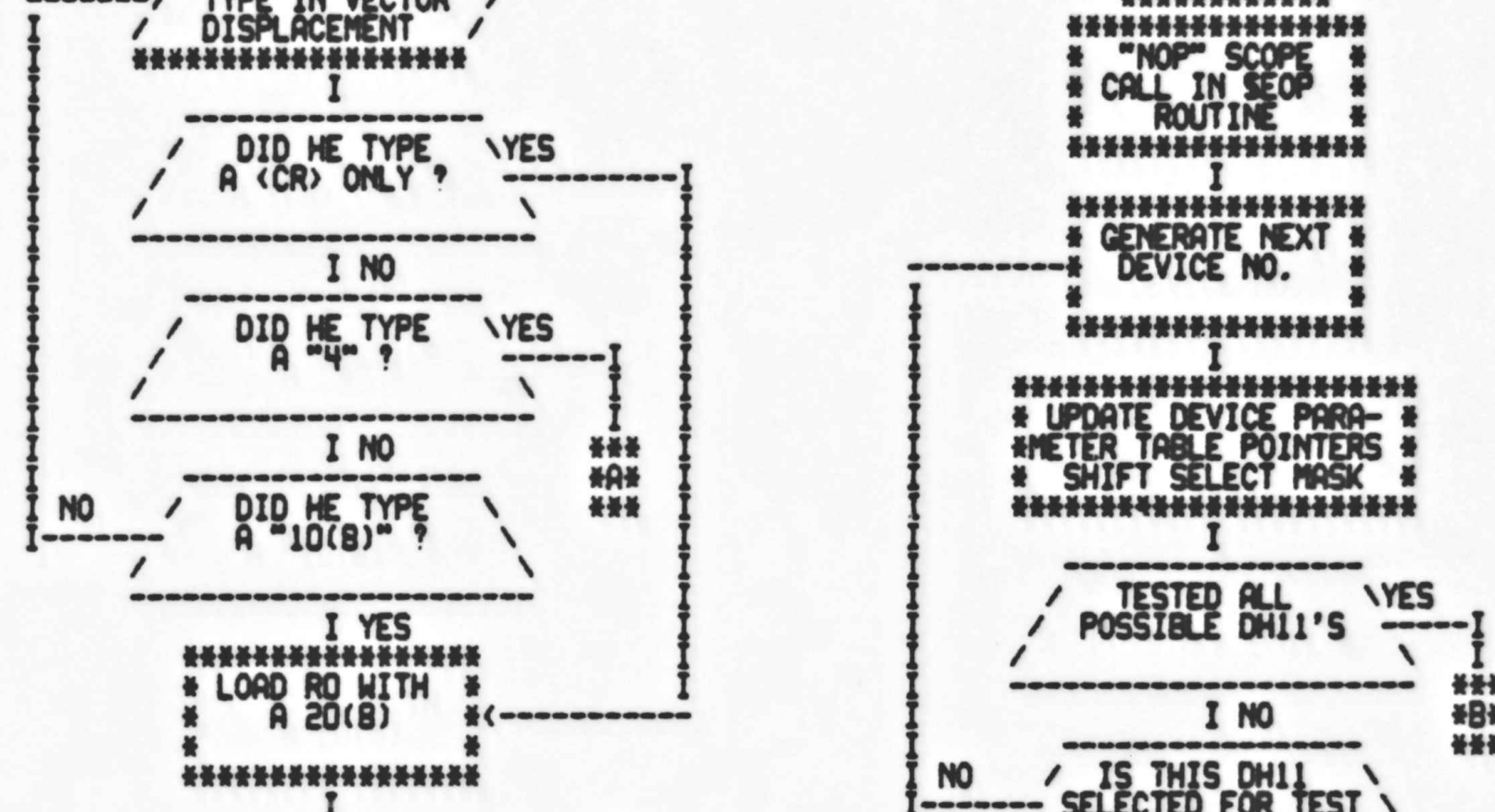
*****
#INPARA *
*****
I
*****
/ \
ASK USER TO TYPE IN VECTOR \ /
DISPLACEMENT
*****
I
-----
/ \
DID HE TYPE A <CR> ONLY ? \ / YES
-----
I NO
-----
/ \
DID HE TYPE A "4" ? \ / YES
-----
I NO
-----
/ \
DID HE TYPE A "10(B)" ? \ / YES
-----
I YES
*****
# LOAD RO WITH *
# A 20(B) *
*****
I
-----
I
-----
I
-----
/ \
LOAD RO WITH A 10(B) \ /
-----
I
-----
I
*****
#RTS **
*****

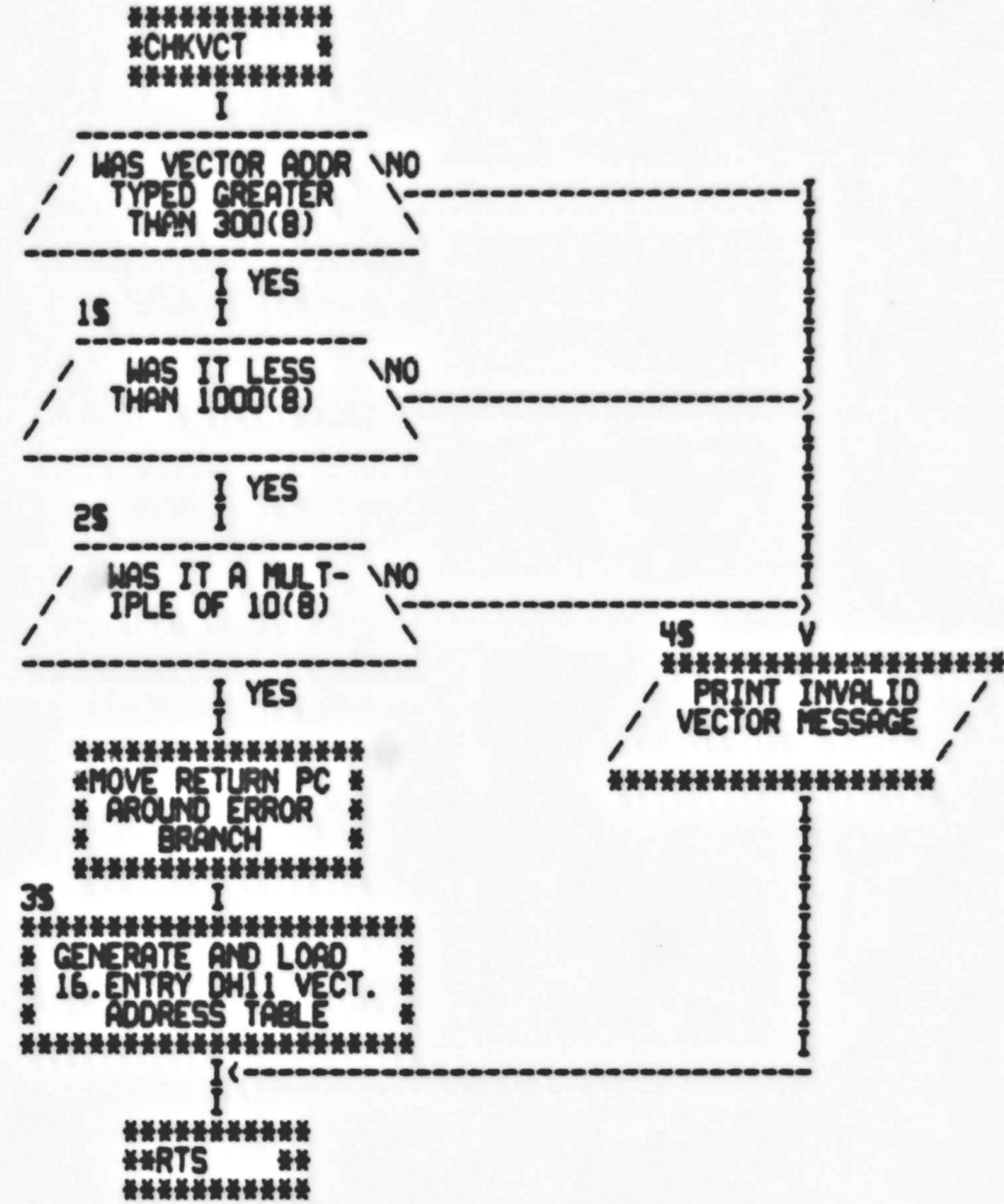
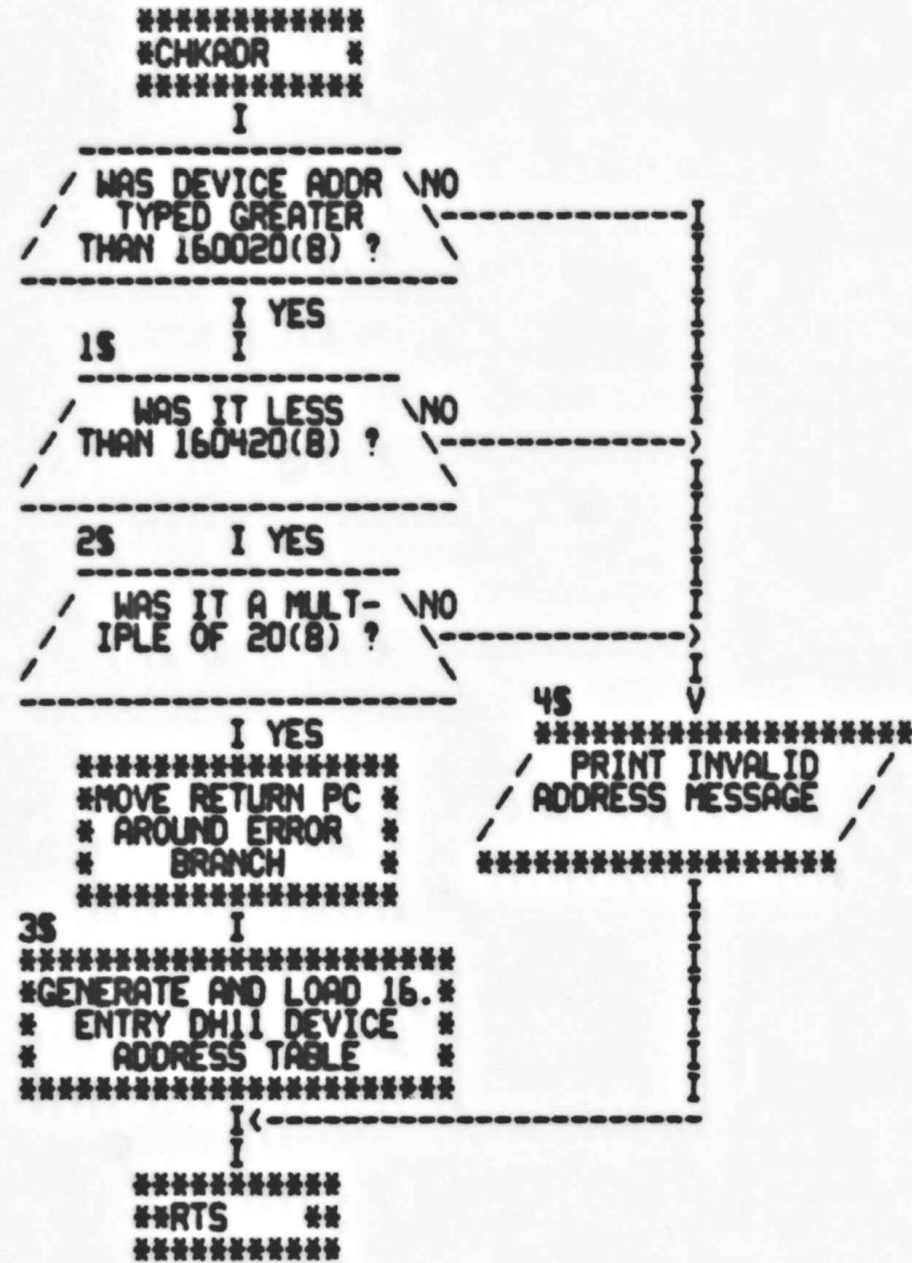
```

```

*****
#ENDA(51) *
*****
I
*****
#SCOPE(52)*
*****
# "NOP" SCOPE *
# CALL IN SEOP *
# ROUTINE *
*****
I
*****
# GENERATE NEXT *
# DEVICE NO. *
*****
I
*****
# UPDATE DEVICE PARA- *
# METER TABLE POINTERS *
# SHIFT SELECT MASK *
*****
I
-----
/ \
TESTED ALL POSSIBLE DH11'S \ / YES
-----
I NO
-----
/ \
IS THIS DH11 SELECTED FOR TEST \ /
-----
I YES
*****
# CLEAR $STNM *
# TO START AT *
# TST 1 AGAIN *
*****
I
*****
#RSTRTA(02)*
*****
I
*****
# CALL "SEOP" TO *
# REPORT END OF *
# PASS *
*****
I
-----
I
-----
I
-----

```





```

*****
#BUSER *
*****
I
*****
# SAVE THE PSW *
# AND STACK *
# POINTER *
*****
I
*****
# RETRIEVE AND SAVE *
# "TRAPPC" AND "TRAPPS" *
# FROM STACK *
*****
I
*****
# INITIALIZE THE *
# STACK POINTER *
*****
I ERROR(53)
*****
## REPORT THE BUS ##
## ERROR ##
## EM14, DM4, DT5, DF2 ##
*****
I
*****
# RESET THE *
# WORLD AND *
# CLEAR PSW *
*****
I
*****
#REST1 *
*****

```

```

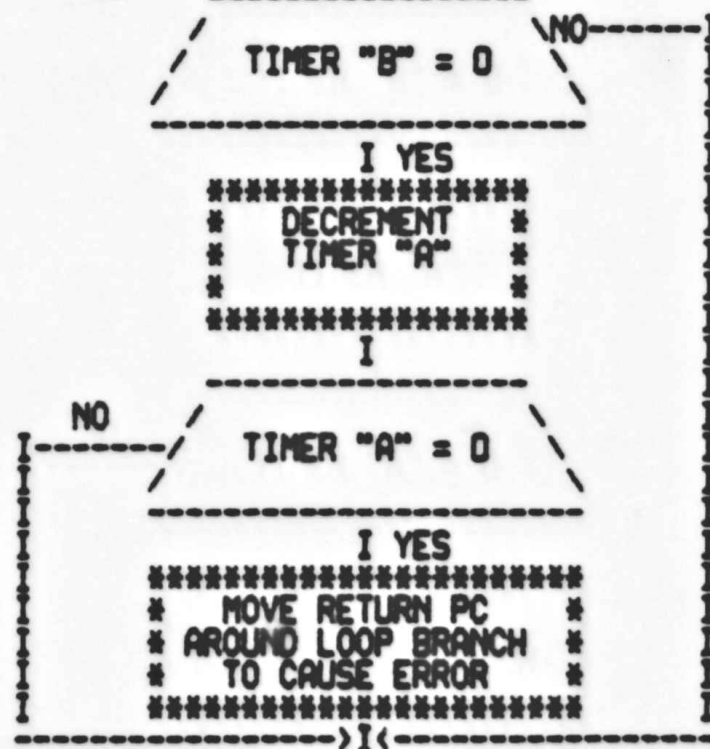
*****
#RESERR *
*****
I
*****
# SAVE THE PSW *
# AND STACK *
# POINTER *
*****
I
*****
# RETRIEVE AND SAVE *
# "TRAPPC" AND "TRAPPS" *
# FROM STACK *
*****
I
*****
# INITIALIZE *
# STACK POINTER *
*****
I ERROR(53)
*****
## REPORT THE RSVD ##
## INSTR ERROR ##
## EM15, DM4, DT5, DF2 ##
*****
I
*****
# RESET THE *
# WORLD AND *
# CLEAR PSW *
*****
I
*****
#REST1 *
*****

```

```

*****
#RESTRP *
*****
I
*****
# GET CURRENT *
# VECTOR ADDRESS *
*****
I
*****
# LOAD VECTOR (RCVR *
# AND (XMIT) WITH *
# TRAP CATCHER *
*****
I
*****
##RTS ##
*****
*****
#TIMEIT *
*****
I
*****
# INCREMENT *
# TIMER "B" *
*****
I
-----
TIMER "B" = 0
-----
I YES
*****
# DECREMENT *
# TIMER "A" *
*****
I
-----
TIMER "A" = 0
-----
I YES
*****
# MOVE RETURN PC *
# AROUND LOOP BRANCH *
# TO CAUSE ERROR *
*****
>I<
*****
##RTS ##
*****

```



```

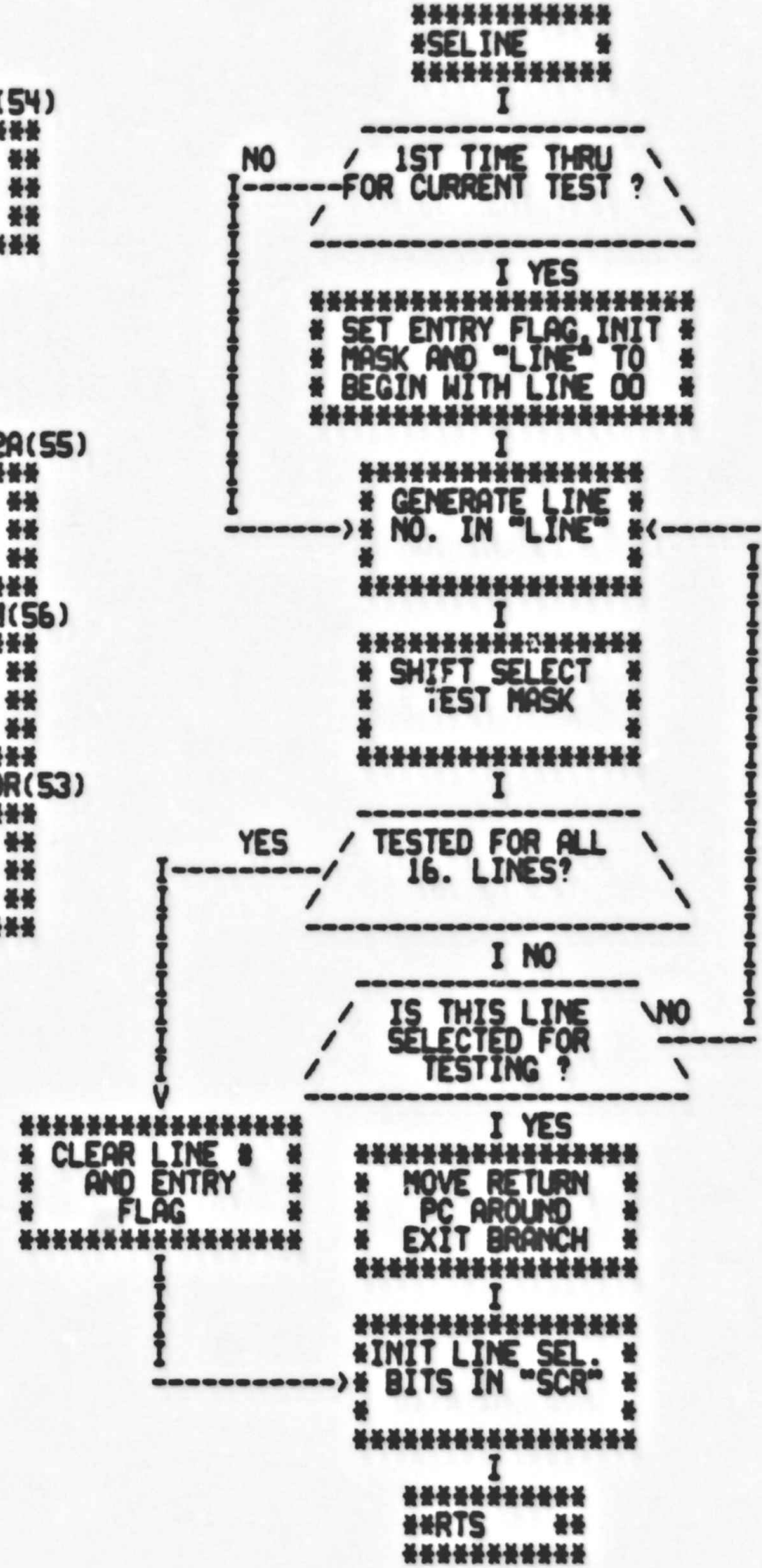
*****
#T40A(35) #
*****
      I      SAPS(54)
*****
** GO SAVE ERROR PSW **
**                               **
*****
      I
*****
# SAVE ERROR #
# INFORMATION #
#                               #
*****
      I      SUER2A(55)
*****
** GO SET UP INFORM- **
** ATION TO BE PRINTED **
**                               **
*****
      I      SUNUM(56)
*****
** GO PUT LINE NO. **
** IN ERROR MSG **
**                               **
*****
      I      SERROR(53)
*****
** REPORT STORAGE **
** OVERFLOW TIMEOUT **
** EM57,DH2,DT2,0 **
*****
      I
*****
#T40B(35) #
*****

```

```

*****
#T44X(39) #
*****
      I      SAPS(54)
*****
** GO SAVE ERROR PSW **
**                               **
*****
      I
*****
# SAVE ERROR #
# INFORMATION #
#                               #
*****
      I      SUER2A(55)
*****
** GO SET UP INFORM- **
** ATION TO BE PRINTED **
**                               **
*****
      I      SUNUM(56)
*****
** GO PUT LINE NO. **
** IN ERROR MSG **
**                               **
*****
      I      SERROR(53)
*****
** REPORT CHAR AVAIL **
** TIMEOUT ERROR **
** EM22,DH2,DT2,0 **
*****
      I
*****
#T44Y(40) #
*****

```



APF1	46																	
APF2	46																	
APF3	46																	
APF4	46																	
APF5	46																	
BEGINA	50																	
BRK1	49																	
BRK2	49																	
BRK3	49																	
BRK4	49																	
BRK5	49																	
BUSER	58																	
CHKADR	58																	
CHKVCT	58																	
CHPS1	50																	
CHPS2	50	56																
CLCABC	50																	
ENDA	50																	
HALT	50																	
INPAR	02																	
INPAR3	02																	
INPARA	02																	
LDBCR	54																	
LDTRF1	54																	
NPR1A	54																	
NPR2A	54																	
NPR3A	54																	
NPR4A	54																	
NPR5A	54																	
NPR6A	54																	
RESERR	59																	
REST1	59																	
RESTRP	01																	
RESTRT	01																	
RSTRTA	02	57																
RTI	43																	
RTS	54	54	54	55	55	55	55	55	56	56	56	57	57	58	58	59	59	
SAPS	54	55	60	60														
SELIN	14	15	17	18	19	20	21	22	28	30	35	36	37	38	39	41	44	
SETALL	48	50	51	60														
SLD1	40																	
SLD2	40	40																
SLD3	40	40	40															
START	01																	
START1	01																	
START2	01	01	02															
SUER1	03	05																
SUER2	04	05	06	07	07	07	08	08	09	09	11	11	12	12	13	13	14	
SUER2A	15	16	17	17	22	32	33	38	39	55								

TST5	07																		
TST50	45																		
TST51	47																		
TST52	48																		
TST53	50																		
TST54	51																		
TST6	58																		
TST7	59																		
ERROR	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78
SERTYP	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97
SOVER	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116
SCOPE	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135
SSVLAD	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154
15	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173
115	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192
25	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211
35	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230
45	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249
55	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268
65	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287
75	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306
85	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325
95	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344

2245	OPERATIONAL SWITCH SETTINGS
2247	TRAP CATCHER
(1)	STARTING ADDRESS(ES)
2251	ACT11 HOOKS
2252	APT PARAMETER BLOCK
2258	BASIC DEFINITIONS
2259	COMMON TAGS
(2)	APT MAILBOX-ETABLE
(1)	ERROR POINTER TABLE
2647	T1 CHECK SSYN RESPONSE FROM ALL DH11 REGISTERS
2671	T2 TEST THAT "MASTER CLR" CAN CLEAR THE "SCR", "LPR", "BKR", AND "SSR" REGS
2698	T3 TEST "SCR" REG R/W BITS CAN SET/CLR (NORMAL MODE)
2732	T4 TEST "SCR" REG. READ ONLY BITS (NORMAL MODE)
2755	T5 TEST "SCR" REG. BITS THAT CAN BE SET/CLR IN MAINT. MODE
2805	T6 TEST THAT ALL R/W BITS IN "LPR" CAN BE SET/CLR
2839	T7 TEST THAT ALL R/W BITS IN "BKR" CAN BE SET/CLR
2873	T10 TEST THAT ALL R/W BITS IN "SSR" CAN BE SET/CLR
2910	T11 TEST THAT CLR/SET OF BIT "N" IN "LPR" DOES NOT CLEAR ANY OTHER BITS
2948	T12 TEST THAT CLR/SET OF BIT "N" IN "BKR" DOES NOT CLEAR ANY OTHER BITS
2986	T13 TEST THAT CLR/SET OF BIT "N" IN "SSR" DOES NOT CLEAR ANY OTHER BITS
3026	T14 "CAR" MEMORY ADDRESSING TEST
3072	T15 "BCR" MEMORY ADDRESSING TEST
3118	T16 "CAR" REGISTER TEST - ALL 1'S / ALL 0'S - ALL LINES
3152	T17 "BCR" REGISTER TEST - ALL 1'S / ALL 0'S - ALL LINES
3186	T20 "CAR" MEMORY PATTERNS TEST / 0'S DISTURB
3232	T21 "BCR" MEMORY PATTERNS TEST / 0'S DISTURB
3278	T22 "CAR" MEMORY PATTERNS TEST / 1'S DISTURB
3323	T23 "BCR" MEMORY PATTERNS TEST / 1'S DISTURB
3368	T24 TEST THAT "CAR" MEMORY EXT BITS SET/CLR PROPERLY
3420	T25 TEST INTR. ENCB. BITS - INTR. CONDITION DISABLED
3464	T26 TEST CHAR. AVAIL. I.E. WITH INTR. CONDITION ACTIVE
3504	T27 TEST SILO OVFLW. I.E. WITH INTR. CONDITION ACTIVE
3544	T30 TEST NON EX MEM I.E. WITH INTR. CONDITION ACTIVE
3584	T31 TEST XMITTR DONE I.E. WITH INTR. CONDITION ACTIVE
3624	T32 BASIC TRANSMITTER "NPR" LOGIC TEST 1
3725	T33 TRANSMITTER NPR LOGIC TEST 2
3797	T34 TEST THAT CHARACTER AVAILABLE CAN CAUSE RCVR INTERRUPT
3853	T35 TEST THAT THE SILO STATUS REG COUNTS UP CORRECTLY
3886	T36 TEST THAT SILO STATUS REGISTER DOWN COUNTS CORRECTLY
3929	T37 TEST SILO ALARM LEVEL FOR COUNTS 0, 1, 2, 4, 8, 16, AND 32
3985	T40 VERIFY STORAGE OVERFLOW - NON MAINT. MODE - ALL LINES
4053	T41 TRANSMITTER TIMING TEST - ALL LINES - ALL SPEEDS
4118	T42 RECEIVER TIMING TEST - ALL LINES - ALL SPEEDS
4182	T43 BASIC DATA TEST - ALL LINES/ALL CHAR LENGTHS
4238	T44 SINGLE LINE DATA TEST - ALL LINES
4353	T45 BASIC PARITY LOGIC TEST - ALL LINES - ODD PARITY
4405	T46 MULTI-LINE PARITY DATA TEST - ALL SELECTED LINES
4515	T47 AUTO ECHO TEST 1 - ALL LINES
4594	T50 AUTO ECHO TEST 2 - ALL LINES
4689	T51 AUTO ECHO TEST 3 - ALL LINES
4750	T52 BREAK BIT TEST - ALL LINES
4868	T53 HALF DUPLEX TEST - ALL LINES
4917	T54 VERIFY THAT OVERRUN CAN SET PROPERLY - ALL LINES
4984	END OF PASS ROUTINE

4985	SCOPE HANDLER ROUTINE
4986	ERROR HANDLER ROUTINE
4987	ERROR MESSAGE TIMEOUT ROUTINE
4988	BINARY TO OCTAL (ASCII) AND TYPE
4989	CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
4990	TYPE ROUTINE
4991	APT COMMUNICATIONS ROUTINE
4992	TTY INPUT ROUTINE
4993	READ AN OCTAL NUMBER FROM THE TTY
4994	TRAP DECODER
(3)	TRAP TABLE
4995	POWER DOWN AND UP ROUTINES

2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
2245
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
2247
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
2248
2249
2250
2251
(1)
(1)
(1)
(1)
(1)

165400

000001

000000

000174 000000
000176 000000

000200 000137 023474
000204 000137 002144
000210 000137 023464

000214 000046

```

.NLIST CND,MD,MC
.LIST TOC,ME,SEQ,BIN
$SWR=165400
.MCALL .HEADER,.SCHTAG,.SETUP,.SERRR
.MCALL .SWRHI,.SWRLO,.SCATCH,.EQUATE,.SETUP,.SEOP
.MCALL .SSCOPE,.SERRTYP,.STYPCT,.STYPDEC,.STYPE,.SPOWER,.STRAP
.MCALL .SREAD,.SRDOCT
.MCALL .SACT11,.SAPTHDR,.SAPTBL5,.SAPTYPE
.ENABLE ABS
.TITLE MAINDEC-11-DZDMM-A
;COPYRIGHT (C) 1976
;DIGITAL EQUIPMENT CORP.
;MAYNARD, MASS. 01754
;*
;PROGRAM BY ED CROWLEY
;*
;THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;PACKAGE (MAINDEC-11-DZQAC-81),AUG 29,1975.
;*
$TN=1

.SBTTL OPERATIONAL SWITCH SETTINGS
;*
;* SWITCH USE
;* -----
;* 15 HALT ON ERROR
;* 14 LOOP ON TEST
;* 13 INHIBIT ERROR TYPEOUTS
;* 11 INHIBIT ITERATIONS
;* 9 LOOP ON ERROR
;* 8 LOOP ON TEST IN SWR<7:0>

.SBTTL TRAP CATCHER
.=0
;#ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
;#SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
;#LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
.=174
DISPREG: .WORD 0 ;;SOFTWARE DISPLAY REGISTER
SWREG: .WORD 0 ;;SOFTWARE SWITCH REGISTER

.SBTTL STARTING ADDRESS(ES)
JMP @INPARX ;;JUMP TO STARTING ADDRESS OF PROGRAM
JMP @BEGIN ;BEGIN EXECUTION WITH DEFAULT PARAMETERS
JMP @INPARC ;INPUT PARAMETERS - DEVICE SELECTION ONLY

;*****

.SBTTL ACT11 HOOKS
;HOOKS REQUIRED BY ACT11
$SVPC=. ;SAVE PC
.=46

```

(1) 000046 020046
 (1) 000052 000052
 (1) 000052 120000
 (1) 000214 000214
 2252
 (1)
 (1)
 (1)
 (2)
 (1) 000214 000214
 (1) 000024 000024
 (1) 000024 000200
 (1) 000044 000044
 (1) 000044 000214
 (1) 000214 000214
 (2)
 (1)
 (1)
 (1)
 (1) 000214 000000
 (1) 000214 000000
 (1) 000216 001230
 (1) 000220 000036
 (1) 000222 000170
 (1) 000224 000170
 (1) 000226 000052

SENDAD ;:1)SET LOC.46 TO ADDRESS OF SENDAD IN .SEOP
 .=52
 .WORD 120000 ;:2)SET LOC.52 TO 120000
 .=\$SVPC ;: RESTORE PC
 ;*****

.SBTTL APT PARAMETER BLOCK
 ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
 ;*****
 .SX=. ;:SAVE CURRENT LOCATION
 =24 ;:SET POWER FAIL TO POINT TO START OF PROGRAM
 200 ;:FOR APT START UP
 =44 ;:POINT TO APT INDIRECT ADDRESS PNTR.
 SAPTHDR ;:POINT TO APT HEADER BLOCK
 =.SX ;:RESET LOCATION COUNTER
 ;*****
 ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
 ;INTERFACE SPEC.

SAPTHD:
 SHIBTS: .WORD 0 ;:TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
 SMBADR: .WORD \$MAIL ;:ADDRESS OF APT MAILBOX (BITS 0-15)
 STSTM: .WORD 30 ;:RUN TIM OF LONGEST TEST
 SPASTM: .WORD 120 ;:RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
 SUNITH: .WORD 120 ;:ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
 .WORD SETEND-\$MAIL/2 ;:LENGTH MAILBOX-ETABLE(WORDS)

2257
 2258
 (1)
 (1)
 (1)
 (1) 001100
 (1)
 (1)
 (1) 177776
 (1)
 (1) 177774
 (1) 177772
 (1) 177570
 (1) 177570
 (1)
 (1)
 (1) 000000
 (1) 000001
 (1) 000002
 (1) 000003
 (1) 000004
 (1) 000005
 (1) 000006
 (1) 000007
 (1)
 (1)
 (1)
 (1)

.SBTTL BASIC DEFINITIONS
 ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
 STACK= 1100
 .EQUIV EMT,ERROR ;:BASIC DEFINITION OF ERROR CALL
 .EQUIV IOT,SCOPE ;:BASIC DEFINITION OF SCOPE CALL
 PS= 177776 ;:PROCESSOR STATUS WORD
 .EQUIV PS,PSW
 STKLM= 177774 ;:STACK LIMIT REGISTER
 PIR= 177772 ;:PROGRAM INTERRUPT REQUEST REGISTER
 DSMR= 177570 ;:HARDWARE SWITCH REGISTER
 DDISP= 177570 ;:HARDWARE DISPLAY REGISTER
 ;*GENERAL PURPOSE REGISTER DEFINITIONS
 R0= %0 ;:GENERAL REGISTER
 R1= %1 ;:GENERAL REGISTER
 R2= %2 ;:GENERAL REGISTER
 R3= %3 ;:GENERAL REGISTER
 R4= %4 ;:GENERAL REGISTER
 R5= %5 ;:GENERAL REGISTER
 R6= %6 ;:GENERAL REGISTER
 R7= %7 ;:GENERAL REGISTER
 .EQUIV R6,SP ;:STACK POINTER
 .EQUIV R7,PC ;:PROGRAM COUNTER
 ;*PRIORITY LEVEL DEFINITIONS

(1)	000000	PR0 =	0	:: PRIORITY LEVEL	0
(1)	000040	PR1 =	40	:: PRIORITY LEVEL	40
(1)	000100	PR2 =	100	:: PRIORITY LEVEL	100
(1)	000140	PR3 =	140	:: PRIORITY LEVEL	140
(1)	000200	PR4 =	200	:: PRIORITY LEVEL	200
(1)	000240	PR5 =	240	:: PRIORITY LEVEL	240
(1)	000300	PR6 =	300	:: PRIORITY LEVEL	300
(1)	000340	PR7 =	340	:: PRIORITY LEVEL	340

.*"SWITCH REGISTER" SWITCH DEFINITIONS

(1)	100000	SW15 =	100000
(1)	040000	SW14 =	40000
(1)	020000	SW13 =	20000
(1)	010000	SW12 =	10000
(1)	004000	SW11 =	4000
(1)	002000	SW10 =	2000
(1)	001000	SW09 =	1000
(1)	000400	SW08 =	400
(1)	000200	SW07 =	200
(1)	000100	SW06 =	100
(1)	000040	SW05 =	40
(1)	000020	SW04 =	20
(1)	000010	SW03 =	10
(1)	000004	SW02 =	4
(1)	000002	SW01 =	2
(1)	000001	SW00 =	1

.EQUIV SW09, SW08, SW07, SW06, SW05, SW04, SW03, SW02, SW01, SW00, SW09, SW08, SW07, SW06, SW05, SW04, SW03, SW02, SW01, SW00

.*DATA BIT DEFINITIONS (BIT00 TO BIT15)

(1)	100000	BIT15 =	100000
(1)	040000	BIT14 =	40000
(1)	020000	BIT13 =	20000
(1)	010000	BIT12 =	10000
(1)	004000	BIT11 =	4000
(1)	002000	BIT10 =	2000
(1)	001000	BIT09 =	1000
(1)	000400	BIT08 =	400
(1)	000200	BIT07 =	200
(1)	000100	BIT06 =	100
(1)	000040	BIT05 =	40
(1)	000020	BIT04 =	20
(1)	000010	BIT03 =	10
(1)	000004	BIT02 =	4
(1)	000002	BIT01 =	2
(1)	000001	BIT00 =	1

```

(1) .EQUIV BIT09,BIT9
(1) .EQUIV BIT08,BIT8
(1) .EQUIV BIT07,BIT7
(1) .EQUIV BIT06,BIT6
(1) .EQUIV BIT05,BIT5
(1) .EQUIV BIT04,BIT4
(1) .EQUIV BIT03,BIT3
(1) .EQUIV BIT02,BIT2
(1) .EQUIV BIT01,BIT1
(1) .EQUIV BIT00,BIT0

```

```

(1)
(1) 000004
(1) 000010
(1) 000014
(1) 000014
(1) 000014
(1) 000020
(1) 000024
(1) 000030
(1) 000034
(1) 000060
(1) C30064
(1) 000240

;#BASIC "CPU" TRAP VECTOR ADDRESSES
ERRVEC= 4           ;; TIME OUT AND OTHER ERRORS
RESVEC= 10          ;; RESERVED AND ILLEGAL INSTRUCTIONS
TBITVEC=14         ;; "T" BIT
TRTVEC= 14         ;; TRACE TRAP
BPTVEC= 14         ;; BREAKPOINT TRAP (BPT)
IOTVEC= 20         ;; INPUT/OUTPUT TRAP (IOT) **SCOPE**
PMRVEC= 24         ;; POWER FAIL
EMTVEC= 30         ;; EMULATOR TRAP (EMT) **ERROR**
TRAPVEC=34        ;; "TRAP" TRAP
TKVEC= 60          ;; TTY KEYBOARD VECTOR
TPVEC= 64          ;; TTY PRINTER VECTOR
PIRQVEC=240       ;; PROGRAM INTERRUPT REQUEST VECTOR

```


(1) 001220 000000
(1) 001222 000000
(1) 001224 077
(1) 001225 015
(1) 001226 000012

STIMES: 0
SESCAPE: 0
SQUES: .ASCII /?/
SCRLF: .ASCII <15>
SLF: .ASCIZ <12>

::: MAX. NUMBER OF ITERATIONS
::: ESCAPE ON ERROR ADDRESS
::: QUESTION MARK
::: CARRIAGE RETURN
::: LINE FEED


```

(2) 001312 000000 $CDW2: .WORD ACDW2 :::CONTROLLER DESCRIPTION WORD#2
(2) 001314 000000 $DDW0: .WORD ADDW0 :::DEVICE DESCRIPTOR WORD#0
(2) 001316 000000 $DDW1: .WORD ADDW1 :::DEVICE DESCRIPTOR WORD#1
(2) 001320 000000 $DDW2: .WORD ADDW2 :::DEVICE DESCRIPTOR WORD#2
(2) 001322 000000 $DDW3: .WORD ADDW3 :::DEVICE DESCRIPTOR WORD#3
(2) 001324 000000 $DDW4: .WORD ADDW4 :::DEVICE DESCRIPTOR WORD#4
(2) 001326 000000 $DDW5: .WORD ADDW5 :::DEVICE DESCRIPTOR WORD#5
(2) 001330 000000 $DDW6: .WORD ADDW6 :::DEVICE DESCRIPTOR WORD#6
(2) 001332 000000 $DDW7: .WORD ADDW7 :::DEVICE DESCRIPTOR WORD#7
(2) 001334 000000 $DDW8: .WORD ADDW8 :::DEVICE DESCRIPTOR WORD#8
(2) 001336 000000 $DDW9: .WORD ADDW9 :::DEVICE DESCRIPTOR WORD#9
(2) 001340 000000 $DDW10: .WORD ADDW10 :::DEVICE DESCRIPTOR WORD#10
(2) 001342 000000 $DDW11: .WORD ADDW11 :::DEVICE DESCRIPTOR WORD#11
(2) 001344 000000 $DDW12: .WORD ADDW12 :::DEVICE DESCRIPTOR WORD#12
(2) 001346 000000 $DDW13: .WORD ADDW13 :::DEVICE DESCRIPTOR WORD#13
(2) 001350 000000 $DDW14: .WORD ADDW14 :::DEVICE DESCRIPTOR WORD#14
(2) 001352 000000 $DDW15: .WORD ADDW15 :::DEVICE DESCRIPTOR WORD#15

```

(2) 001354

SETEND:

.SBTTL ERROR POINTER TABLE

```

;#THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
;#THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
;#LOCATION SITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
;#NOTE1: IF SITEMB IS 0 THE ONLY PERTINENT DATA IS (SERRPC).
;#NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

```

```

;*      EM          ;;POINTS TO THE ERROR MESSAGE
;*      DH          ;;POINTS TO THE DATA HEADER
;*      DT          ;;POINTS TO THE DATA
;*      DF          ;;POINTS TO THE DATA FORMAT

```

(1) 001354

```

SERRTB:
;ERROR TABLE ITEM FOR ERROR MESSAGE 1

```

```

2260
2261
2262 001354 025202      EM1          ;"DH11 REGISTER REFERENCE CAUSED TIMEOUT"
2263 001356 025251      DH1          ;" (PC) (PS) (SP) TEST DEVADR REGADR "
2264 001360 025330      DT1          ;SERRPC, STMPD, SREG6, SREG0, SREG1, SREG2
2265 001362 000000      0           ;PRINT ALL OCTAL

```

;ERROR TABLE ITEM FOR ERROR MESSAGE 2

```

2268
2269 001364 025346      EM2          ;"SYSTEM CONTROL REGISTER ERROR"
2270 001366 025404      DH2          ;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
2271 001370 025502      DT2          ;SERRPC, STMPD, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
2272 001372 000000      0           ;PRINT ALL OCTAL

```

;ERROR TABLE ITEM FOR ERROR MESSAGE 3

```

2273
2274
2275
2276 001374 025524      EM3          ;"DH11 MASTER CLEAR FAILED TO CLR SPECIFIED REG"

```

```

2277 001376 025404          DH2          : " (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
2278 001400 025502          DT2          : SERRPC, STMPD, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
2279 001402 000000          0            : PRINT ALL OCTAL
2280
2281 ;ERROR TABLE ITEM FOR ERROR MESSAGE 4
2282
2283 001404 025602          EM4          : "LINE PARAMETER REGISTER ERROR"
2284 001406 025404          DH2          : " (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
2285 001410 025502          DT2          : SERRPC, STMPD, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
2286 001412 000000          0            : PRINT ALL OCTAL
2287
2288 ;ERROR TABLE ITEM FOR ERROR MESSAGE 5
2289
2290 001414 025640          EM5          : "BREAK CONTROL REGISTER ERROR"
2291 001416 025404          DH2          : " (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
2292 001420 025502          DT2          : SERRPC, STMPD, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
2293 001422 000000          0            : PRINT ALL OCTAL
2294
2295 ;ERROR TABLE ITEM FOR ERROR MESSAGE 6
2296
2297 001424 025675          EM6          : "SILO STATUS REGISTER ERROR"
2298 001426 025404          DH2          : " (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
2299 001430 025502          DT2          : SERRPC, STMPD, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
2300 001432 000000          0            : PRINT ALL OCTAL
2301
2302 ;ERROR TABLE ITEM FOR ERROR MESSAGE 7
2303
2304 001434 025730          EM7          : "CURRENT ADDRESS REGISTER ERROR - LINE #XX"
2305 001436 025404          DH2          : " (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
2306 001440 025502          DT2          : SERRPC, STMPD, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
2307 001442 000000          0            : PRINT ALL OCTAL
2308
2309 ;ERROR TABLE ITEM FOR ERROR MESSAGE 10
2310
2311 001444 026002          EM10         : "BYTE COUNTER REGISTER ERROR - LINE #XX"
2312 001446 025404          DH2          : " (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
2313 001450 025502          DT2          : SERRPC, STMPD, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
2314 001452 000000          0            : PRINT ALL OCTAL
2315
2316 ;ERROR TABLE ITEM FOR ERROR MESSAGE 11
2317
2318 001454 026051          EM11         : "UNEXPECTED DH11 RCVR INTERRUPT"
2319 001456 025404          DH2          : " (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
2320 001460 025502          DT2          : SERRPC, STMPD, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
2321 001462 000000          0            : PRINT ALL OCTAL
2322
2323 ;ERROR TABLE ITEM FOR ERROR MESSAGE 12
2324
2325 001464 026110          EM12         : "UNEXPECTED DH11 XMITTR INTERRUPT"
2326 001466 025404          DH2          : " (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
2327 001470 025502          DT2          : SERRPC, STMPD, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
2328 001472 000000          0            : PRINT ALL OCTAL
2329
2330 ;ERROR TABLE ITEM FOR ERROR MESSAGE 13

```

2331				
2332	001474	026151	EM13	:"CHAR AVAILABLE FAILED TO GENERATE RCVR INTERRUPT"
2333	001476	025404	DH2	:" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
2334	001500	025502	DT2	:"SERRPC, STMPO, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
2335	001502	000000	0	:"PRINT ALL OCTAL
2336				
2337				;ERROR TABLE ITEM FOR ERROR MESSAGE 14
2338				
2339	001504	026232	EM14	:"TRANSMITTER NPR LOGIC ERROR - LINE # "
2340	001506	025404	DH2	:" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
2341	001510	025502	DT2	:"SERRPC, STMPO, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
2342	001512	000000	0	:"PRINT ALL OCTAL
2343				
2344				;ERROR TABLE ITEM FOR ERROR MESSAGE 15
2345				
2346	001514	026301	EM15	:"XMITTR FAILED TO INTERRUPT - LINE # "
2347	001516	025404	DH2	:" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
2348	001520	025502	DT2	:"SERRPC, STMPO, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
2349	001522	000000	0	:"PRINT ALL OCTAL
2350				
2351				;ERROR TABLE ITEM FOR ERROR MESSAGE 16
2352				
2353	001524	026347	EM16	:"RCVR FAILED TO INTERRUPT"
2354	001526	025404	DH2	:" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
2355	001530	025502	DT2	:"SERRPC, STMPO, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
2356	001532	000000	0	:"PRINT ALL OCTAL
2357				
2358				;ERROR TABLE ITEM FOR ERROR MESSAGE 17
2359				
2360	001534	026400	EM17	:"TRANSMITTER TIMING ERROR - LINE # "
2361	001536	026444	DH6	:" (PC) (PS) (SP) TEST DEVADR SPEED TIMEB TIMEC"
2362	001540	025502	DT2	:"SERRPC, STMPO, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
2363	001542	000000	0	:"PRINT ALL OCTAL
2364				
2365				;ERROR TABLE ITEM FOR ERROR MESSAGE 20
2366				
2367	001544	026542	EM20	:"RECEIVER TIMING ERROR - LINE # "
2368	001546	026444	DH6	:" (PC) (PS) (SP) TEST DEVADR SPEED TIMEB TIMEC"
2369	001550	025502	DT2	:"SERRPC, STMPO, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
2370	001552	000000	0	:"PRINT ALL OCTAL
2371				
2372				;ERROR TABLE ITEM FOR ERROR MESSAGE 21
2373				
2374	001554	026603	EM21	:"RCVR FAILED TO INTERRUPT - LINE # "
2375	001556	025404	DH2	:" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
2376	001560	025502	DT2	:"SERRPC, STMPO, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
2377	001562	000000	0	:"PRINT ALL OCTAL
2378				
2379				;ERROR TABLE ITEM FOR ERROR MESSAGE 22
2380				
2381	001564	026647	EM22	:"CHAR AVAIL FAILED TO SET ON TIME - LINE # "
2382	001566	025404	DH2	:" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
2383	001570	025502	DT2	:"SERRPC, STMPO, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
2384	001572	000000	0	:"PRINT ALL OCTAL

2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438

```

;ERROR TABLE ITEM FOR ERROR MESSAGE 23
EM23      ;"BASIC DATA TEST ERROR - LINE # "  
DH7      ;" (PC) (PS) (SP) TEST DEVADR CHRLNG WAS S/B "  
DT2      ;SERPC, STMPD, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4  
0        ;PRINT ALL OCTAL
    
```

```

;ERROR TABLE ITEM FOR ERROR MESSAGE 24
EM24      ;"AUTO ECHO TEST ERROR - LINE # "  
DH2      ;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "  
DT2      ;SERPC, STMPD, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4  
0        ;PRINT ALL OCTAL
    
```

```

;ERROR TABLE ITEM FOR ERROR MESSAGE 25
EM25      ;"BREAK BIT TEST ERROR - LINE # "  
DH2      ;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "  
DT2      ;SERPC, STMPD, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4  
0        ;PRINT ALL OCTAL
    
```

```

;ERROR TABLE ITEM FOR ERROR MESSAGE 26
EM26      ;"HALF-DUPLEX TEST ERROR - LINE # "  
DH2      ;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "  
DT2      ;SERPC, STMPD, SREG6, SREG7, SREG1, SREG2, SREG3, SREG4  
0        ;PRINT ALL OCTAL
    
```

```

;ERROR TABLE ITEM FOR ERROR MESSAGE 27
EM27      ;"UNEXPECTED BUS ERROR TRAP"  
DH3      ;" (PC) (PS) (SP) TEST TRPPC TRPPS  
DT3      ;SERPC, STMPD, SREG6, SREG0, SREG1, SREG2  
0        ;PRINT ALL OCTAL
    
```

```

;ERROR TABLE ITEM FOR ERROR MESSAGE 30
EM30      ;"UNEXPECTED RSVD INSTR TRAP"  
DH3      ;" (PC) (PS) (SP) TEST TRPPC TRPPS  
DT3      ;SERPC, STMPD, SREG6, SREG0, SREG1, SREG2  
0        ;PRINT ALL OCTAL
    
```

```

;ERROR TABLE ITEM FOR ERROR MESSAGE 31
EM31      ;"AUTO ECHO DATA COMPARE ERROR - LINE # "  
DH4      ;" (PC) (PS) (SP) TEST WASADR SBADR WAS S/B "  
DT2      ;SERPC, STMPD, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4  
0        ;PRINT ALL OCTAL
    
```

```

;ERROR TABLE ITEM FOR ERROR MESSAGE 32
EM32      ;"AUTO ECHO TEST TIMEOUT - LINE # "  

    
```

33
34
35
36
37
40
41
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

001666 027614
001670 027642
001672 000000

001674 027652
001676 025404
001700 025502
001702 000000

001704 027715
001706 027456
001710 025502
001712 000000

001714 030010
001716 030054
001720 030104
001722 000000

001724 030114
001726 027614
001730 027642
001732 000000

001734 030143
001736 027455
001740 025502
001742 000000

001744 030201
001746 025404
001750 025502
001752 000000

001754 030244
001756 027614
001760 027642
001762 000000

DH5
DT4
0
; (PC) (LPRG) TEST
;SERAPC,STMP0,STMP2
;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 33

EM33
DH2
DT2
0
;"PARITY LOGIC TEST ERROR - LINE # "
;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B"
;SERAPC,STMP0,SREG6,SREG0,SREG1,SREG2,SREG3,SREG4
;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 34

EM34
DH4
DT2
0
;"MULTI-LINE PARITY DATA TEST ERROR - LINE # - SUBTEST # "
;" (PC) (PS) (SP) TEST WASADR SBADR WAS S/B "
;SERAPC,STMP0,SREG6,SREG0,SREG1,SREG2,SREG3,SREG4
;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 35

EM35
DH14
DT6
0
;"MULTI-LINE PARITY DATA TEST TIMEOUT"
;" (PC) (LPRG) LINACTION "
;SERAPC,STMP0,STMP3
;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 36

EM36
DH5
DT4
0
;"CHAR AVAILABLE TIMEOUT"
;" (PC) (LPRG) TEST"
;SERAPC,STMP0,STMP2
;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 37

EM37
DH4
DT2
0
;"DATA COMPARE ERROR - LINE # "
;" (PC) (PS) (SP) TEST WASADR SBADR WAS S/B "
;SERAPC,STMP0,SREG6,SREG0,SREG1,SREG2,SREG3,SREG4
;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 40

EM40
DH2
DT2
0
;"BUFFER ACTIVE REG ERROR - LINE # "
;" (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
;SERAPC,STMP0,SREG6,SREG0,SREG1,SREG2,SREG3,SREG4
;PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 41

EM41
DH5
DT4
0
;"RCVR FALSE INTERRUPT"
;" (PC) (LPRG) TEST"
;SERAPC,STMP0,STMP2
;PRINT ALL OCTAL

03
04
05
06
07
08
09
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

001764 030271
001766 027614
001770 027642
001772 000000

001774 030315
001776 025404
002000 025502
002002 000000

002004 030375
002006 025404
002010 025502
002012 000000

002014 030457
002016 025404
002020 025502
002022 000000

002024 030535
002026 030623
002030 030720
002032 000000

002034 030742
002036 030623
002040 030720
002042 000000

002044 031023
002046 025404
002050 025502
002052 000000

002054 031101
002056 031130
002060 030720

;ERROR TABLE ITEM FOR ERROR MESSAGE 42

EM42 : "SILO OVERFLOW ERROR"
DH5 : " (PC) (LPRG) TEST"
DT4 : SERRPC, STMP0, STMP2
0 : PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 43

EM43 : "SILO OVERFLOW FAILED TO GENERATE RCVR INTERRUPT"
DH2 : " (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
DT2 : SERRPC, STMP0, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
0 : PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 44

EM44 : "NON EX MEMORY FAILED TO GENERATE XMITTR INTERRUPT"
DH2 : " (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
DT2 : SERRPC, STMP0, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
0 : PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 45

EM45 : "XMIT DONE FAILED TO GENERATE XMITTR INTERRUPT"
DH2 : " (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
DT2 : SERRPC, STMP0, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
0 : PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 46

EM46 : "CURRENT ADDRESS MEMORY PATTERNS TEST ERROR - LINE # "
DH10 : " (PC) LINE#R PATTRN TEST DEVADR REGADR WAS S/B "
DT5 : SERRPC, STMP0, STMP1, SREG0, SREG1, SREG2, SREG3, SREG4
0 : PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 47

EM47 : "BYTE COUNT MEMORY PATTERNS TEST ERROR - LINE # "
DH10 : " (PC) LINE#R PATTRN TEST DEVADR REGADR WAS S/B "
DT5 : SERRPC, STMP0, STMP1, SREG0, SREG1, SREG2, SREG3, SREG4
0 : PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 50

EM50 : "TEST TIMEOUT WAITING FOR XMIT DONE - LINE # "
DH2 : " (PC) (PS) (SP) TEST DEVADR REGADR WAS S/B "
DT2 : SERRPC, STMP0, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
0 : PRINT ALL OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 51

EM51 : "NPR LOGIC TEST 2 ERROR"
DH11 : " (PC) LINACT LINCHK TEST DEVADR REGADR WAS S/B "
DT5 : SERRPC, STMP0, STMP1, SREG0, SREG1, SREG2, SREG3, SREG4

```

2547 002062 000000          0          ;PRINT ALL OCTAL
2548
2549 ;ERROR TABLE ITEM FOR ERROR MESSAGE 52
2550
2551 002064 031225          EM52          ;"BASIC DATA COMPARE ERROR"
2552 002066 025404          DH2           ;" (PC) (PS) (SP) TEST DEVAOR REGADR WAS S/B"
2553 002070 025502          DT2           ;SERRPC, STMP0, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4"
2554 002072 000000          0           ;PRINT ALL OCTAL
2555
2556 ;ERROR TABLE ITEM FOR ERROR MESSAGE 53
2557
2558 002074 031023          EM50          ;"TEST TIMEOUT WAITING FOR XMIT DONE - LINE # "
2559 002076 031256          DH12          ;" (PC) SPEED (SP) TEST DEVAOR REGADR WAS S/B"
2560 002100 025502          DT2           ;SERRPC, STMP0, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
2561 002102 000000          0           ;PRINT ALL OCTAL
2562
2563 ;ERROR TABLE ITEM FOR ERROR MESSAGE 54
2564
2565 002104 026647          EM22          ;"CHAR AVAIL FAILED TO SET ON TIME - LINE # "
2566 002106 031256          DH12          ;" (PC) SPEED (SP) TEST DEVAOR REGADR WAS S/B"
2567 002110 025502          DT2           ;SERRPC, STMP0, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
2568 002112 000000          0           ;PRINT ALL OCTAL
2569
2570 ;ERROR TABLE ITEM FOR ERROR MESSAGE 55
2571
2572 002114 026647          EM22          ;"CHAR AVAIL FAILED TO SET ON TIME - LINE # "
2573 002116 031353          DH13          ;" (PC) (PS) (SP) TEST DEVAOR CHRLNG SCRNAS SCRS/B
2574 002120 025502          DT2           ;SERRPC, STMP0, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4
2575 002122 000000          0           ;PRINT ALL OCTAL
2576
2577 ;ERROR TABLE ITEM FOR ERROR MESSAGE 56
2578
2579 002124 031452          EM56          ;"OVERRUN BIT FAILED TO SET - LINE # "
2580 002126 025404          DH2           ;" (PC) (PS) (SP) TEST DEVAOR REGADR WAS S/B"
2581 002130 025502          DT2           ;SERRPC, STMP0, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4"
2582 002132 000000          0           ;PRINT ALL OCTAL
2583
2584 ;ERROR TABLE ITEM FOR ERROR MESSAGE 57
2585
2586 002134 031517          EM57          ;"STORAGE OVERFLOW BIT FAILED - LINE # "
2587 002136 025404          DH2           ;" (PC) (PS) (SP) TEST DEVAOR REGADR WAS S/B"
2588 002140 025502          DT2           ;SERRPC, STMP0, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4"
2589 002142 000000          0           ;PRINT ALL OCTAL
2590

```


2593	002144	005000		BEGIN:	CLR	RO	:: INIT RO TO INDICATE DEFAULT PARAMETERS	
2594	002146	005067	023016	BEGINA:	CLR	TITFLG	:: INIT TITLE MESSAGE FLAG	
2596	002152	012706	001100		MOV	#SCNTAG,R6	:: FIRST LOCATION TO BE CLEARED	
(1)	002156	005026			CLR	(R6)+	:: CLEAR MEMORY LOCATION	
(1)	002160	022706	001126		CMP	#SBODAT,R6	:: DONE?	
(1)	002164	001374			BNE	.-6	:: LOOP BACK IF NO	
(1)	002166	012706	001100		MOV	#STACK,SP	:: SETUP THE STACK POINTER	
(1)	002172	012737	020102	000020	MOV	#SCOPE,#IOTVEC	:: IOT VECTOR FOR SCOPE ROUTINE	
(1)	002200	012737	000340	000022	MOV	#340,#IOTVEC+2	:: LEVEL 7	
(1)	002206	012737	020370	000030	MOV	#ERROR,#EMTVEC	:: EMT VECTOR FOR ERROR ROUTINE	
(1)	002214	012737	000340	000032	MOV	#340,#EMTVEC+2	:: LEVEL 7	
(1)	002222	012737	022402	000034	MOV	#TRAP,#TRAPVEC	:: TRAP VECTOR FOR TRAP CALLS	
(1)	002230	012737	000340	000036	MOV	#340,#TRAPVEC+2	:: LEVEL 7	
(1)	002236	012737	022444	000024	MOV	#SPWRDN,#PMRVEC	:: POWER FAILURE VECTOR	
(1)	002244	012737	000340	000026	MOV	#340,#PMRVEC+2	:: LEVEL 7	
(1)	002252	005067	176742		CLR	#TIMES	:: INITIALIZE NUMBER OF ITERATIONS	
(1)	002256	005067	176740		CLF	#ESCAPE	:: CLEAR THE ESCAPE ON ERROR ADDRESS	
(1)	002262	112767	000001	176625	MOVB	#1,#SERMAX	:: ALLOW ONE ERROR PER TEST	
(1)	002270	012767	002270	176610	MOV	#.#,SLPADR	:: INITIALIZE THE LOOP ADDRESS FOR SCOPE	
(1)	002276	012767	002276	176604	MOV	#.#,SLPERR	:: SETUP THE ERROR LOOP ADDRESS	
(2)	002304	013746	000004		MOV	#4,-(SP)	:: SAVE ERROR VECTOR	
(2)	002310	013746	000006		MOV	#6,-(SP)		
(2)	002314	012767	002330	175462	MOV	#64,4	:: SET UP TIME OUT VECTOR	
(2)	002322	005777	176610		TST	#SWR	:: TRY TO REFERENCE HARDWARE SWR	
(2)	002326	000407			BR	655	:: BRANCH IF NO TIMEOUT TRAP OCCURS	
(2)	002330	012767	000176	176600	545:	MOV	#SWREG,#MR	:: POINT TO SOFTWARE SWR
(2)	002336	012767	000174	176574		MOV	#DISPREG,DISPLAY	:: POINT TO SOFTWARE DISPLAY REG
(2)	002344	022626			CMP	(SP)+,(SP)+	:: RESTORE STACK	
(2)	002346	012637	000006	655:	MOV	(SP)+,#6	:: RESTORE ERROR VECTOR	
(2)	002352	012637	000004		MOV	(SP)+,#4		
(1)	002356	005067	176654		CLR	#PASS	:: CLEAR PASS COUNT	
(1)	002362	132767	000200	176661	BITB	#APTSIZE,#ENVH	:: TEST USER SIZE UNDER APT	
(1)	002370	001403			BEG	35	:: YES,USE NON-APT SWITCH	
(1)	002372	012767	001252	176536	MOV	#SSWREG,#MR	:: NO,USE APT SWITCH REGISTER	
(1)	002400			35:				

```

2599 002400 012767 023774 175376 START1: MOV      #BUSER,ERRVEC ;SET UP THE BUS ERROR VECTOR
2600 002406 012767 000340 175372      MOV      #340,ERRVEC+2
2601 002414 012767 024050 175366      MOV      #RESERR,RESVEC ;SET UP THE RSVD INSTR VECTOR
2602 002422 012767 000340 175362      MOV      #340,RESVEC+2
2603 002430 005767 022534          TST      TITFLG ;HAVE WE TYPED TITLE ONCE ?
2604 002434 001004          BNE      IS ;BR IF YES
2605 002436 104400          TYPE ;GO TYPE PROGRAM TITLE
2606 002440 031566          TITLE
2607 002442 005167 022522          COM      TITFLG ;SET FLAG - TYPE TITLE ONLY ONCE PER LOAD
2608 002446 005767 022424          IS: TST      VCFLG ;START AT 200 ??
2609 002452 001404          BEQ      IIS ;BR IF NOT
2610 002454 004767 020740          JSR      PC,INPARA ;GO ASK FOR PARAMETERS
2611 002460 005067 022412          CLR      VCFLG ;RE INIT VECTOR FLAG
2612 002464 005700          IIS: TST      RD ;USE DEFAULT PARAMETERS ?
2613 002466 001407          BEQ      START2 ;BR IF YES
2614 002470 022700 177777          CMP      #1,RO ;CHANGE DH SELECT PARAM ONLY ?
2615 002474 001002          BNE      2S ;BR IF NOT
2616 002476 000167 021044          JMP      INPAR3 ;GO ASK FOR SELECT PARAM.
2617 002502 000167 021000          2S: JMP      INPAR ;GO ASK FOR ALL PARAMETERS
2618
2619 002506 012767 024774 022434 START2: MOV      #DHADTB-2,ADPTR ;GET POINTER TO ADDRESS TABLE
2620 002514 012767 025034 022430      MOV      #DHVCTB-2,VCPTR ;GET POINTER TO VECTOR TABLE
2621 002522 012767 025076 022424      MOV      #BRVL-2,BRPTR ;GET POINTER TO BR LEVEL TABLE
2622 002530 012767 177777 022404      MOV      #1,DHNUM ;START WITH DH #00
2623 002536 012767 000001 021640      MOV      #1,SELMSK ;SET UP DH11 BIT TEST MARKER
2624
2625 002544 005267 022372          RESTRT: INC      DHNUM ;GENERATE DH11 DEV NUMBER
2626 002550 062767 000002 022372      ADD      #2,ADPTR ;UPDATE TABLE POINTERS
2627 002556 062767 000002 022366      ADD      #2,VCPTR
2628 002564 062767 000002 022362      ADD      #2,BRPTR
2629 002572 036767 021606 021606      BIT      SELMSK,DHSEL ;TEST FOR SELECTED DH11
2630 002600 001004          BNE      RSTRTA ;BR IF SELECTED FOR TEST
2631 002602 006367 021576          REST1: ASL      SELMSK ;SHIFT MARKER TO TEST NEXT DH11
2632 002606 001737          BEQ      START2 ;BR IF 16 TESTED - START OVER
2633 002610 000755          BR      RESTRT ;GO TEST IF THIS ONE SELECTED
2634 002612 017767 022332 021560 RSTRTA: MOV      #ADPTR,DHADR ;SET UP DH11 ADDRESS
2635 002620 017767 022326 021554      MOV      #VCPTR,DHVCT ;SET UP THE DH11 VECTOR ENTRY
2636 002626 017767 022322 022304      MOV      #BRPTR,DHRLVL ;GET BR LEVEL VALUES
2637 002634 004567 020256          JSR      RS,SUNUM ;GO SET DH NUMBER IN THE MESSAGE BUFFER
2638 002640 025142          DHNUM
2639 002642 031656          TITLE2+20
2640 002644 104400          TYPE ;GO PRINT "TESTING DH11 #XX"
2641 002646 031636          TITLE2
2642 002650 004767 017734          JSR      PC,LDTBF1 ;GO LOAD XMITTR OUTPUT BUFFER WITH
2643 ;BINARY COUNT PATTERN
2644 002654 012767 002654 176224      MOV      #.,SLPADR ;INIT SCOPE LOOP RETURN

```

```

2647
(3)
(3)
(2) 002662 000004
(1) 002664 012767 000001 176342
2648

```

```

;*****
;*TEST 1 CHECK SSYN RESPONSE FROM ALL DH11 REGISTERS
;*****
TST1: SCOPE
MOV #STN-1,STESTN ;;SET TEST NUMBER IN MAIL BOX

```

```

REM %
TEST ABSTRACT:
*****

```

THIS TEST ATTEMPTS TO REFERENCE EACH OF THE EIGHT REGISTERS IN THE DH11 SELECTED FOR TEST USING ITS ASSIGNED UNIBUS ADDRESS. IF ANY ADDRESS FAILS TO RESPOND A BUS ERROR TRAP VECTORS THE TEST TO THE ERROR SET-UP AND CALL ROUTINE. AFTER THE ERROR IS TYPED THE TEST WILL TEST THE NEXT DH11 ADDRESS IN SEQUENCE UNTIL ALL EIGHT ARE TESTED.

```

ERRORS:
*****

```

1.) [ERROR 1] REPORTS THAT THE REGISTER WHOSE ADDRESS IS IN R2 FAILED TO RESPOND WITH "SSYN" WHEN REFERENCED.

```

SYNC: (NONE)
*****

```

```

DEBUG:
*****

```

- 1.) PROBLEM IS MOST LIKELY THE M7277 MODULE.
- 2.) IF ALL EIGHT REGISTERS FAIL TO RESPOND, MAKE SURE THAT YOU CONFIGURED THE PROGRAM PROPERLY BEFORE STARTING. IF YOU DID, CHECK THE SETTINGS OF THE ADDRESS SELECT JUMPERS ON THE M7277 MODULE.
- 3.) IF ONE OR MORE RESPONDED PROPERLY, SET UP AN ERROR SCOPE LOOP AND BACKTRACK THROUGH THE LOGIC STARTING WITH THE KEY LOGIC SIGNALS LISTED BELOW.

```

KEY LOGIC:
*****

```

```

M7277 SH3 SSYN H CE2
DEVICE RESPONDING L E72-6
SH4 DEVICE SELECTED H E09-11

```

```

2649 002672 010102
2650 002674 010205
2651 002676 062705 000016
2652 002702 016746 175076
2653 002706 012767 002734 175070
2654 002714 162702 000002
2655
2656 002720 062702 000002 1$: ADD #2,R2 ;POINT TO A DH11 REGISTER
2657 002724 005712 2$: TST (R2) ;ACCESS DH11 REG ADDR
2658 002726 020205 CMP R2,R5 ;TESTED ALL EIGHT ??
2659 002730 001373 BNE 1$ ;BR IF NOT
2660 002732 000410 BR 4$ ;BR WHEN ALL 8 ARE DONE

```

2661									
2662	002734	004767	017672	3S:	JSR	PC, SUER1			:GO SET UP ERROR INFO
2663	002740	022626			CMP	(SP)+, (SP)+			:FIX SP BECAUSE OF TRAP
2664	002742	012767	002724	176140	MOV	#2S, SLPERR			:SET UP ERROR LOOP RETURN
2665	002750	104001			ERROR	1			:DH11 REGISTER FAILED TO RESPOND TO MSYN
2666									
2667	002752	000762			BR	1S			:GO TEST NEXT ONE
2668	002754	012667	175024	4S:	MOV	(SP)+, ERRVEC			:RESTORE BUS ERROR VECTOR

K11

MAINDEC-11-DZDMM-A
DZDMM.A.P11 T2

MACY11 27(663) 12-DEC-75 08:41 PAGE 49-1
TEST THAT "MASTER CLR" CAN CLEAR THE "SCR", "LPR", "BKR", AND "SSR" REGS

SEQ 0139

2677	003002	010125			MOV	R1,(R5)+	
2678	003004	062745	000016		ADD	#SSR,-(R5)	;GENERATE SSR ADDRESS
2679	003010	062745	000014		ADD	#BKR,-(R5)	;GENERATE BKR ADDRESS
2680	003014	062745	000004		ADD	#LPR,-(R5)	;GENERATE LPR ADDRESS
2681	003020	005745			TST	-(R5)	;POINT R5 TO FIRST ADDR ENTRY (SCR)
2682	003022	005004			CLR	R4	;RESULT S/B 000000 AFTER MASTER CLEAR
2683							
2684	003024	012502		1S:	MOV	(R5)+,R2	;GET REG ADDRESS
2685	003026	022705	024426		CMP	#MSTCLR+10,R5	;DONE ALL FOUR REGS ??
2686	003032	001415			BEQ	TST3	::BR IF YES
2687	003034	012712	177777	2S:	MOV	#-1,(R2)	;SET 1'S IN REGISTER
2688	003040	052711	004000		BIS	#BIT11,(R1)	;ISSUE MASTER CLEAR
2689	003044	011203			MOV	(R2),R3	;GET CONTENT OF REGISTER
2690	003046	001766			BEQ	1S	::BR IF IT'S ALL ZEROES
2691							
2692	003050	004767	017616		JSR	PC,SUER2	;GO SET UP ERROR INFO
2693	003054	012767	003034 176026		MOV	#2\$,SLPERR	;SET UP ERROR LOOP RETURN
2694	003062	104003			ERROR	3	;MASTER CLR FAILED TO CLR SEL. REG.
2695	003064	000757			BR	1S	;GO TEST NEXT REGISTER

M11

MAINDEC-11-DZDMM-A
DZDMM.A.P11 T3

MACY11 27(663) 12-DEC-75 08:41 PAGE 50-1
TEST "SCR" REG R/W BITS CAN SET/CLR (NORMAL MODE)

SEQ 0141

```

(1)
(1)
(1)
2700 003076 012767 003126 176004 X      MOV    #45,SLPERR      ;SET UP ERROR LOOP RETURN
2701 003104 010102                MOV    R1,R2          ;GET REGISTER ADDRESS
2702 003106 012705 000001                MOV    #1,R5          ;SET UP TO START WITH BIT00
2703
2704 003112 030567 021642 15:          BIT    R5,RGMSK1      ;SHALL WE TEST THIS BIT ?
2705 003116 001003                BNE    45             ;BR IF YES
2706 003120 006305 25:          ASL    R5             ;SHIFT TO TST NEXT BIT
2707 003122 35:
(2) 003122 001430                BEQ    TST4           ;: <BR IF DONE ALL R/W BITS>
2708 003124 000772                BR     15             ;GO TEST NEXT BIT
2709
2710 003126 010504 45:          MOV    R5,R4          ;RESULT S/B IN R4
2711 003130 005012                CLR    (R2)           ;INIT REG BEING TESTED
2712 003132 112761 000000 000016        MOVB   #0,SSR(R1)     ;SCOPE SYNC
2713 003140 010512                MOV    R5,(R2)        ;SET THE BIT
2714 003142 011203                MOV    (R2),R3        ;GET THE WAS DATA
2715 003144 020403                CMP    R4,R3          ;RESULT = S/B DATA ??
2716 003146 001403                BEQ    55             ;BR IF YES
2717
2718 003150 004767 017516                JSR    PC,SUER2       ;GO SET UP ERROR INFO
2719 003154 104002                ERROR  2              ;SELECTED BIT FAILED TO SET IN SCR
2720
2721 003156 005004 55:          CLR    R4             ;SET UP TO CLEAR THE BIT S/B=000000
2722 003160 112761 000000 000017        MOVB   #0,SSR+1(R1)  ;SCOPE SYNC
2723 003166 040512                BIC    R5,(R2)        ;CLR THE SELECTED BIT
2724 003170 011203                MOV    (R2),R3        ;GET THE WAS DATA
2725 003172 001403                BEQ    65             ;BR IF IT CLEARED
2726
2727 003174 004767 017472                JSR    PC,SUER2       ;GO SET UP THE ERROR INFO
2728 003200 104002                ERROR  2              ;SELECTED BIT FAILED TO CLEAR IN SCR
2729 003202 000746 65:          BR     25             ;GO SELECT NEXT BIT

```


MAINDEC-11-DZDMM-A
DZDMM.P11 T4

MACY11 27(663) 12-DEC-75 08:41 PAGE 51-1
TEST "SCR" REG. READ ONLY BITS (NORMAL MODE)

SEQ 0143

2750 003262 012767 003236 175620
2751 003270 104022
2752 003272 000756

MOV 838,SLPERR
ERROR 2
BR 28

;SET UP ERROR LOOP RETURN ADDR
;READ ONLY BIT SET IN "SCR"
;CONTINUE WITH NEXT BIT


```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
2757 003304 012767 003334 175576      X      MOV      #35,SLPERR      ;SET UP THE ERROR LOOP RETURN
2758 003312 010102                    MOV      R1,R2          ;MAKE IT REG ADDR TOO
2759 003314 012705 000001                    MOV      #1,R5          ;INIT BIT TEST MARKER
2760 003320 030567 021446      1S:     BIT      R5,RGMSK6     ;IS IT A READ ONLY BIT ??
2761 003324 001003                    BNE     3$              ;BR IF YES - TEST IT
2762 003326 006305      2S:     ASL     R5          ;SHIFT THE BIT MARKER
2763 003330 001457                    BEQ     TST6            ;BR IF DONE ALL SELECTED BITS
2764 003332 000772                    BR      1$              ;GO TEST FOR THIS BIT
2765
2766 003334 010504      3S:     MOV      R5,R4          ;SET UP S/B DATA
2767 003336 052704 001000                    BIS     #BIT09,R4      ;PUT IN THE MAINT. BIT
2768 003342 005012                    CLR     (R2)           ;INIT REG BEING TESTED
2769 003344 052712 001000                    BIS     #BIT09,(R2)    ;TURN ON MAINT. MODE
2770 003350 112761 000000 000016      MOVB    #0,SSR(R1)     ;SCOPE SYNC
2771 003356 050512                    BIS     R5,(R2)        ;SET THE SELECTED BIT
2772 003360 011203                    MOV     (R2),R3        ;GET THE WAS DATA
2773 003362 020304                    CMP     R3,R4          ;DID SELECTED BIT GET SET ??
2774 003364 001404                    BEQ     4$              ;BR IF IT DID
2775
2776 003366 004767 017300      JSR     PC,SUER2       ;GO SET UP ERROR INFO
2777 003372 104002      ERROR  2              ;SELECTED BIT FAILED TO SET IN MAINT MODE
2778 003374 000754      BR      2$              ;GO TEST NEXT BIT
2779
2780 003376 042712 001000      4S:     BIC     #BIT09,(R2)   ;TURN OFF MAINT. MODE
2781 003402 042704 001000                    BIC     #BIT09,R4      ;CLR MAINT BIT IN S/B DATA
2782 003406 112761 000000 000017      MOVB    #0,SSR+1(R1)  ;SCOPE SYNC
2783 003414 040512                    BIC     R5,(R2)        ;ATTEMPT TO CLR SELECTED BIT
2784 003416 011203                    MOV     (R2),R3        ;GET THE WAS DATA
2785 003420 020304                    CMP     R3,R4          ;DID BIT GET CLEARED ??
2786 003422 001404                    BEQ     5$              ;BR IF IT DIDN'T
2787
2788 003424 004767 017242      JSR     PC,SUER2       ;GO SET UP ERROR INFO
2789 003430 104002      ERROR  2              ;SELECTED BIT GOT CLEARED WITH MAINT MODE OFF
2790 003432 000735      BR      2$              ;GO TEST NEXT BIT
2791
2792 003434 012704 001000      5S:     MOV      #BIT09,R4   ;SET UP S/B DATA
2793 003440 050412                    BIS     R4,(R2)        ;SET MAINT. MODE
2794 003442 012761 000000 000004      MOV     #0,LPR(R1)    ;SCOPE SYNC
2795 003450 040512                    BIC     R5,(R2)        ;NOW CLR SELECTED BIT
2796 003452 011203                    MOV     (R2),R3        ;GET THE WAS DATA
2797 003454 020304                    CMP     R3,R4          ;DID BIT GET CLEARED OK ??
2798 003456 001723                    BEQ     2$              ;BR IF YES
2799
2800 003460 004767 017206      JSR     PC,SUER2       ;GO SET UP ERROR INFO
2801 003464 104002      ERROR  2              ;FAILED TO CLR SELECTED BIT IN MAINT MODE
2802 003466 000717      BR      2$              ;GO SELECT NEXT BIT FOR TEST

```

2805
(3)
(3)
(2)
(1)
2806
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)

003470 000004
003472 012767 000006 1755:14

: #TEST 6 TEST THAT ALL R/W BITS IN "LPR" CAN BE SET/CLR
: *****
TST6: SCOPE
MOV #STN-1,STESTN ;;SET TEST NUMBER IN MAIL BOX
REM X
TEST ABSTRACT:

THIS TEST VERIFIES THAT ALL R/W BITS IN THE "LPR" REGISTER CAN BE SET AND CLEARED INDIVIDUALLY. A BIT MASK (RGMSK3: 177767) IS USED TO DEFINE THE BITS TO BE TESTED (ALL BUT BIT03). THE TEST SEQUENCE IS AS FOLLOWS:

- 1. SELECT A BIT TO TEST
- 2. SET THE BIT AND VERIFY IT SET
- 3. CLEAR THE BIT AND VERIFY IT CLEARED
- 4. REPEAT 1 THRU 3 UNTIL ALL BITS TESTED

ANY ERRORS DETECTED ARE REPORTED AND AFTER THE ERROR, THE TEST RESUMES WITH THE NEXT BIT IN SEQUENCE.

ERRORS:

- 1.) [ERROR 4] IS CALLED TO REPORT BOTH FAIL TO SET AND FAIL TO CLEAR FAILTS.

SYNC:

- 1.) FAIL TO SET: M7277 SH4 LOAD SSR LOW BYTE H CR1
- 2.) FAIL TO CLEAR: M7277 SH4 LOAD SSR HIGH BYTE H CP2

DEBUG:

- 1.) IF ALL BITS FAIL THE PROBLEM IS MOST LIKELY THE M7277 MODULE (LPR LOAD SIGNALS)
- 2.) IF NOT THEN IT IS PROBABLY AN "LPR" REGISTER CHIP OR BAD OUTPUT DATA MUX CHIP, BOTH ON THE M7278 MODULE.

KEY LOGIC:

M7277	SH4	LOAD LPR H	EP2
M7278	SH5	LPR <15:12> L	(E52)
	SH6	LPR <11:08> L	(E37)
	SH7	LPR <07:04> L	(E59)
	SH8	LPR <03:00> L	(E61)
	SH5,6,7,8	OUTPUT MUX CHIPS	(74151'S PIN 2)

%

2807	003500	012767	003534	175402		MOV	#3\$,SLPERR	:SET UP THE ERROR LOOP RETURN
2808	003506	010102				MOV	R1,R2	:COPY IT IN R2
2809	003510	062702	000004			ADD	#LPR,R2	:GENERATE REGADR IN R2
2810	003514	012709	000001			MOV	#1,R5	:INIT BIT TEST MARKER
2811	003520	030567	021240		1\$:	BIT	R5,RGMSK3	:TEST THIS BIT ??
2812	003524	001003				BNE	3\$:BR IF YES
2813	003526	006305			2\$:	ASL	R5	:SHIFT THE MARKER
2814	003530	001430				BEQ	TST7	:BR IF DONE ALL BITS
2815	003532	000772				BR	1\$:GO TEST NXT BIT
2816								
2817	003534	010504			3\$:	MOV	R5,R4	:SET UP S/B DATA
2818	003536	005012				CLR	(R2)	:INIT REG BEING TESTED
2819	003540	112761	000000	000016		MOVB	#0,SSR(R1)	:SCOPE SYNC
2820	003546	010512				MOV	R5,(R2)	:SET LPR BIT
2821	003550	011203				MOV	(R2),R3	:GET THE WAS DATA
2822	003552	020304				CMP	R3,R4	:DID IT SET
2823	003554	001403				BEQ	4\$:BR IF IT SET PROPERLY
2824								
2825	003556	004767	017110			JSR	PC,SUER2	:GO SET UP ERROR INFO
2826	003562	104004				ERROR	4	:LPR BIT FAILED TO SET PROPERLY
2827								
2828	003564	005004			4\$:	CLR	R4	:GET READY TO CLEAR SELECTED BIT
2829	003566	112761	000000	000017		MOVB	#0,SSR+1(R1)	:SCOPE SYNC
2830	003574	040512				BIC	R5,(R2)	:CLEAR THE BIT
2831	003576	011203				MOV	(R2),R3	:GET THE WAS DATA
2832	003600	001752				BEQ	2\$:BR IF BIT CLEARED PROPERLY
2833								
2834	003602	004767	017064			JSR	PC,SUER2	:GO SET UP ERROR INFO
2835	003606	104004				ERROR	4	:LPR BIT FAILED TO CLEAR PROPERLY
2836	003610	000746				BR	2\$:GO SELECT NEXT BIT

2839
(3)
(3)
(2) 003612 000004
(1) 003614 012767 000007 175412
2840

: #TEST 7 TEST THAT ALL R/W BITS IN "BKR" CAN BE SET/CLR

TST7: SCOPE
MOV #STN-1,STESTN ;;SET TEST NUMBER IN MAIL BOX

REM %
TEST ABSTRACT:

THIS TEST VERIFIES THAT ALL BITS IN THE BREAK CONTROL REGISTER CAN BE SET AND CLEARED INDIVIDUALLY. IT USES A BIT MASK (RGMSK4: 177777) TO DEFINE THE R/W BITS (ALL 16.). RS ALWAYS CONTAINS THE BIT CURRENTLY SELECTED FOR TEST. THE TEST SEQUENCE IS AS FOLLOWS:

1. SELECT A BIT TO TEST
2. SET THE BIT AND VERIFY THAT IT SET PROPERLY
3. CLEAR THE BIT AND VERIFY THAT IT CLEARED PROPERLY
4. REPEAT 1 THRU 4 UNTIL ALL BITS HAVE BEEN TESTED.

ANY ERROR DETECTED IS REPORTED AND THE TEST RESUMES WITH THE NEXT BIT IN SEQUENCE.

ERRORS:

- 1.) [ERROR 5] IS CALLED TO REPORT BOTH FAIL TO SET PROPERLY AND FAIL TO CLEAR PROPERLY FAULTS.

SYNC:

- 1.) FAIL TO SET: M7277 SH4 LOAD SSR LOW BYTE H CR1
- 2.) FAIL TO CLR: M7277 SH4 LOAD SSR HIGH BYTE H CP2

DEBUG:

- 1.) THE ONLY DIFFERENCES IN THE DATA PATH HERE AND THAT FOT THE PREVIOUS TESTS ARE THE ACTUAL REGISTER CHIPS AND THE INPUT SELECTED ON THE OUTPUT DATA MULTIPLEXORS.
- 2.) IF ALL BITS FAIL THE PROBLEM IS MOST LIKELY THE M7277.
- 3.) IF ONLY ONE OR TWO FAIL THE PROBLEM IS MOST LIKELY THE M7278.

KEY LOGIC:

- | | | | |
|-------|-----|--|-------------|
| M7277 | SH4 | LOAD BCR H | FUI |
| | | DATA TO BUS H | EN2 |
| | | DATA SOURCE (A,B,C) H | DUI,DU2,DT2 |
| M7278 | SH5 | - SH8 74175 REGISTER CHIPS (E51,E38,E67,E60) | |
| | SH5 | - SH8 74151'S MUX CHIPS INPUT PIN 13 | |

(1)									
2841	003622	012767	003656	175260	%	MOV	#35, SLPERR	:	SET UP THE ERROR LOOP RETURN
2842	003630	010102				MOV	R1, R2	:	GENERATE "BKR" ADDRESS IN R2
2843	003632	062702	000014			ADD	#BKR, R2	:	
2844	003636	012705	000001			MOV	#1, R5	:	INIT BIT TEST MARKER
2845	003642	030567	021120		15:	BIT	R5, RG1SK4	:	TEST THIS BIT ??
2846	003646	001003				BNE	35	:	BR IF YES
2847	003650	00E305			25:	ASL	R5	:	SHIFT BIT MARKER
2848	003652	001430				BEQ	TST10	:	BR IF ALL BITS TESTED
2849	003654	000772				BR	15	:	GO TEST THE BIT
2850									
2851	003656	010504			35:	MOV	R5, R4	:	SET UP S/B DATA
2852	003660	005012				CLR	(R2)	:	INIT REG BEING TESTED
2853	003662	112761	000000	000016		MOVB	#0, SSR(R1)	:	SCOPE SYNC
2854	003670	050512				BIS	R5, (R2)	:	SET THE SELECTED BIT IN "BKR"
2855	003672	011203				MOV	(R2), R3	:	GET THE WAS DATA
2856	003674	020304				CMF	R3, R4	:	DID BIT SET OK
2857	003676	001403				BEQ	45	:	BR IF YES
2858									
2859	003700	004767	016766			JSR	PC, SUER2	:	GO SET UP ERROR INFO
2860	003704	104005				ERROR	5	:	BKR BIT FAILED TO SET PROPERLY
2861									
2862	003706	005004			45:	CLR	R4	:	SET UP S/B DATA
2863	003710	112761	000000	000017		MOVB	#0, SSR+1(R1)	:	SCOPE SYNC
2864	003716	040512				BIC	R5, (R2)	:	CLEAR BKR BIT
2865	003720	011203				MOV	(R2), R3	:	GET THE BKR WAS DATA
2866	003722	001752				BEQ	25	:	BR IF BKR BIT CLEARED OK
2867									
2868	003724	004767	016742			JSR	PC, SUER2	:	GO SET UP ERROR INFO
2869	003730	104005				ERROR	5	:	BKR BIT FAILED TO CLR PROPERLY
2870	003732	000746				BR	25	:	GO SELECT NEXT BIT


```

(1)
(1)
(1)
M7278 SHS - SH8 REGISTER CHIPS E53,E68, OR E69 (74175'S)
OUTPUT MUX CHIPS - (74151'S PIN 12)
2875 003744 012767 004000 175136 X MOV #35,SLPERR ;SET UP THE ERROR LOOP RETURN
2876 003752 010102 MOV R1,R2 ;GENERATE "SSR" ADDRESS IN R2
2877 003754 062702 000016 ADD #SSR,R2
2878 003760 012705 000001 MOV #1,R5 ;INIT BIT TEST MARKER
2879 003764 030567 021000 1S: BIT R5,RGMSK5 ;TEST THIS BIT ??
2880 003770 001003 BNE 3S ;BR IF YES
2881 003772 006305 2S: ASL R5 ;SHIFT BIT MARKER
2882 003774 001435 BEQ TST11 ;BR IF ALL BITS TESTED
2883 003776 000772 BR 1S ;GO TEST THE BIT
2884
2885 004000 010504 3S: MOV R5,R4 ;SET UP S/B DATA
2886 004002 005012 CLR (R2) ;INIT REG BEING TESTED
2887 004004 012761 000000 000004 MOV #0,LPR(R1) ;SCOPE SYNC
2888 004012 050512 BIS R5,(R2) ;SET THE SELECTED BIT IN "SSR"
2889 004014 011203 MOV (R2),R3 ;GET THE WAS DATA
2890 004016 042703 077700 BIC #77700,R3 ;CLEAR OUT DON'T CARE BITS
2891 004022 020304 CMP R3,R4 ;DID BIT SET OK
2892 004024 001403 BEQ 4S ;BR IF YES
2893
2894 004026 004767 016640 JSR PC,SUER2 ;GO SET UP ERROR INFO
2895 004032 104006 ERROR 6 ;SSR BIT FAILED TO SET PROPERLY
2896
2897 004034 005004 4S: CLR R4 ;SET UP S/B DATA
2898 004036 012761 000000 000014 MOV #0,BKR(R1) ;SCOPE SYNC
2899 004044 040512 BIC R5,(R2) ;CLEAR SSR BIT
2900 004046 011203 MOV (R2),R3 ;GET THE SSR WAS DATA
2901 004050 042703 077700 BIC #77700,R3 ;CLEAR JUNK BITS
2902 004054 020304 CMP R3,R4 ;DID THE SSR BIT GET CLEARED ??
2903 004056 001403 BEQ 2S ;BR IF SSR BIT CLEARED OK
2904
2905 004060 004767 016606 JSR PC,SUER2 ;GO SET UP ERROR INFO
2906 004064 104006 ERROR 6 ;SSR BIT FAILED TO CLR PROPERLY
2907 004066 000741 BR 2S ;GO SELECT NEXT BIT

```

2910 (3) 004070 000004
 (3) (2) 004072 012767 000011 175134

2911 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)

```

*****
;TEST 11 TEST THAT CLR/SET OF BIT "N" IN "LPR" DOES NOT CLEAR ANY OTHER BITS
*****
TST11: SCOPE
MOV #STN-1,STESTN ;;SET TEST NUMBER IN MAIL BOX

```

THIS TEST VERIFIES THAT SETTING AND CLEARING EACH R/W BIT IN THE "LPR" REGISTER DOES NOT DISTURB (CLEAR) ANY OTHER BIT IN THE REGISTER. A BIT MASK (RGMSK3: 177767) IS USED TO DEFINE THE R/W BITS (ALL BUT BIT 03). RS ALWAYS CONTAINS THE BIT CURRENTLY SELECTED FOR TEST. THE TEST SEQUENCE IS AS FOLLOWS:

1. SELECT A BIT TO TEST
2. SET ALL THE WRITABLE BITS
3. CLEAR THE SELECTED BIT - VERIFY IT CLEARED PROPERLY
4. SET THE SELECTED BIT - VERIFY IT SET PROPERLY
5. REPEAT 1 THRU 4 UNTIL ALL BITS ARE TESTED

ANY ERRORS DETECTED ARE REPORTED AND THEN THE TEST RESUMES WITH THE NEXT BIT IN SEQUENCE .

ERRORS: *****
 1.) [ERROR 4] IS CALLED TO REPORT BOTH FAIL TO CLEAR PROPERLY AND FAIL TO SET PROPERLY FAULTS.

SYNC: *****
 1.) FAIL TO CLR: M7277 LOAD SSR LOW BYTE H CR1
 2.) FAIL TO SET: M7277 LOAD SSR HIGH BYTE H CP2

DEBUG: *****
 1.) PROBLEMS DETECTED BY THIS TEST INDICATE ADJACENT BIT INTERFERENCE CAUSED BY CROSS TALK OR NOISE. PROBLEM IS MOST LIKELY THE M7278.

KEY LOGIC: (SAME AS FOR TEST 6)

2912 004100 012767 004134 175002
 2913 004106 010102
 2914 004110 062702 000004
 2915 004114 012705 000001
 2916 004120 030567 020640
 2917 004124 001003
 2918 004126 006305
 2919 004130 001436
 2920 004132 000772

```

%
MOV #3$,SLPERR ;SET UP THE ERROR LOOP RETURN
MOV R1,R2 ;SET UP THE REG ADDR
ADD #LPR,R2
MOV #1,RS ;INIT BIT TEST MASK
1$: BIT RS,RGMSK3 ;TEST THIS PIT ??
BNE 3$ ;BR IF YES
2$: ASL RS ;SHIFT THE BIT TEST MASK
BEQ TST12 ;BR IF TESTED ALL BITS
BR 1$ ;GO TEST THIS BIT

```

2921											
2922	004134	016704	020624		3S:	MOV	RGMSK3,R4				:SET UP S/B DATA
2923	004140	005012				CLR	(R2)				:INIT REG BEING TESTED
2924	004142	112761	000000	000016		MOVB	#0,SSR(R1)				:SCOPE SYNC
2925	004150	040504				BIC	R5,R4				:CLR BIT "N"
2926	004152	016712	020606			MOV	RGMSK3,(R2)				:SET ALL R/W BITS IN LPR
2927	004156	040512				BIC	R5,(R2)				:CLEAR BIT "N" IN LPR
2928	004160	011203				MOV	(R2),R3				:GET THE WAS DATA
2929	004162	020304				CMP	R3,R4				:DID IT CLEAR OK ?
2930	004164	001404				BEQ	4S				:BR IF YES
2931											
2932	004166	004767	016500			JSR	PC,SUER2				:GO SET UP ERROR INFO
2933	004172	104004				ERROR	4				:BIT "N" FAILED TO CLR PROPERLY
2934	004174	000754				BR	2S				:GO TEST NEXT BIT
2935											
2936	004176	050504			4S:	BIS	R5,R4				:SET BIT "N" IN S/B DATA
2937	004200	112761	000000	000017		MOVB	#0,SSR+1(R1)				:SCOPE SYNC
2938	004206	050512				BIS	R5,(R2)				:SET BIT "N" IN LPR
2939	004210	011203				MOV	(R2),R3				:GET THE WAS DATA
2940	004212	020304				CMP	R3,R4				:DID BIT "N" SET PROPERLY ?
2941	004214	001744				BEQ	2S				:BR IF YES
2942											
2943	004216	004767	016450			JSR	PC,SUER2				:GO SET UP ERROR INFO
2944	004222	104004				ERROR	4				:BIT "N" FAILED TO SET PROPERLY
2945	004224	000740				BR	2S				:GO SELECT NEXT BIT

N12

MAINDEC-11-DZDMM-A
DZDMM.P11 T12

MACY11 27(663) 12-DEC-75 08:41 PAGE 57-1
TEST THAT CLR/SET OF BIT "N" IN "BKR" DOES NOT CLEAR ANY OTHER BITS

SEQ 0155

2959								
2960	004272	016704	020470	3S:	MOV	RGMSK4,R4		:SET UP S/B DATA
2961	004276	040504			BIC	R5,R4		:CLR BIT "N"
2962	004300	005012			CLR	(R2)		:INIT REG BEING TESTED
2963	004302	016712	020460		MOV	RGMSK4,(R2)		:SET ALL R/W BITS IN BKR
2964	004306	112761	000000	000016	MOVB	#0,SSR(R1)		:SCOPE SYNC
2965	004314	040512			BIC	R5,(R2)		:CLEAR BIT "N" IN BKR
2966	004316	011203			MOV	(R2),R3		:GET THE WAS DATA
2967	004320	020304			CMP	R3,R4		:DID IT CLEAR OK ?
2968	004322	001404			BEQ	4S		:BR IF YES
2969								
2970	004324	004767	016342		JSR	PC,SUER2		:GO SET UP ERROR INFO
2971	004330	104005			ERROR	5		:BIT "N" FAILED TO CLR PROPERLY
2972	004332	000754			BR	2S		:GO TEST NEXT BIT
2973								
2974	004334	050504		4S:	BIS	R5,R4		:SET BIT "N" IN S/B DATA
2975	004336	112761	000000	000017	MOVB	#0,SSR+1(R1)		:SCOPE SYNC
2976	004344	050512			BIS	R5,(R2)		:SET BIT "N" IN BKR
2977	004346	011203			MOV	(R2),R3		:GET THE WAS DATA
2978	004350	020304			CMP	R3,R4		:DID BIT "N" SET PROPERLY ?
2979	004352	001744			BEQ	2S		:BR IF YES
2980								
2981	004354	004767	016312		JSR	PC,SUER2		:GO SET UP ERROR INFO
2982	004360	104005			ERROR	5		:BIT "N" FAILED TO SET PROPERLY
2983	004362	000740			BR	2S		:GO SELECT NEXT BIT

2986

(3)

(3)

(2)

(1)

2987

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

2988

2989

2990

2991

2992

2993

2994

2995

2996

004364 000004
004366 012767 000013 174640

:TEST 13 TEST THAT CLR/SET OF BIT "N" IN "SSR" DOES NOT CLEAR ANY OTHER BITS

TST13: SCOPE
MOV #STN-1,STESTN ;;SET TEST NUMBER IN MAIL BOX

.REM X
TEST ABSTRACT:

THIS TEST VERIFIES THAT CLEARING AND SETTING EACH R/W BIT IN THE SILO STATUS REGISTER INDIVIDUALLY DOES NOT DISTURB ANY OF THE OTHER BITS. A BIT MASK (RGMASK: 100077) IS USED TO DEFINE THE R/W BITS (15,5,4,3,2,1, AND 0). RS ALWAYS CONTAINS THE BIT CURRENTLY SELECTED FOR TEST. THE TEST SEQUENCE IS AS FOLLOWS:

1. SELECT A BIT TO TEST
2. SET ALL WRITABLE BITS IN THE "SSR"
3. CLEAR THE SELECTED BIT AND VERIFY THAT IT CLEARED PROPERLY
4. SET THE SELECTED BIT AND VERIFY THAT IT SET PROPERLY
5. REPEAT 1 THRU 4 UNTIL ALL BITS HAVE BEEN TESTED

ANY ERROR DETECTED IS REORTRD AND THEN THE TEST RESUMES WITH THE NEXT BIT IN SEQUENCE.

ERRORS:

1.) [ERROR 6] IS CALLED TO REPORT BOTH CLEAR AND SET FAULTS.

SYNC:

1.) FAIL TO CLR: M7277 SH4 LOAD LPR H EP2
2.) FAIL TO SET: M7277 SH4 LOAD BCR H FU1

DEBUG:

1.) LIKE THE PREVIOUS TEST, FAILURES HERE INDICATE ADJACENT BIT INTERFERENCE CAUSED BY CROSS TALK OR NOISE. THE FAULT IS MOST LIKELY THE M7278 MODULE.

KEY LOGIC: (SAME AS FOR TEST 10)

%
MOV #3S,SLPERR ;SET UP THE ERROR LOOP RETURN
MOV R1,R2 ;SET UP THE REG ADDR
ADD #SSR,R2
MOV #1,RS ;INIT BIT TEST MASK
1S: BIT RS,RGMASK ;TEST THIS BIT ??
BNE 3S ;BR IF YES
2S: ASL RS ;SHIFT THE BIT TEST MASK
BEQ TST14 ;BR IF TESTED ALL BITS
BR 1S ;GO TEST THIS BIT

2997											
2998	004430	016704	020334		3S:	MOV	RGMSK5,R4				:SET UP S/B DATA
2999	004434	040504				BIC	R5,R4				:CLR BIT "N"
3000	004436	005012				CLR	(R2)				:INIT REG BEING TESTED
3001	004440	016712	020324			MOV	RGMSK5,(R2)				:SET ALL R/W BITS IN SSR
3002	004444	012761	000000	000004		MOV	#0,LPR(R1)				:SCOPE SYNC
3003	004452	040512				BIC	R5,(R2)				:CLEAR BIT "N" IN SSR
3004	004454	011203				MOV	(R2),R3				:GET THE WAS DATA
3005	004456	042703	077700			BIC	#77700,R3				:CLEAR JUNK BITS
3006	004462	020304				CMP	R3,R4				:DID IT CLEAR OK ?
3007	004464	001404				BEQ	4S				:BR IF YES
3008											
3009	004466	004767	016200			JSR	PC,SUER2				:GO SET UP ERROR INFO
3010	004472	104006				ERROR	6				:BIT "N" FAILED TO CLR PROPERLY
3011	004474	000752				BR	2S				:GO TEST NEXT BIT
3012											
3013	004476	050504			4S:	BIS	R5,R4				:SET BIT "N" IN S/B DATA
3014	004500	012761	000000	000014		MOV	#0,BKR(R1)				:SCOPE SYNC
3015	004506	050512				BIS	R5,(R2)				:SET BIT "N" IN SSR
3016	004510	011203				MOV	(R2),R3				:GET THE WAS DATA
3017	004512	042703	077700			BIC	#77700,R3				:CLEAR JUNK BITS
3018	004516	020304				CMP	R3,R4				:DID BIT "N" SET PROPERLY ?
3019	004520	001740				BEQ	2S				:BR IF YES
3020											
3021	004522	004767	016144			JSR	PC,SUER2				:GO SET UP ERROR INFO
3022	004526	104006				ERROR	6				:BIT "N" FAILED TO SET PROPERLY
3023	004530	000734				BR	2S				:GO SELECT NEXT BIT

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)

1.) WRITE SYNC: M7277 SH4 LOAD SSR LOW BYTE H CR1
2.) READ SYNC: M7277 SH4 LOAD SSR HIGH BYTE H CP2

DEBUG:

1.) ANALYZE THE ERROR REPORTS CAREFULLY ASKING THE FOLLOWING QUESTIONS:

- A. DOES THE FAULT AFFECT ONLY ONE LINE ?
- B. DOES THE FAULT AFFECT ONLY ONE 4-BIT DATA GROUP ?
IE <15:12>, <11:08>, <07:04>, OR <03:00>
- C. DOES ANY DATA AT ALL APPEAR TO BE WRITTEN ?

- 2.) IF "A" IS TRUE THEN SUSPECT AN ADDRESSING PROBLEM IN THE MEMORY ADDRESS MUX.
- 3.) IF "B" IS TRUE THEN SUSPECT A DATA MUX, UP-COUNTER, MEMORY, OR INVERTER CHIP PROBLEM.
- 4.) IF "C" IS TRUE SUSPECT A MEMORY WRITE TIMING PROBLEM.
- 5.) IN MOST CASES THE FAULT IS MOST LIKELY THE M7277 OR M7278.

KEY LOGIC:

M7277 SH4 LOAD CA H E58-13
DATA TO BUS H EN2
DATA SOURCE (A,B,C) DU1,DU2,DT2

SH5 MEMADD SOURCE SEL H E55-8
CA MEM WRITE ENAB L E50-1
BUF ADDRS TO BUS H E33-1 (SHD BE LOW)
74157 MUX CHIPS E33,E27,E20 BITS<17:08>
74193 COUNTER CHIPS E19,E26,E32 BITS<17:08>
7489 MEMORY CHIPS E18,E25,E31 BITS<17:08>
7404 INVERTER CHIPS E30,E24,E17 BITS<17:08>

SH5 74157 MUX CHIP E48
74157 DATA MUX CHIPS E13,E06 BITS<07:00>
74193 COUNTER CHIPS E12,E05 BITS<07:00>
7489 MEMORY CHIPS E11,E04 BITS<07:00>

M7278 SH5 THRU SH8 74151 DATA MUX OUTPUT CHIPS (PIN 1 INPUT)

%
MOV R1,R2 ;COPY IT IN R2
ADD #CAR,R2 ;SET UP REGADR IN R2
MOV LINSEL,STMP7 ;SAVE LINE SELECT PARAMETER
MOV #-1,LINSEL ;DO ALL LINES FOR THIS TEST
JSR PC,SELIN ;GO SELECT A LINE NO.
BR 35 ;BR IF DONE ALL SELECTED LINES
TSTB LINE ;DOING LINE 00 ?
BNE 25 ;BR IF NOT

3028	004542	010102			
3029	004544	062702	000006		
3030	004550	016767	017634	174440	
3031	004556	012767	177777	017624	
3032	004564	004767	016234		
3033	004570	000415			
3034	004572	105767	020346		
3035	004576	001001			

3036	004600	005004			CLR	R4		; INIT TEST DATA
3037								
3038	004602	156711	020336		25:	BISB	LINE, (R1)	; SELECT A LINE
3039	004606	112761	000000	000016		MOVB	#0, SSR(R1)	; SCOPE SYNC
3040	004614	010412				MOV	R4, (R2)	; LOAD THE CAR REG.
3041	004616	062704	010421			ADD	#10421, R4	; GENERATE NEW DATA
3042	004622	000760				BR	IS	; GO DO NEXT LINE
3043								
3044	004624	004767	016174		35:	JSR	PC, SELINE	; GO SELECT A LINE NO.
3045	004630	000434				BR	75	; BR IF CHECKED ALL LINES
3046	004632	105757	020306			TSTB	LINE	; DOING LINE 00 ?
3047	004636	001001				BNE	45	; BR IF NOT
3048	004640	005004				CLR	R4	; INIT S/B DATA
3049								
3050	004642	156711	020276		45:	BISB	LINE, (R1)	; SELECT A LINE
3051	004646	112761	000000	000017		MOVB	#0, SSR+1(R1)	; SCOPE SYNC
3052	004654	011203				MOV	(R2), R3	; GET CONTENTS OF CAR
3053	004656	020304				CMP	R3, R4	; WAS DATA OK ?
3054	004660	001412				BEQ	55	; BR IF YES
3055								
3056	004662	004767	016004			JSR	PC, SUER2	; GO SET UP ERROR INFO
3057	004666	004567	016224			JSR	R5, SUNUM	; SET UP LINE NO. IN MSG BUFFER
3058	004672	025144				LINE		
3059	004674	025777				EM7+47		
3060	004676	012767	004714	174204		MOV	#65, SLPERR	; SET UP ERROR LOOP RETURN
3061	004704	104007				ERROR	7	; CAR ADDRESSING ERROR
3062								
3063	004706	062704	010421		55:	ADD	#10421, R4	; GENERATE NEW S/B DATA
3064	004712	000744				BR	35	; GO CHECK NEXT LINE
3065								
3066	004714	005067	020224		65:	CLR	LINE	; RESTART AT LINE 00 IF LOOPING
3067	004720	000721				BR	IS	; GO RESTART
3068								
3069	004722	016767	174270	017460	75:	MOV	STMP7, LINSEL	; RESTORE LINE SELECT PARAMETER

3081	004774	001001				BNE	25		:BR IF NOT
3082	004776	005004				CLR	R4		:INIT TEST DATA
3083									
3084	005000	156711	020140		25:	BISB	LINE,(R1)		:SELECT A LINE
3085	005004	112761	000000	000016		MOVB	#0,SSR(R1)		:SCOPE SYNC
3086	005012	010412				MOV	R4,(R2)		:LOAD THE BCR REG.
3087	005014	062704	010421			ADD	#10421,R4		:GENERATE NEW DATA
3088	005020	000760				BR	15		:GO DO NEXT LINE
3089									
3090	005022	004767	015776		35:	JSR	PC,SELINE		:GO SELECT A LINE NO.
3091	005026	000434				BR	75		:BR IF CHECKED ALL LINES
3092	005030	105767	020110			TSTB	LINE		:DOING LINE DO ?
3093	005034	001001				BNE	45		:BR IF NOT
3094	005036	005004				CLR	R4		:INIT S/B DATA
3095									
3096	005040	156711	020100		45:	BISB	LINE,(R1)		:SELECT A LINE
3097	005044	112761	000000	000017		MOVB	#0,SSR+1(R1)		:SCOPE SYNC
3098	005052	011203				MOV	(R2),R3		:GET CONTENTS OF BCR
3099	005054	020304				CMP	R3,R4		:WAS DATA OK ?
3100	005056	001412				BEG	55		:BR IF YES
3101									
3102	005060	004767	015606			JSR	PC,SUER2		:GO SET UP ERROR INFO
3103	005064	004567	016026			JSR	RS,SUNUM		:GO SET UP LINE NO. IN MSG BUFFER
3104	005070	025144				LINE			
3105	005072	026046				EM10+44			
3106	005074	012767	005112	174006		MOV	#65,SLPERR		:SET UP ERROR LOOP RETURN
3107	005102	104010				ERROR	10		:BCR ADDRESSING ERROR
3108									
3109	005104	062704	010421		55:	ADD	#10421,R4		:GENERATE NEW S/B DATA
3110	005110	000744				BR	35		:GO CHECK NEXT LINE
3111									
3112	005112	005067	020026		65:	CLR	LINE		:RESTART AT LINE 00 IF LOOPING
3113	005116	000721				BR	15		:GO RESTART
3114									
3115	005120	016767	174072	017262	75:	MOV	STMP7,LINSEL		:RESTORE THE LINE SELECT PARAMETER

K13

MAINDEC-11-DZDMM-A
DZDMM.P11 T16

MACY11 27(663) 12-DEC-75 08:41 PAGE 61-1
"CAR" REGISTER TEST - ALL 1'S / ALL 0'S - ALL LINES

SEQ 0165

3131	005200	010412			MOV	R4, (R2)	:LOAD A CAR WITH ALL ONES
3132	005202	011203			MOV	(R2), R3	:GET THE WAS DATA FROM THE CAR
3133	005204	020403			CMP	R4, R3	:DID IT CONTAIN ALL ONES ??
3134	005206	001406			BEG	3\$::BR IF ALL 1'S
3135							
3136	005210	004767	015456		JSR	PC, SUER2	:GO SET UP ERROR INFO
3137	005214	012767	005172	173666	MOV	#2\$, \$LPERR	:SET UP ERROR LOOP RETURN
3138	005222	104007			ERROR	7	:FAILED TO SET ALL 1'S IN SELECTED CAR
3139							
3140	005224	005004			CLR	R4	:RESULT IN CAR 5/8 = 000000
3141	005226	112761	000000	000017	MOVB	#0, SSR+1(R1)	:SCOPE SYNC
3142	005234	010412			MOV	R4, (R2)	:CLEAR SELECTED CAR
3143	005236	011203			MOV	(R2), R3	:GET THE WAS DATA
3144	005240	001741			BEG	1\$:BR IF CAR GOT CLEARED
3145							
3146	005242	004767	015424		JSR	PC, SUER2	:GO SET UP FOR ERROR CALL
3147	005246	012767	005224	173634	MOV	#3\$, \$LPERR	:SET UP ERROR LOOP RETURN
3148	005254	104007			ERROR	7	:FAILED TO CLR ALL BITS IN SELECTED CAR
3149	005256	000732			BR	1\$:GO TEST NEXT LINE

3152
(3)
(3)
(2) 005260 000004
(1) 005262 012767 000017 173744

: #TEST 17 "BCR" REGISTER TEST - ALL 1'S / ALL 0'S - ALL LINES
: *****
TST17: SCOPE
MOV #STN-1,STESTN ;;SET TEST NUMBER IN MAIL BOX

3153
(1)

REM %
TEST ABSTRACT:

THIS TEST VERIFIES THE ABILITY TO SET AND CLEAR ALL BITS IN ALL THE SELECTED LOCATIONS (LINES) OF THE BYTE COUNT MEMORY. IT USES THE CONFIGURATION PARAMETER (LINSEL:) TO DEFINE WHICH LINES TO TEST. THE TEST SEQUENCE IS AS FOLLOWS:

1. SELECT A LINE # TO TEST
2. LOAD THE SELECTED LOCATION WITH 177777
3. READ IT BACK TO VERIFY ALL BITS SET
4. LOAD THE SELECTED LOCATION WITH 000000
5. READ IT BACK TO VERIFY ALL BITS CLEARED
6. REPEAT STEPS 1 THRU 5 UNTIL ALL SELECTED LINES ARE TESTED.

ALL ERRORS ARE REPORTED AND THEN THE TEST RESUMES WITH THE NEXT LINE # IN SEQUENCE AS DEFINED BY "LINSEL".

ERRORS:

1.) [ERROR 10] IS CALLED TO REPORT ALL DATA COMPARE ERRORS

SYNC:

1.) WRITE 1'S: M7277 SH4 LOAD SSR LOW BYTE H CR1

2.) WRITE 0'S: M7277 SH4 LOAD SSR HIGH BYTE H CP2

DEBUG: (REFER TO TEST 15)

KEY LOGIC: (REFER TO TEST 15)

3154 005270 010102
3155 005272 062702 000010
3156 005276 004767 015522
3157 005302 000443
3158 005304 012704 177777
3159 005310 156711 017630
3160 005314 004567 015576
3161 005320 025144
3162 005322 026046
3163
3164 005324 112761 000000 000016 25:

%
MOV R1,R2 ;COPY IT INTO R2
ADD #BCR,R2 ;R2 GETS BCR ADDRESS
15: JSR PC,SELIN ;GO SELECT A LINE NO.
BR TST20 ;BR IF DONE ALL SELECTED LINES
MOV #-1,R4 ;RESULT IN BCR S/B = 177777
BISB LINE,(R1) ;SELECT A LINE NO.
JSR RS,SUNUM ;GO SET UP LINE NO. IN MSG BUFFER
LINE
EM10+44
25: MOVB #0,SSR(R1) ;SCOPE SYNC

M13

MAINDEC-11-DZDMM-A
DZDMM.PI; T17MACY11 27(663) 12-DEC-75 08:41 PAGE 62-1
"BCR" REGISTER TEST - ALL 1'S / ALL 0'S - ALL LINES

SEQ 0167

3165	005332	010412			MOV	R4, (R2)	: LOAD A BCR WITH ALL ONES
3166	005334	011203			MOV	(R2), R3	: GET THE WAS DATA FROM THE BCR
3167	005336	020403			CMP	R4, R3	: DID IT CONTAIN ALL ONES ??
3168	005340	001406			BEQ	3\$: BR IF ALL 1'S
3169							
3170	005342	004767	015324		JSR	PC, SUER2	: GO SET UP ERROR INFO
3171	005346	012767	005324	173534	MOV	#2\$, SLPERR	: SET UP ERROR LOOP RETURN
3172	005354	104010			ERROR	10	: FAILED TO SET ALL 1'S IN SELECTED BCR
3173							
3174	005356	005004			CLR	R4	: RESULT IN BCR S/B = 000000
3175	005360	112761	000000	000017	MOVB	#0, SSR+1(R1)	: SCOPE SYNC
3176	005366	010412			MOV	R4, (R2)	: CLEAR SELECTED BCR
3177	005370	011203			MOV	(R2), R3	: GET THE WAS DATA
3178	005372	001741			BEQ	1\$: BR IF BCR GOT CLEARED
3179							
3180	005374	004767	015272		JSR	PC, SUEP2	: GO SET UP FOR ERROR CALL
3181	005400	012767	005356	173502	MOV	#3\$, SLPERR	: SET UP ERROR LOOP RETURN
3182	005406	104010			ERROR	10	: FAILED TO CLR ALL BITS IN SELECTED BCR
3183	005410	000732			BR	1\$: GO TEST NEXT LINE

3186
(3)
(3)
(2) 005412 000004
(1) 005414 012767 000020 173612

;TEST 20 "CAR" MEMORY PATTERNS TEST / O'S DISTURB

TST20: SCOPE
MOV #STN-1,\$STEN ;;SET TEST NUMBER IN MAIL BOX

3187

.REM %
TEST ABSTRACT:

THIS TEST VERIFIES THAT WHEN A TEST PATTERN IS WRITTEN INTO LOCATION "N" OF THE "CAR" MEMORY, IT DOES NOT DISTURB ANY BITS IN ANY OTHER LOCATIONS. THERE ARE THREE TEST PATTERNS USED* (177777, 125252, 052525) FOR EACH LOCATION SELECTED BY THE CONFIGURATION PARAMETER "LINSEL". THE TEST SEQUENCE IS AS FOLLOWS:

1. SELECT A TEST PATTERN
2. SELECT A LINE # TO TEST
3. CLEAR ALL 16 LOCATIONS IN THE MEMORY
4. WRITE THE TEST PATTERN INTO THE SELECTED LOCATION
5. VERIFY THAT THE PATTERN WAS WRITTEN CORRECTLY AND THAT NO OTHER LOCATIONS WERE DISTURBED.
6. REPEAT 2 THRU 5 UNTIL ALL SELECTED LINES TESTED
7. REPEAT 1 THRU 6 UNTIL ALL THREE PATTERNS TESTED

ALL ERRORS ARE REPORTED AND THEN THE TEST RESUMES WITH CHECKING THE NEXT LINE IN SEQUENCE.

ERRORS:

1.) [ERROR 46] IS CALLED TO REPORT ANY ERROR DETECTED. THE INFORMATION PRINTED INCLUDES THE LINE # WRITTEN, THE LINE # BEING CHECKED, AND THE PATTERN USED.

SYNC*

1.) WRITE LINE: M7277 SH4 LOAD SSR LOW BYTE H CR1

2.) READ CHECK: M7277 SH4 LOAD SSR HIGH BYTE H CP2

DEBUG: (REFER TO TEST 14)

KEY LOGIC: (REFER TO TEST 14)

3188 005422 010102
3189 005424 062702 000006
3190 005430 012705 024426
3191 005434 012567 173542
3192 005440 001472
3193 005442 004767 015356
3194 005446 000772

%
MOV R1,R2 ;SET UP REGADR
ADD #CAR,R2
MOV #PATRMA,R5 ;SET UP POINTER TO DATA PATTERNS
1\$: MOV (R5)+,\$TMP1 ;GET A DATA TEST PATTERN
;BR IF DONE THREE PATTERNS
3192: BEQ TST21
;GO SELECT A LINE TO TEST
3193: JSR PC,SELIN
;BR IF DONE ALL SELECTED LINES
3194: BR 1\$

3195	005450	116767	017470	017470		MOVB	LINE,LINEA	;SAVE THE LINE NO. FOR ERROR LOOPING
3196								
3197	005456	105067	173524		2\$:	CLRB	STMP3	;INIT LINE COUNTER
3198	005462	116711	173520		3\$:	MOVB	STMP3,(R1)	;SELECT A LINE TO CLEAR
3199	005466	005012				CLR	(R2)	;CLR CAR FOR THAT LINE
3200	005470	105267	173512			INCB	STMP3	;GENERATE NEW LINE NO.
3201	005474	126727	173506	000020		CMPB	STMP3,#20	;DONE CLEARING ALL LINES ?
3202	005502	001367				BNE	3\$;BR IF NOT
3203								
3204	005504	116711	017436			MOVB	LINEA,(R1)	;SET LINE SELECT BITS
3205	005510	112761	000000	000016		MOVB	#0,SSA(R1)	;SCOPE SYNC
3206	005516	016712	173460			MOV	STMP1,(R2)	;LOAD CAR WITH TEST PATTERN
3207								
3208	005522	105067	173456			CLRB	STMP2	;INIT A LINE COUNTER
3209	005526	016704	173450		4\$:	MOV	STMP1,R4	;SET UP S/B DATA
3210	005532	116711	173446			MOVB	STMP2,(R1)	;SET LINE SELECT IN SCR
3211	005536	112761	000000	000017		MOVB	#0,SSA+1(R1)	;SCOPE SYNC
3212	005544	011203				MOV	(R2),R3	;GET MSG DATA
3213	005546	126767	173432	017370		CMPB	STMP2,LINE	;IS THIS THE LINE WITH THE TEST PATTERN
3214	005554	001401				BEQ	5\$;BR IF IT IS
3215	005556	005004				CLR	R4	;MAKE S/B DATA = 000000
3216	005560	020304			5\$:	CMP	R3,R4	;CORRECT DATA IN CAR ?
3217	005562	001412				BEQ	6\$;BR IF YES
3218								
3219	005564	004767	015170			JSR	PC,SUER4	;GO SET UP ERROR IN FO
3220	005570	004567	015322			JSR	R5,SUNUM	;GO SET UP LINE NO. IN MSG BUFFER
3221	005574	001204				STMP2		
3222	005576	030620				EM46+63		
3223	005600	012767	005456	173302		MOV	#2\$,SLPERR	;SET UP ERROR LOOP RETURN
3224	005606	104046				ERROR	46	;INCORRECT DATA READ FROM CAR
3225								
3226	005610	105267	173370		6\$:	INCB	STMP2	;GENERATE NEXT LINE NO.
3227	005614	122767	000020	173362		CMPB	#20,STMP2	;DONE ALL LINES ?
3228	005622	001707				BEQ	11\$;BR IF YES
3229	005624	000740				BR	4\$;GO CHECK NEXT LINE

3241	005664	116767	017254	017254	MOVB	LINE,LINEA	;SAVE THE LINE NO. FOR ERROR LOOP	
3242								
3243	005672	105067	173310		2S:	CLRB	STMP3	;INIT LINE COUNTER
3244	005676	116711	173304		3S:	MOVB	STMP3,(R1)	;SELECT A LINE TO CLEAR
3245	005702	005012				CLR	(R2)	;CLR BCR FOR THAT LINE
3246	005704	105267	173276			INCB	STMP3	;GENERATE NEW LINE NO.
3247	005710	126727	173272	000020		CMPB	STMP3,#20	;DONE CLEARING ALL LINES ?
3248	005716	001367				BNE	3S	;BR IF NOT
3249								
3250	005720	116711	017222			MOVB	LINEA,(R1)	;SET LINE SELECT BITS
3251	005724	112761	000000	000016		MOVB	#0,SSR(R1)	;SCOPE SYNC
3252	005732	016712	173244			MOV	STMP1,(R2)	;LOAD BCR WITH TEST PATTERN
3253								
3254	005736	105067	173242			CLRB	STMP2	;INIT A LINE COUNTER
3255	005742	016704	173234		4S:	MOV	STMP1,R4	;SET UP S/B DATA
3256	005746	116711	173232			MOVB	STMP2,(R1)	;SELECT A LINE TO CHECK
3257	005752	112761	000000	000017		MOVB	#0,SSR+1(R1)	;SCOPE SYNC
3258	005760	011203				MOV	(R2),R3	;GET WAS DATA
3259	005762	126767	173216	017154		CMPB	STMP2,LINE	;IS THIS THE LINE WITH THE TEST PATTERN
3260	005770	001401				BEQ	5S	;BR IF IT IS
3261	005772	005004				CLR	R4	;MAKE S/B DATA = 00000J
3262	005774	020304			5S:	CMP	R3,R4	;CORRECT DATA IN BCR ?
3263	005776	001412				BEQ	6S	;BR IF YES
3264								
3265	006000	004757	014754			JSR	PC,SUER4	;GO SET UP ERROR IN FO
3266	006004	004567	015106			JSR	R5,SUNUM	;GO SET UP LINE NO. IN MSG BUFFER
3267	006010	001204				STMP2		
3268	006012	031020				EM47+56		
3269	006014	012767	005672	173066		MOV	#2S,SLPERR	;SET UP ERROR LOOP RETURN
3270	006022	104047				ERROR	47	;INCORRECT DATA READ FROM BCR
3271								
3272	006024	105267	173154		6S:	INCB	STMP2	;GENERATE NEXT LINE NO.
3273	006030	122767	000020	173146		CMPB	#20,STMP2	;DONE ALL LINES ?
3274	006036	001707				BEQ	11S	;BR IF YES
3275	006040	000740				BR	4S	;GO CHECK NEXT LINE

3278
 (3)
 (3)
 (2) 006042 000004
 (1) 006044 012767 000022 173162
 3279

```

:*****
:*TEST 22      "CAR" MEMORY PATTERNS TEST / 1'S DISTURB
:*****
TST22: SCOPE
      MOV      #STN-1,STESTN  ;;SET TEST NUMBER IN MAIL BOX
      REM      X

```

```

TEST ABSTRACT:
*****

```

THIS TEST VERIFIES THAT WHEN ALL ZEROS ARE WRITTEN INTO LINE "N" IN THE "CAR" MEMORY, IT DOES NOT CLEAR ANY BITS IN ANY OTHER LOCATIONS. ONLY THE LINES SELECTED BY "LINSEL" ARE TESTED. THE TEST SEQUENCE IS AS FOLLOWS:

1. SELECT A LINE TO TEST
2. SET ALL ONES (177777) INTO ALL MEMORY LOCATIONS
3. CLEAR THE SELECTED LINE
4. VERIFY THAT ONLY THE SELECTED LINE WAS CLEARED AND ALL OTHER LINES STILL CONTAIN 177777
5. REPEAT STEPS 1 THRU 4 UNTIL ALL SELECTED LINES ARE TESTED

ALL ERRORS ARE REPORTED AND THEN THE TEST RESUMES CHECKING THE NEXT LINE IN SEQUENCE.

```

ERRORS:
*****

```

- 1.) [ERROR 46] IS CALLED TO REPORT ALL ERRORS. THE INFORMATION PRINTED INCLUDES THE LINE # WRITTEN, THE LINE # BEING CHECKED, AND THE PATTERN USED.

```

SYNC:
*****

```

- 1.) WRITE LINE: M7277 SH4 LOAD SSR LOW BYTE H CR1
- 2.) CHECK LINE: M7277 SH4 LOAD SSR HIGH BYTE H CP2

```

DEBUG: (REFER TO TEST 14)
*****

```

```

KEY LOGIC: (REFER TO TEST 14)
*****
%

```

3280 006052 010102
 3281 006054 062702 000006
 3282 006060 012705 177777
 3283 006064 010567 173112
 3284 006070 004767 014730
 3285 006074 000465
 3286 006076 116767 017042 017042
 3287
 3288 006104 105067 173076
 3289 006110 116711 173072

```

      MOV      R1,R2      ;SET UP REGADR
      ADD      #CAR,R2
      MOV      #-1,R5     ;TEST PATERN IN R5 = 177777
      MOV      R5,$TMP1   ;SAVE FOR ERROR REPORTING
1$:   JSR      PC,SELINE  ;GO SELECT A LINE TO TEST
      BR      TST23      ;BR IF DONE ALL LINES
      MOVB    LINS,LINEA ;SAVE THE LINE NO. FOR ERROR LOOP
2$:   CLRB    $TMP3      ;INIT LINE COUNTER
3$:   MOVB    $TMP3,(R1) ;SELECT A LINE TO CLEAR

```

3290	006114	010512			MOV	R5, (R2)		:LOAD CAR WITH 177777
3291	006116	105267	173064		INCB	STMP3		:GENERATE NEW LINE NO.
3292	006122	126727	173060	000020	CMPB	STMP3, #20		:DONE SETTING ALL LINES TO 177777 ?
3293	006130	001367			BNE	3\$:BR IF NOT
3295	006132	116711	017010		MOVB	LINEA, (R1)		:SET LINE SELECT IN SCR
3296	006136	112761	000000	000016	MOVB	#0, SSR(R1)		:SCOPE SYNC
3297	006144	005012			CLR	(R2)		:CLEAR THE CAR UNDER TEST
3299	006146	105067	173032		CLRB	STMP2		:INIT A LINE COUNTER
3300	006152	005004		4\$:	CLR	R4		:MAKE S/B DATA = 000000
3301	006154	116711	173024		MOVB	STMP2, (R1)		:SELECT A LINE TO CHECK
3302	006160	112761	000000	000017	MOVB	#0, SSR+1(R1)		:SCOPE SYNC
3303	006166	011203			MOV	(R2), R3		:GET WAS DATA
3304	006170	126767	173010	016746	CMPB	STMP2, LINE		:IS THIS THE LINE WITH THE TEST PATTERN
3305	006176	001401			BEQ	5\$:BR IF IT IS
3306	006200	010504			MOV	R5, R4		:MAKE S/B DATA = 177777
3307	006202	020304		5\$:	CMP	R3, R4		:CORRECT DATA IN CAR ?
3308	006204	001412			BEQ	6\$:BR IF YES
3309								
3310	006206	004767	014546		JSR	PC, SUER4		:GO SET UP ERROR IN FO
3311	006212	004567	014700		JSR	R5, SUNUM		:GO SET UP LINE NO. IN MSG BUFFER
3312	006216	001204			STMP2			
3313	006220	030620			EM46+63			
3314	006222	012767	006104	172660	MOV	#2\$, SLPERR		:SET UP ERROR LOOP RETURN
3315	006230	104046			ERROR	46		:INCORRECT DATA READ FROM CAR
3316								
3317	006232	105267	172746		INCB	STMP2		:GENERATE NEXT LINE NO.
3318	006236	122767	000020	172740	CMPB	#20, STMP2		:DONE ALL LINES ?
3319	006244	001711			BEQ	1\$:BR IF YES
3320	006246	000741			BR	4\$:GO CHECK NEXT LINE

3323
(3)
(3)
(2)
(1)
3324
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334

```

006250 000004
006252 012767 000023 172754

010102 000010
062702 177777
012705 172704
010567 014522
004767 016634 016634
105067 172670
116711 172664

```

```

*****
*TEST 23      "BCR" MEMORY PATTERNS TEST / 1'S DISTURB
*****
TST23: SCOPE
MOV      #STN-1,STESTN ;;SET TEST NUMBER IN MAIL BOX

```

```

.REM      %
TEST ABSTRACT:
*****

```

THIS TEST VERIFIES THAT WHEN ALL ZEROS ARE WRITTEN INTO LINE "N" IN THE "BCR" MEMORY, IT DOES NOT CLEAR ANY BITS IN ANY OTHER LOCATIONS. ONLY THE LINES SELECTED BY "LINSEL" ARE TESTED. THE TEST SEQUENCE IS AS FOLLOWS:

1. SELECT A LINE TO TEST
2. SET ALL ONES (177777) INTO ALL MEMORY LOCATIONS
3. CLEAR THE SELECTED LINE
4. VERIFY THAT ONLY THE SELECTED LINE WAS CLEARED AND ALL OTHER LINES STILL CONTAIN 177777
5. REPEAT STEPS 1 THRU 4 UNTIL ALL SELECTED LINES ARE TESTED

ALL ERRORS ARE REPORTED AND THEN THE TEST RESUMES CHECKING THE NEXT LINE IN SEQUENCE.

ERRORS:

1.) [ERROR 47] IS CALLED TO REPORT ALL ERRORS. THE INFORMATION PRINTED INCLUDES THE LINE # WRITTEN, THE LINE # BEING CHECKED, AND THE PATTERN USED.

SYNC:

- 1.) WRITE LINE: M7277 SH4 LOAD SSR LOW BYTE H CR1
- 2.) CHECK LINE: M7277 SH4 LOAD SSR HIGH BYTE H CP2

DEBUG: (REFER TO TEST 15)

KEY LOGIC: (REFER TO TEST 15)

```

X      MOV      R1,R2      ;SET UP REGADR
      ADD      #BCR,R2
      MOV      2-1,R5      ;TEST PATERAN IN R5 = 177777
      MOV      R5,$TMP1    ;SAVE IT FOR ERROR REPORTING
15:    JSR      PC,SELINE  ;GO SELECT A LINE TO TEST
      BR      TST24        ;BR IF DONE ALL LINES
      MOVB     LINE,LINEA  ;SAVE THE LINE NO.
25:    CLRB    $TMP3
35:    MOVB    $TMP3,(R1)  ;INIT LINE COUNTER
      ;SELECT A LINE TO INIT

```

3335	006322	010512			MOV	R5, (R2)		:LOAD BCR WITH 177777
3336	006324	105267	172656		INCB	\$TMP3		:GENERATE NEW LINE NO.
3337	006330	126727	172652	000020	CMPB	\$TMP3, #20		:DONE SETTING ALL LINES TO 177777 ?
3338	006336	001367			BNE	3\$:BR IF NOT
3339								
3340	006340	116711	016602		MOVB	LINEA, (R1)		:SET LINE SELECT BITS
3341	006344	112761	000070	000016	MOVB	#0, SSR(R1)		:SCOPE SYNC
3342	006352	005012			CLR	(R2)		:CLEAR THE BCR UNDER TEST
3343								
3344	006354	105067	172624		CLRB	\$TMP2		:INIT A LINE COUNTER
3345	006360	005004			4\$: CLR	R4		:MAKE S/B DATA = 000000
3346	006362	116711	172616		MOVB	\$TMP2, (R1)		:SELECT A LINE TO CHECK
3347	006366	112761	000000	000017	MOVB	#0, SSR+1(R1)		:SCOPE SYNC
3348	006374	011203			MOV	(R2), R3		:GET WAS DATA
3349	006376	126767	172602	016540	CMPB	\$TMP2, LINE		:IS THIS THE LINE WITH THE TEST PATTERN
3350	006404	001401			BEQ	5\$:BR IF IT IS
3351	006406	010504			MOV	R5, R4		:MAKE S/B DATA = 177777
3352	006410	020304			5\$: CMP	R3, R4		:CORRECT DATA IN BCR ?
3353	006412	001412			BEQ	6\$:BR IF YES
3354								
3355	006414	004767	014340		JSR	PC, SUER4		:GO SET UP ERROR IN FO
3356	006420	004567	014472		JSR	R5, SUNUM		:GO SET UP LINE NO. IN MSG BUFFER
3357	006424	001204			\$TMP2			
3358	006426	031020			EM47+56			
3359	006430	012767	006312	172452	MOV	#2\$, SLPERR		:SET UP ERROR LOOP RETURN
3360	006436	104047			ERROR	47		:INCORRECT DATA READ FROM BCR
3361								
3362	006440	105267	172540		6\$: INCB	\$TMP2		:GENERATE NEXT LINE NO.
3363	006444	122767	000020	172532	CMPB	#20, \$TMP2		:DONE ALL LINES ?
3364	006452	001711			BEQ	1\$:BR IF YES
3365	006454	000741			BR	4\$:GO CHECK NEXT LINE

3368
 (3)
 (3)
 (2) 006456 C00004
 (1) 006460 012767 000024 172546
 3369
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)

```

:*****
:*TEST 24 TEST THAT "CAR" MEMORY EXT BITS SET/CLR PROPERLY
:*****
TST24: SCOPE
      MOV #STN-1,STESTN ;;SET TEST NUMBER IN MAIL BOX
      REM X
TEST ABSTRACT:
*****
  
```

THIS TEST VERIFIES THAT THE "EXT MEM" BITS (CAR<17:16> CAN BE SET AND CLEARED IN ALL "CAR" MEMORY LOCATIONS. IT WRITES THE BINARY TEST PATTERNS (11, 01, AND 10) INTO BITS<17:16> TO CHECK EVERY MEMORY LOCATION. THE TEST SEQUENCE IS AS FOLLOWS:

1. SELECT A TEST PATTERN TO USE
2. CLEAR ALL 18 BITS IN ALL 16 LOCATIONS
3. SELECT A LINE TO TEST
4. WRITE THE TEST PATTERN INTO <17:16> OF THE SELECTED LOCATION
5. READ CHECK ALL LOCATIONS TO VERIFY THAT ONLY THE SELECTED LOCATION CONTAINS THE PATTERN
6. REPEAT STEPS 3 THRU 5 UNTIL ALL SELECTED LINES TESTED
7. REPEAT STEPS 1 THRU 6 UNTIL ALL PATTERNS USED

ALL ERRORS ARE REPORTED AND THEN THE TEST RESUMES CHECKING THE NEXT LINE IN SEQUENCE.

- NOTES: 1.) BITS<05:04> IN THE "SCR" ARE USED TO WRITE THE EXT MEM BITS
 2.) BITS<07:06> IN THE "SSR" ARE USED TO CHECK BITS<17:16>

ERRORS:

- 1.) [ERROR 7] IS CALLED TO REPORT ALL ERRORS

SYNC:

- 1.) WRITE CAR: M7277 SH4 LOAD LPR H EP2
- 2.) READ CAR: M7277 SH4 LOAD BCR H FU2

DEBUG:

- 1.) ASSUMING THAT THE PREVIOUS "CAR" MEMORY TESTS RAN ERROR FREE, THE PROBLEM IS EITHER THE M7277 OR THE M7278
- 2.) SET UP SCOPE ERROR LOOP AND START BACKTRACKING THROUGH THE LOGIC STARTING WITH THE KEY SIGNALS BELOW.

KEY LOGIC:

```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
3370 006466 010102 M7277 SH5 SCRO5 H CD2
3371 006470 062702 000016 SCRO4 H CE1
3372 006474 012705 024436 SSR07 H CF1
3373 006500 012567 172476 SSR06 H CH1
3374 006504 012567 172474
3375 006510 001505
3376
3377 006512 105067 172470 M7278 SH7 7415i MUX CHIPS E66 AND E58 (INPUT PIN 12)
3378 006516 142711 000017
3379 006522 156711 172460
3380 006526 142711 000060
3381 006532 005061 000006
3382 006536 105267 172444
3383 006542 122767 000020 172436
3384 006550 001362
3385
3386 006552 004767 014246
3387 006556 000750
3388 006560 156711 016360
3389 006564 156711 172412
3390 006570 012761 000000 000004
3391 006576 012761 000000 000006
3392
3393 006604 105067 172400
3394 006610 016704 172370
3395 006614 142711 000017
3396 006620 156711 172364
3397 006624 012761 000000 000014
3398 006632 016103 000006
3399 006636 011203
3400 006640 042703 177477
3401 006644 126767 016274 172336
3402 006652 001401
3403 006654 005004
3404 006656 020304
3405 006660 001412
3406
3407 006662 004767 014004
3408 006666 004567 014224
3409 006672 001210
3410 006674 025777
3411 006676 012767 006512 172204

```

```

%
MOV R1,R2 ;SET UP REGADR
ADD #SSR,R2
MOV #PATRN,R5 ;SET UP POINTER TO DATA PATTERNS
1S: MOV (R5)+,STMP1 ;GET THE PATTERNS
MOV (R5)+,STMP2
REQ TST25 ;;BR IF DONE ALL PATTERNS

2S: CLRB STMP3 ;INIT A LINE COUNTER
3S: BICB #17,(R1) ;INIT LINE SELECT BITS IN "SCR"
BISB STMP3,(R1) ;SELECT A LINE IN SCR
BICB #60,(R1) ;SO WE CLEAR ALL THE MEM EXT BITS
CLR CAR(R1) ;CLEAR A CAR
INCB STMP3 ;GENERATE NXT LINE NO.
CMPB #20,STMP3 ;CLEARED THE WHOLE THING ?
3S BNE ;BR IF NOT

4S: JSR PC,SELINE ;GO SELECT A LINE NO.
BR 1S ;BR IF DONE ALL LINES
BISB LINE,(R1) ;SET UP LINE SELECT BITS
BISB STMP1,(R1) ;SET UP MEM EXT BIT PATTERN
MOV #0,LPR(R1) ;SCOPE SYNC
MOV #0,CAR(R1) ;WRITE EXT BITS IN THIS LOCATION

5S: CLRB STMP4 ;INIT A LINE COUNTER
MOV STMP2,R4 ;SET UP S/B DATA
BICB #17,(R1) ;INIT SELECT BITS IN "SCR"
BISB STMP4,(R1) ;SET SELECT BITS IN SCR
MOV #0,BKR(R1) ;SCOPE SYNC
MOV CAR(R1),R3 ;READ THE SELECTED "CAR"
MOV (R2),R3 ;GET THE WAS DATA
BIC #177477,R3 ;CLEAR JUNK BITS
CMPB LINE,STMP4 ;LINE UNDER TEST ??
6S BEQ 6S ;BR IF YES
CLR R4 ;MAKE S/B DATA = 000000
CMP R3,R4 ;WERE MEM EXT BITS CORRECT ?
7S BEQ ;BR IF YES

JSR PC,SUER2 ;GO SET UP ERROR INFO
JSR R5,SUNUM ;GO SET LINE NO. IN MSG BUFFER
STMP4
EM7+47
NOV #2S,$L.PERR ;SET UP ERROR LOOP RETURN

```

K14

MAINDEC-11-DZDMM-A
DZDMM.A.P11 T24

MACY11 27(663) 12-DEC-75 08:41 PAGE 67-2
TEST THAT "CAR" MEMORY EXT BITS SET/CLR PROPERLY

SEQ 0178

3412	006704	104007			ERROR	7		;MEM EXT BITS READ INCORRECTLY
3413								
3414	006706	105267	172276		INCB	STMP4		;GENERATE NXT LINE NO.
3415	006712	122767	000020	172270	CMPB	#20,STMP4		;DONE ALL LINES
3416	006720	001674			BEQ	25		;BR IF YES
3417	006722	000732			BR	55		;GO CHECK NEXT LINE

3420
(3)
(3)
(2) 006724 000004
(1) 006726 012767 000025 172300

```
*****  
:TEST 25 TEST INTR. ENAB. BITS - INTR. CONDITION DISABLED  
*****  
TST25: SCOPE  
MOV #STN-1,STESTN ;;SET TEST NUMBER IN MAIL BOX
```

3421
(1)
(1)
(1)

```
REM %  
TEST ABSTRACT:  
*****
```

THIS TEST VERIFIES THAT NO TRANSMITTER OR RECEIVER INTERRUPT OCCURS WHEN THE ENABLE BIT IS SET WITH OUT THE INTERRUPTING CONDITION ACTIVE. A BIT MASK (INTMSK: 030100) IS USED TO DEFINE THE I.E. BITS. IN THE "SCR" (BITS 13, 12, AND 06). THE TEST SEQUENCE IS AS FOLLOWS:

1. SET UP THE XMIT AND RCVR VECTORS
2. SELECT AN I.E. BIT TO TEST
3. INIT THE SP AND LOCK OUT INTERRUPTS
4. SET THE SELECTED BIT IN THE "SCR"
5. CLEAR THE PSM TO ALLOW INTR
6. IF NO INTR: REPEAT 2 THRU 5 UNTIL ALL BITS TESTED
7. IF INTR: REPORT ERROR AND CONTINUE WITH NEXT BIT TO TEST

ALL ERRORS ARE REPORTED AND THEN THE TEST RESUMES WITH THE NEXT BIT IN SEQUENCE .

ERRORS:

- 1.) [ERROR 11] IS CALLED TO REPORT RCVR INTR FAULTS
- 2.) [ERROR 12] IS CALLED TO REPORT XMITTR INTR FAULTS

SYNC: M7277 SH3 INIT A H EF2

DEBUG:

- 1.) PROBLEM IS MOST LIKELY THE M7289 MODULE IF THIS IS THE FIRST TEST TO FAIL.
- 2.) SET UP SCOPE ERROR LOOP AND BACKTRACK THROUGH THE LOGIC STARTING WITH THE KEY LOGIC BELOW.

KEY LOGIC:

3422 006734 012767 007012 172146
3423 006742 010102
3424 006744 016703 015432
3425 006750 012723 007042

```

M7289 SH6 XMIT INT REG H FM1
RCV INT REG H DP1
%
MOV #3$,SLPERR ;SET UP THE ERROR LOOP RETURN
MOV R1,R2 ;MAKE IT REGADR TOO
MOV DMVCT,R3 ;GET FIRST VECTOR ADDRESS
MOV #1$, (R3)+ ;GO TO 3$ IF RCVR INTR
```

3426	006754	116723	016160		MOV B	DHRLVL, (R3)+	
3427	006760	105723			TST B	(R3)+	; UPDATE POINTER
3428	006762	012723	007064		MOV	#5\$, (R3)+	; GO TO 5\$ IF XMITTR INTR
3429	006766	116713	016147		MOV B	DHTLVL, (R3)	
3430	006772	012705	000001		MOV	#1, R5	; INIT BIT TEST MARKER
3431	006776	030567	015772	1\$:	BIT	R5, INTMSK	; TEST THIS BIT ??
3432	007002	001003			BNE	3\$; BR IF YES
3433	007004	006305		2\$:	ASL	R5	; SHIFT THE MARKER
3434	007006	001437			BEQ	6\$; BR IF TESTED ALL REQUIRED BITS
3435	007010	000772			BR	1\$; GO TEST FOR THIS ONE
3436							
3437	007012	012706	001100	3\$:	MOV	#STACK, SP	; RESET SP FOR ERROR LOOPING
3438	007016	004767	015276		JSR	PC, CHPS2	; GO LOCK OUT INTR
3439	007022	012711	004000		MOV	#BIT11, (R1)	; CLEAR THE DH11 INTERFACE
3440	007026	010504			MOV	R5, R4	; SET UP S/B DATA
3441	007030	050511			BIS	R5, (R1)	; SET THE TEST I.E. BIT
3442	007032	004767	015246		JSR	PC, CHPS1	; GO CLEAR PSM
3443	007036	000240			NOP		; WAIT A BIT TO ALLOW INTR
3444	007040	000761			BR	2\$; OK - GO DO NEXT I.E. BIT
3445							
3446	007042	004767	015266	4\$:	JSR	PC, SAPS	; SAVE THE ERROR PSM
3447	007046	011103			MOV	(R1), R3	; GET THE WAS DATA
3448	007050	004767	013622		JSR	PC, SUER2A	; GO SET UP ERROR INFO
3449	007054	104011			ERROR	11	; DH11 RCVR SHOULD NOT HAVE INTERRUPTED
3450	007056	012716	007004		MOV	#2\$, (SP)	; SET UP TO RETURN
3451	007062	000002			RTI		; RETURN TO TEST NEXT BIT
3452							
3453	007064	004767	015244	5\$:	JSR	PC, SAPS	; SAVE THE ERROR PSM
3454	007070	011103			MOV	(R1), R3	; GET THE WAS DATA
3455	007072	004767	013600		JSR	PC, SUER2A	; GO SET UP ERROR INFO
3456	007076	104012			ERROR	12	; XMITTER SHOULD NOT HAVE INTERRUPTED
3457	007100	012716	007004		MOV	#2\$, (SP)	; SET UP TO RETURN
3458	007104	000002			RTI		; RETURN TO TEST NEXT BIT
3459							
3460	007106	012706	001100	6\$:	MOV	#STACK, SP	; RESET THE SP JUST IN CASE
3461	007112	004767	015006		JSR	PC, RESTRP	; GO RESTORE TRAP CATCHER IN VECTOR


```

(1)
3466 007126 012767 007164 171754 %      MOV    #15,SLPERR      ;SET UP THE ERROR LOOP RETURN
3467 007134 010102                MOV    R1,R2          ;MAKE IT REGADR TOO
3468 007136 016703 015240                MOV    DHVCT,R3      ;GET FIRST VECTOR ADDR
3469 007142 012723 007274                MOV    #3$, (R3)+    ;GO TO 3$ IF RCVR INTR$
3470 007146 116723 015766                MOVSB  DHALVL, (R3)+
3471 007152 105723                TSTB   (R3)+         ;UPDATE POINTER
3472 007154 012723 007250                MOV    #2$, (R3)+    ;GO TO 3$ ON XMITTR INTR$
3473 007160 116713 015755                MOVSB  DHTLVL, (R3)
3474 007164 012711 004000 1$:      MOV    #BIT11, (R1)   ;CLR THE DM11
3475 007170 012706 001100                MOV    #STACK,SP    ;RESET THE SP FOR ERROR LOOPS
3476 007174 004767 015120                JSR    PC,CHPS2     ;GO LOCK OUT INTR$
3477 007200 012711 001000                MOV    #BIT09, (R1) ;SET MAINT MODE BIT
3478 007234 052711 000100                BIS    #BIT06, (R1) ;SET CHAR AVAILABLE I.E. BIT
3479 007210 052711 000200                BIS    #BIT07, (R1) ;SET THE CHAR AVAIL BIT TO FORCE INTR
3480 007214 004767 015064                JSR    PC,CHPS1     ;GO CLEAR PSW
3481 007220 000240                NOP                 ;GIVE IT A LITTLE TIME
3482
3483 007222 004767 015106                JSR    PC,SAPS      ;SAVE THE ERROR PSW
3484 007226 011103                MOV    (R1),R3      ;GET THE WAS DATA
3485 007230 005011                CLR    (R1)         ;CLEAR OUT THE SCR
3486 007232 005011                CLR    (R1)
3487 007234 012704 001300                MOV    #1300,R4     ;SET UP S/B DATA
3488 007240 004767 013432                JSR    PC,SUER2A    ;GO SET UP ERROR INFO
3489 007244 104013                ERROR  13           ;TIMED OUT AWAITING CHAR AVAIL INTR
3490 007246 000412                BR     3$           ;GO EXIT TEST
3491
3492 007250 004767 015060 2$:      JSR    PC,SAPS      ;SAVE THE ERROR PSW
3493 007254 011103                MOV    (R1),R3      ;GET WAS DATA
3494 007256 012704 001300                MOV    #1300,R4     ;SET UP S/B DATA
3495 007262 005011                CLR    (R1)         ;CLR OUT SCR REG
3496 007264 005011                CLR    (R1)
3497 007266 004767 013404                JSR    PC,SUER2A    ;GO SET UP ERROR INFO
3498 007272 104012                ERROR  12           ;UNEXPECTED XMITTR INTR
3499
3500 007274 012706 001100 3$:      MOV    #STACK,SP    ;RESET THE SP
3501 007300 004767 014620                JSR    PC,RESTRP    ;GO RESTORE TRAP CATCHER

```


3515	007356	012706	001100		MOV	#STACK, SP	; RESET THE SP FOR ERROR LOOPS
3516	007362	004767	014732		JSR	PC, CHPS2	; GO LOCK OUT INTR
3517	007366	012711	001000		MOV	#BIT09, (R1)	; SET MAINT MODE BIT
3518	007372	052711	040000		BIS	#BIT14, (R1)	; SET SILO OVFLW I.E. BIT
3519	007376	052711	010000		BIS	#BIT12, (R1)	; SET THE SILO FULL BIT TO FORCE INTR
3520	007402	004767	014676		JSR	PC, CHPS1	; GO CLEAR PSW
3521	007406	000240			NOP		; GIVE IT A LITTLE TIME
3522	007410	004767	014720		JSR	PC, SAPS	; SAVE THE ERROR PSW
3523	007414	011103			MOV	(R1), R3	; GET THE WAS DATA
3524	007416	005011			CLR	(R1)	; CLEAR OUT THE SCR
3525	007420	005011			CLR	(R1)	
3526	007422	012704	051000		MOV	#51000, R4	; SET UP S/B DATA
3527	007426	004767	013244		JSR	PC, SUER2A	; GO SET UP ERROR INFO
3528	007432	104043			ERROR	43	; TIMED OUT AWAITING SILO OVFLW INTR
3529	007434	000412			BR	3\$; GO EXIT TEST
3530							
3531							
3532	007436	004767	014672	2\$:	JSR	PC, SAPS	; SAVE THE ERROR PSW
3533	007442	011103			MOV	(R1), R3	; GET WAS DATA
3534	007444	012704	051000		MOV	#51000, R4	; SET UP S/B DATA
3535	007450	005011			CLR	(R1)	; CLR OUT SCR REG
3536	007452	005011			CLR	(R1)	
3537	007454	004767	013216		JSR	PC, SUER2A	; GO SET UP ERROR INFO
3538	007460	104012			ERROR	12	; UNEXPECTED XMITTR INTR
3539							
3540	007462	012706	001100	3\$:	MOV	#STACK, SP	; RESET THE SP
3541	007466	004767	014432		JSR	PC, RESTRP	; GO RESTORE TRAP CATCHER

MAINDEC-11-DZDMM-A
DZDMM.A.P11 T30

MACY11 27(663) 12-DEC-75 08:41 PAGE 71
TEST NON EX MEM I.E. WITH INTR. CONDITION ACTIVE

SEQ 0185

3544
(3)
(3)
(2) 007472 000004
(1) 007474 012767 000030 171532
3545
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)

```
*****
;TEST 30 TEST NON EX MEM I.E. WITH INTR. CONDITION ACTIVE
*****
TST30: SCOPE
MOV #STN-1,STESTN ;;SET TEST NUMBER IN MAIL BOX
```

.REM X
TEST ABSTRACT:

THIS TEST VERIFIES THAT THE NON-EX-MEM BIT (SCR10) CAN CAUSE A TRANSMITTER INTERRUPT VIA THE PROPER VECTOR. THE TEST SEQUENCE IS AS FOLLOWS:

1. SET UP XMIT AND RCVR VECTORS
2. CLEAR THE DH11, RESET SP, AND LOCK OUT INTRS
3. PRIME DH11 TO GENERATE XMIT INTR IN MAINT. MODE
4. ALLOW INTRS.
5. REPORT ERROR IF NO XMIT INTR OCCURS OR IF A FALSE RCVR INTR OCCURS
6. REST SP AND VECTORS THEN GO TO TEST 31

ERRORS:

1. [ERROR 44] IS CALLED IF NON-EX-MEM FAILS TO GENERATE XMIT INTR
2. [ERROR 11] IS CALLED IF FALSE RCVR INTR OCCURS

SYNC: M7277 SH3 INIT A H EF2

DEBUG:

1. IF THE NON-EX-MEM INTERRUPT FAILS TO OCCUR PROBLEM IS MOST LIKELY THE M7289 MODULE
2. IF A FALSE RCVR INTR OCCURS PROBLEM IS MOST LIKELY THE M7289 OR THE M7281 MODULES.

KEY LOGIC:

M7289 SH6 SCR 10 H (NO EX MEM) FL1
E35, E41, OR E48

3546 007502 012767 007540 171400
3547 007510 010102
3548 007512 016703 014664
3549 007516 012723 007624
3550 007522 116723 015412
3551 007526 105723
3552 007530 012723 007650
3553 007534 116713 015401
3554 007540 012711 004000

```

%
MOV #15,SLPERR ;SET UP THE ERROR LOOP RETURN
MOV R1,R2 ;MAKE IT REGADR TOO
MOV DHVCT,R3 ;GET FIRST VECTOR ADDR
MOV #25,(R3)+ ;GO TO 25 IF RCVR INTRS
MOVB DMT(LV,(R3)+
TSTB (R3)+ ;UPDATE POINTER
MOV #35,(R3)+ ;GO TO 35 ON XMITTR INTRS
MOVB DMT(LV,(R3)
15: MOV #BIT11,(R1) ;CLR THE DH11
```

3555	007544	012706	001100		MOV	#STACK, SP	:RESET THE SP FOR ERROR LOOPS
3556	007550	004767	014544		JSR	PC, CHPS2	:GO LOCK OUT INTR
3557	007554	012711	001000		MOV	#BIT09, (R1)	:SET MAINT MODE BIT
3558	007560	052711	020000		BIS	#BIT13, (R1)	:SET XMITTR I.E. BIT
3559	007564	052711	002000		BIS	#BIT10, (R1)	:SET THE NON EX MEM BIT TO FORCE INTR
3560	007570	004767	014510		JSR	PC, CHPS1	:GO CLEAR PSW
3561	007574	000240			NOP		:GIVE IT A LITTLE TIME
3562							
3563	007576	004767	014532		JSR	PC, SAPS	:SAVE THE ERROR PSW
3564	007602	011103			MOV	(R1), R3	:GET THE WAS DATA
3565	007604	005011			CLR	(R1)	:CLEAR OUT THE SCR
3566	007606	005011			CLR	(R1)	
3567	007610	012704	023000		MOV	#23000, R4	:SET UP S/B DATA
3568	007614	004767	013056		JSR	PC, SUER2A	:GO SET UP ERROR INFO
3569	007620	104044			ERROR	44	:TIMED OUT AWAITING NON EX MEM INTR
3570	007622	000412			BR	3\$:GC EXIT TEST
3571							
3572	007624	004767	014504	2\$:	JSR	PC, SAPS	:SAVE THE ERROR PSW
3573	007630	011103			MOV	(R1), R3	:GET WAS DATA
3574	007632	012704	023000		MOV	#23000, R4	:SET UP S/B DATA
3575	007636	005011			CLR	(R1)	:CLR OUT SCR REG
3576	007640	005011			CLR	(R1)	
3577	007642	004767	013030		JSR	PC, SUER2A	:GO SET UP ERROR INFO
3578	007646	104011			ERROR	11	:UNEXPECTED RCVR INTR
3579							
3580	007650	012706	001100	3\$:	MOV	#STACK, SP	:RESET THE SP
3581	007654	004767	014244		JSR	PC, RESTRP	:GO RESTORE TRAP CATCHER

H15

MAINDEC-11-DZDMM-A
DZDMM.A.P11 T31

MACY11 27(663) 12-DEC-75 08:41 PAGE 72-1
TEST XMITTR DONE I.E. WITH INTR. CONDITION ACTIVE

SEQ 0188

3597	007742	012711	001000		MOV	#BIT09,(R1)	:SET MAINT MODE BIT
3598	007746	052711	020000		BIS	#BIT13,(R1)	:SET XMIT DONE I.E. BIT
3599	007752	052711	100000		BIS	#BIT15,(R1)	:SET THE XMITTR DONE BIT TO FORCE INTR
3600	007756	004767	014322		JSR	PC,CHPS1	:GO CLEAR PSW
3601	007762	000240			NOP		:GIVE IT A LITTLE TIME
3602							
3603	007764	004767	014344		JSR	PC,SAPS	:SAVE THE ERROR PSW
3604	007770	011103			MOV	(R1),R3	:GET THE WAS DATA
3605	007772	005011			CLR	(R1)	:CLEAR OUT THE SCR
3606	007774	005011			CLR	(R1)	
3607	007776	012704	121000		MOV	#121000,R4	:SET UP S/B DATA
3608	010002	004767	012670		JSR	PC,SUER2A	:GO SET UP ERROR INFO
3609	010006	104045			ERROR	45	:TIMED OUT AWAITING XMIT DONE INTR
3610	010010	000412			BR	35	:GO EXIT TEST
3611							
3612	010012	004767	014316	25:	JSR	PC,SAPS	:SAVE THE ERROR PSW
3613	010016	011103			MOV	(R1),R3	:GET WAS DATA
3614	010020	012704	121000		MOV	#121000,R4	:SET UP S/B DATA
3615	010024	005011			CLR	(R1)	:CLR OUT SCR REG
3616	010026	005011			CLR	(R1)	
3617	010030	004767	012642		JSR	PC,SUER2A	:GO SET UP ERROR INFO
3618	010034	104011			ERROR	11	:UNEXPECTED RCVR INTR
3619							
3620	010036	012706	001100	35:	MOV	#STACK,SP	:RESET THE SP
3621	010042	004767	014056		JSR	PC,RESTRP	:GO RESTORE TRAP CATCHER

3640	010140	012711	004000		MOV	#BIT11,(R1)	;	CLEAR THE DM11 INTERFACE
3641	010144	156711	014774		BISB	LINE,(R1)	;	SELECT A LINE NO.
3642	010150	012761	177777	000010	MOV	#-1,BCR(R1)	;	SET BYTE COUNT TO -1
3643	010156	012761	033500	000004	MOV	#33500,LPR(R1)	;	SET UP LINE PARAMETERS
3644	010164	056761	014222	000012	BIS	LINMSK,BAR(R1)	;	ACTIVATE SELECTED LINE
3645	010172	052711	020000		BIS	#BIT13,(R1)	;	ENABLE INTERRUPT ON XMIT DONE
3646	010176	004767	014102		JSR	PC,CHPS1	;	GO CLEAR PSW
3647								
3648	010203	012767	000001	014762	MOV	#1,TIMEA	;	INIT TIMER A
3649	010210	005067	014760		CLR	TIMEB	;	INIT TIMER B
3650	010214	000240		36:	NOP		;	DO NOTHING WAIT
3651	010216	004767	013730		JSR	PC,TIMEIT	;	CALL TIMER
3652	010222	000774			BR	36	;	TIMER ROUTINE WILL MOVE RETURN PC AROUND THIS BRANCH IF TIMEOUT OCCURS
3653								
3654								
3655	010224	004767	014104		JSR	PC,SAPS	;	SAVE THE ERROR PSW
3656	010230	011103			MOV	(R1),R3	;	GET THE WAS DATA
3657	010232	042703	000200		BIC	#BIT07,R3	;	WE'RE NOT INTERESTED IN THIS BIT
3658	010236	004767	012434		JSR	PC,SUER2A	;	GO SET UP ERROR INFO
3659	010242	004557	012650		JSR	RS,SUNUM	;	GO SET LINE NO. IN ERROR MSG
3660	010246	025144			LINE			
3661	010250	026344			EM15+43			
3662	010252	104015			ERROR	15	;	TIMEOUT WHILE AWAITING XMIT INTR
3663	010254	000713			BR	15	;	GO TEST NEXT LINE
3664								
3665	010256	005711		45:	TST	(R1)	;	DID XMIT DONE SET ??
3666	010260	100411			BMI	55	;	BR IF YES
3667								
3668	010262	004767	014046		JSR	PC,SAPS	;	SAVE THE ERROR PSW
3669	010266	011103			MOV	(R1),R3	;	GET THE WAS DATA
3670	010270	004767	012402		JSR	PC,SUER2A	;	GO SET UP ERROR INFO
3671	010274	004767	000174		JSR	PC,95	;	GO SET UP SOME ERROR STUFF
3672	010300	104014			ERROR	14	;	XMIT DONE FAILED TO SET
3673	010302	000700			BR	15	;	GO TEST NEXT LINE
3674								
3675	010304	016103	000012	55:	MOV	BAR(R1),R3	;	GET WAS DATA FROM "BAR"
3676	010310	001413			BEQ	65	;	BR IF BAR BIT GOT CLEARED
3677								
3678	010312	004767	014016		JSR	PC,SAPS	;	SAVE THE ERROR PSW
3679	010316	062702	000012		ADD	#BAR,R2	;	SET UP REGADR
3680	010322	005004			CLR	R4	;	SET UP S/B DATA
3681	010324	004767	012346		JSR	PC,SUER2A	;	GO SET UP ERROR INFO
3682	010330	004767	000140		JSR	PC,95	;	GO SET UP SOME ERROR STUFF
3683	010334	104014			ERROR	14	;	BAR BIT FAILED TO CLEAR
3684	010336	000662			BR	15	;	GO TEST NEXT LINE
3685								
3686	010340	016103	000006	65:	MOV	CAR(R1),R3	;	GET THE WAS DATA FROM CAR
3687	010344	022703	000001		CMP	#1,R3	;	DID IT GET INCREMENTED ?
3688	010350	001414			BEQ	75	;	BR IF YES
3689								
3690	010352	004767	013756		JSR	PC,SAPS	;	SAVE THE ERROR PSW
3691	010356	012704	000001		MOV	#1,R4	;	SET UP S/B DATA
3692	010362	062702	000006		ADD	#CAR,R2	;	SET UP REGADR
3693	010366	004767	012304		JSR	PC,SUER2A	;	GO SET UP ERROR INFO

3694	010372	004767	000076		JSR	PC, 95	; GO SET UP SOME ERROR STUFF
3695	010376	104014			ERROR	14	; CAR REG NOT INCREMENTED PROPERLY
3696	010400	000641			BR	15	; GO TEST NEXT LINE
3697							
3698	010402	016103	000010	75:	MOV	BCR(R1), R3	; GET MSG DATA FROM BCR
3699	010406	001636			BEQ	15	; BR IF BCR GOT INCREMENTED TO 000000
3700							
3701	010410	004767	013720		JSR	PC, SAPS	; SAVE THE ERROR PSW
3702	010414	005004			CLR	R4	; SET UP S/B DATA
3703	010416	062702	000010		ADD	#BCR, R2	; SET UP REGADR
3704	010422	004767	012250		JSR	PC, SUER2A	; GO SET UP ERROR INFO
3705	010426	004767	000042		JSR	PC, 95	; GO SET UP SOME ERROR STUFF
3706	010432	104014			ERROR	14	; BCR REG NOT INCREMENTED PROPERLY
3707	010434	000623			BR	15	; GO TEST NEXT LINE
3708							
3709	010436	012711	004000	85:	MOV	#BIT11, (R1)	; CLEAR THE DH11
3710	010442	016703	013734		MOV	DHVCT, R3	; GET THE VECTOR ADDR
3711	010446	062703	000004		ADD	#4, R3	; POINT TO XMT VECTOR
3712	010452	010313			MOV	R3, (R3)	; RESTORE TRAP CATCHER
3713	010454	062723	000002		ADD	#2, (R3)+	
3714	010460	005013			CLR	(R3)	
3715	010462	004767	013616		JSR	PC, CHPS1	; GO CLEAR PSW
3716	010466	012706	001100		MOV	#STACK, SP	; RESET THE STACK POINTER
3717	010472	000405			BR	TST33	; GO TO NEXT TEST
3718							
3719	010474	004567	012416	95:	JSR	R5, SUNUM	; GO SET UP LINE NO. IN MSG.
3720	010500	025144			LINE		
3721	010502	026276			EM14+44		
3722	010504	000207			RTS	PC	; RETURN TO REPORT ERROR

3730	010532	052711	004000		2S:	BIS	#BIT11,(R1)	;CLEAR THE DH11
3731	010536	004767	012434			JSR	PC,CLCABC	;GO CLEAR "CAR" AND "BCR" MEMORIES
3732	010542	004767	012472			JSR	PC,LDBCR	;GO LOAD "BCR" MEMORY WITH ALL CNES
3733	010546	156711	014372			BISB	LINE,(R1)	;SELECT THE LINE
3734	010552	012761	033500	000004		MOV	#33500,LPR(R1)	;SET UP PARAMETERS
3735	010560	016761	013626	000012		MOV	LINMSK,BAR(R1)	;ACTIVATE XMIT ON SELECTED LINE
3736								
3737	010566	012767	000001	014376		MOV	#1,TIMEA	;INIT TIMER A
3738	010574	005067	014374			CLR	TIMEB	;INIT TIMER B
3739	010600	005711			3S:	TST	(R1)	;XMITTR DONE YET
3740	010602	100423				BMI	4S	;BR IF YES
3741	010604	004767	013342			JSR	PC,TIMEIT	;CALL THE TIMER
3742	010610	000773				BR	3S	;TIMER ROUTINE WILL MOVE RETURN PC
3743								;AROUND THIS BRANCH IF TIME OUT OCCURS
3744								
3745	010612	004767	013516			JSR	PC,SAPS	;SAVE THE ERROR PSM
3746	010616	011103				MOV	(R1),R3	;GET THE WAS DATA
3747	010620	012704	100000			MOV	#BIT15,R4	;SET UP S/B DATA
3748	010624	156704	014314			BISB	LINE,R4	
3749	010630	010102				MOV	R1,R2	;MAKE REGADR = DEVADR
3750	010632	004767	012040			JSR	PC,SUER2A	;GO SET UP ERROR INFO
3751	010636	004567	012254			JSR	R5,SUNUM	;SET LINE NO. IN MSG
3752	010642	025144				LINE		
3753	010644	031076				EM50+53		
3754	010646	104050				ERROR	50	;TIMED OUT AWAITING XMIT DONE ON SEL LINE
3755	010650	000725				BR	1S	;GO TRY THE NEXT LINE
3756								
3757	010652	005067	170340		4S:	CLR	STMP7	;INIT A LINE COUNTER
3758	010656	116711	170334		5S:	MOVB	STMP7,(R1)	;SELECT LINE NO. IN "SCR"
3759	010662	012704	177777			MOV	#-1,R4	;SET UP S/B DATA
3760	010666	016103	000010			MOV	BCR(R1),R3	;GET THE WAS BYTE COUNT
3761	010672	126767	014246	170316		CMPB	LINE,STMP7	;WAS THIS THE ACTIVE LINE ??
3762	010700	001001				BNE	6S	;BR IF NOT
3763	010702	005004				CLR	R4	;CHANGE S/B DATA TO 000000
3764	010704	020304			6S:	CMP	R3,R4	;WAS BYTE COUNT CORRECT ??
3765	010706	001416				BEQ	7S	;BR IF YES
3766								
3767	010710	005067	170264			CLR	STMP0	;SAVE THE ACTIVE LINE NO.
3768	010714	116767	014224	170256		MOVB	LINE,STMP0	
3769	010722	016767	170270	170252		MOV	STMP7,STMP1	;SAVE THE LINE NO. BEING CHECKED
3770	010730	010102				MOV	R1,R2	;SET UP REGADR = BCR REG ADDR
3771	010732	062702	000010			ADD	#BCR,R2	
3772	010736	004767	012016			JSR	PC,SUER4	;GO SET UP ERROR INFO
3773	010742	104051				ERROR	51	;BYTE COUNT INCORRECT
3774								
3775	010744	005004			7S:	CLR	R4	;SET UP S/B DATA
3776	010746	016103	000006			MOV	CAR(R1),R3	;GET THE WAS DATA
3777	010752	126767	014166	170236		CMPB	LINE,STMP7	;IS THIS THE ACTIVE LINE
3778	010760	001001				BNE	8S	;BR IF NOT
3779	010762	005204				INC	R4	;BUMP THE CAR ADDRESS FOR ACTIVE LINE
3780	010764	020304			8S:	CMP	R3,R4	;CAR CONTENTS CORRECT ??
3781	010766	001416				BEQ	9S	;BR IF YES
3782								
3783	010770	005067	170204			CLR	STMP0	;SET UP ACT LINE NO.

3784	010774	116767	014144	170176		MOVB	LINE, STMP0	
3785	011002	016767	170210	170172		MOV	STMP7, STMP1	; SAVE THE LINE NO. BEING CHECKED
3786	011010	010102				MOV	R1, R2	; SET UP REGADR
3787	011012	062702	000006			ADD	#CAR, R2	
3788	011016	004767	011736			JSR	PC, SUER4	; SET UP THE ERROR INFO
3789	011022	104051				ERROR	S1	; CAR REG INCORRECT
3790								
3791	011024	005267	170166		9S:	INC	STMP7	; GENERATE NEW LINE NO.
3792	011030	022767	000020	170160		CMP	#20, STMP7	; TESTED ALL LINES
3793	011036	001307				BNE	SS	; BR IF NOT
3794	011040	000631				BR	IS	; GO SELECT NEXT ACTIVE LINE

3797
(3)
(3)
(2) 011042 000004
(1) 011044 012767 000034 170162
3798
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)

: TEST 34 TEST THAT CHARACTER AVAILABLE CAN CAUSE RCVR INTERRUPT
: *****
TST34: SCOPE
MOV #STN-1,STESTN ;;SET TEST NUMBER IN MAIL BOX

.REM %
TEST ABSTRACT:

THIS TEST VERIFIES THAT WHEN "CHAR AVAIL" (BIT07 IN "SCR") SETS
AS A RESULT OF XMITTING AND RECEIVING ONE CHARACTER (USING SILO MAINT MODE)
IT CAUSES A RCVR INTR VIA THE PROPER VECTOR. THE TEST SEQUENCE IS AS
FOLLOWS:

1. SET UP THE RCVR VECTOR
2. LOCK OUT INTRs, RESET SP, AND CLEAR DH11
3. USE MAINT MODE TO LOAD A CHAR INTO THE SILO (DATA=125252)
4. CLEAR PSM TO ALLOW INTERRUPTS
5. ACTIVATE TIMER TO WAIT FOR INTR TO OCCUR
6. IF NO RCVR INTR OCCURS REPORT ERROR AND GO TO STEP 9
7. WHEN RCVR INTRs - CHECK SILO DATA FOR 125252 - IF NOT
CORRECT REPORT ERROR AND GO TO STEP 9
8. CHECK THAT SILO FILL LEVEL=1 - IF NOT REPORT ERROR
9. RESET SP, CLEAR PSM, RESET VECTOR, AND GO TO TEST 35

ERRORS:

1. [ERROR 13] IS CALLED TO REPORT RCVR TIMEOUT ERROR
2. [ERROR 52] IS CALLED TO REPORT SILO DATA INCORRECT
3. [ERROR 6] IS CALLED TO REPORT INCORRECT SILO FILL COUNT

SYNC: M7277 SH3 INIT A H EF2

DEBUG:

1. IF NOT RCVR INTR OCCURS SUSPECT THE M7277, M7281, OR M7279 MODULES
2. IF SILO DATA OR FILL-ERRORS SUSPECT THE M7279 MODULE

KEY LOGIC:

M7279	SH1	SILO DATA MUX'S (74157'S)	
		SSR15 H	CR1
	SH2	DATA READY L	OV1
		NRC 15 H	DL1
		SILO MEMORY	(E13,E17,E8,E3) [3341'S]
		LOAD SILO L	DJ2
		5.068 MHZ (CLOCK)	DN1
		SSR <13:00> H	(E20,E24)

```

(1)
(1)
(1)
(1)
(1)
(1)
(1)
3799 011052 012767 011074 170030      MOV      #15,SLPERR      ;SET UP THE ERROR LOOP RETURN
3800 011060 016703 013316      MOV      DHVCT,R3      ;GET FIRST VECTOR ADDR
3801 011064 012723 011160      MOV      #35,(R3)+     ;GO TO 3$ ON RCVR INTERRUPT
3802 011070 116713 014044      MOV      DHRLVL,(R3)
3803 011074 004767 013220      JSR      PC,CHPS2      ;GO LOCK OUT INTR
3804 011100 012706 001100      MOV      #STACK,SP    ;RESET SP FOR ERROR LOOPS
3805 011104 012711 004000      MOV      #BIT11,(R1)   ;CLEAR THE DH11
3806 011110 052761 100000 000016      BIS      #BIT15,SSR(R1);SET SILO MAINT. BIT TO LOAD SILO
3807 011116 012711 000100      MOV      #BIT06,(R1)  ;ENABLE CHAR. AVAIL INTERRUPT
3808 011122 004767 013156      JSR      PC,CHPS1      ;GO CLEAR PSM
3809 011126 012703 001000      MOV      #1000,R3     ;INIT TIMER
3810 011132 005303      DEC      R3           ;DEC TIMER
3811 011134 001376      BNE      2$          ;BR IF NO TIMEOUT
3812
3813 011136 004767 013172      JSR      PC,SAPS      ;SAVE THE ERROR PSM
3814 011142 011103      MOV      (R1),R3      ;GET THE WAS DATA
3815 011144 012704 000300      MOV      #300,R4      ;SET UP S/B DATA
3816 011150 004767 011522      JSR      PC,SUER2A    ;GO SET UP ERROR INFO
3817 011154 104013      ERROR   13           ;CHAR AVAIL FAILED TO SET ON TIME
3818 011156 000436      BR       5$          ;ESCAPE FROM THIS TEST - CATASTROPHIC ERROR
3819
3820 011160 016105 000016      3$:      MOV      SSR(R1),R5   ;SAVE THE SILO STATUS REG.
3821 011164 016103 000002      MOV      NRC(R1),R3   ;GET THE WAS DATA
3822 011170 012704 125252      MOV      #125252,R4  ;SET UP S/B DATA
3823 011174 020304      CMP     R3,R4        ;WAS = S/B = 125252 ??
3824 011176 001410      BEQ     4$          ;BR IF IT IS
3825
3826 011200 004767 013130      JSR      PC,SAPS      ;SAVE THE ERROR PSM
3827 011204 062702 000002      ADD     #NRC,R2      ;SET UP REGADR
3828 011210 004767 011462      JSR      PC,SUER2A    ;GO SET UP ERROR INFO
3829 011214 104052      ERROR   52           ;DATA COMPARE ERROR
3830 011216 000416      BR       5$          ;GET OUT
3831
3832 011220 010503      4$:      MOV      R5,R3        ;NOW GET THE SILO STATUS REG AGAIN
3833 011222 042703 140377      BIC     #140377,R3   ;CLR OUT JUNK
3834 011226 012704 000400      MOV     #400,R4      ;SET UP S/B DATA
3835 011232 020304      CMP     R3,R4        ;SSR CHAR COUNT = 1 ??
3836 011234 001407      BEQ     5$          ;BR IF IT IS
3837
3838 011236 004767 013072      JSR      PC,SAPS      ;SAVE THE ERROR PSM
3839 011242 062702 000016      ADD     #SSR,R2      ;SET UP REGADR
3840 011246 004767 011424      JSR      PC,SUER2A    ;SET UP ERROR INFO
3841 011252 104006      ERROR   6           ;SSR COUNT NOT CORRECT
3842
3843 011254 012706 001100      5$:      MOV      #STACK,SP   ;RESET THE STACK POINTER
3844 011260 004767 013020      JSR      PC,CHPS1    ;GO CLEAR PSM
3845 011264 005011      CLR     (R1)         ;RESET I.E. BIT

```


E16

MAINDEC-11-DZDMM-A
DZDMM.A.P11 T34

MACY11 27(663) 12-DEC-75 08:41 PAGE 75-2
TEST THAT CHARACTER AVAILABLE CAN CAUSE RCVR INTERRUPT

SEQ 0198

3846	011266	016703	013110	MOV	DHVCT,R3	;GET FIRST VECTOR ADDR
3847	011272	010313		MOV	R3,(R3)	;RESTORE TRAP CATCHER
3848	011274	062723	000002	ADD	#2,(R3)+	
3849	011300	005013		CLR	(R3)	
3850						

3865	011354	042712	100000		BIC	#BIT15,(R2)	:CLEAR SILO MAINT. BIT
3866	011360	005204			INC	R4	:COUNT A CHAR LOADED
3867	011362	005305			DEC	R5	:DECREMENT TEST COUNT
3868	011364	001365			BNE	25	:BR UNTIL WE'VE LOADED THE TEST COUNT
3869							
3870	011366	011203			MOV	(R2),R3	:SET THE WAS COUNT
3871	011370	042703	140377		BIC	#140377,R3	:CLR JUNK BITS
3872	011374	000304			SWAB	R4	:SET UP S/B DATA
3873	011376	020304			CMP	R3,R4	:TEST COUNT = SILO COUNTER ?
3874	011400	001406			BEG	45	:BR IF YES
3875							
3876	011402	004767	011264		PC,SUER2		:GO SET UP ERROR INFO
3877	011406	012767	011326	167474	MOV	#15,SLPERR	:SET UP ERROR LOOP RETURN
3878	011414	104006			ERROR	6	:SSR FAILED TO UP-COUNT CORRECTLY
3879							
3880	011416	005267	167574		INC	\$TMP7	:INCREMENT TO NEXT COUNT TO TEST
3881	011422	022767	000100	167566	CMP	#100,\$TMP7	:MAXIMUM COUNT ??
3882	011430	001336			BNE	15	:BR IF NOT
3883							

3886
(3)
(3)
(2) 01:432 000004
(1) 011434 012767 000036 167572

:TEST 36 TEST THAT SILO STATUS REGISTER DOWN COUNTS CORRECTLY

TST36: SCOPE
MOV #STN-1,STESTN ;SET TEST NUMBER IN MAIL BOX

3887
(1)
(1)
(1)

REM X
TEST ABSTRACT:

THIS TEST VERIFIES THAT THE SILO FILL LEVEL COUNTS DOWN PROPERLY
WHEN WORDS ARE READ FROM THE SILO. ALL COUNTS FROM 77-00 ARE
TESTED. THE TEST SEQUENCE IS AS FOLLOWS:

1. INIT "STMP7" TO START WITH A COUNT OF 1
2. CLEAR THE DH11 AND FILL SILO WITH 64. WORDS
3. READ THE NO. OF WORDS SPECIFIED BY COUNT
4. CHECK THAT FILL LEVEL=64. MINUS (COUNT) - REPORT ERRORS
5. INCREMENT "STMP7" AND REPEAT 2 THRU 4 UNTIL ALL COUNTS TESTED.

ERRORS:

1. [ERROR 6] IS CALLED TO REPORT SILO FILL LEVEL ERRORS

SYNC: M7277 SH3 INIT A H EF2

DEBUG: (REFER TO TEST 35)

KEY LOGIC: (REFER TO TEST 35)

3888 011442 010102
3889 011444 062702 000016
3890 011450 012767 000001 167540
3891 011456 012711 004000
3892 011462 012705 000100
3893 011466 166705 167524
3894 011472 012703 000100
3895 011476 012704 001000
3896 011502 052712 100000
3897 011506 005304
3898 011510 001376
3899 011512 042712 100000
3900 011516 005303
3901 011520 001366
3902
3903 011522 016703 167470
3904 011526 012704 001000
3905 0:1532 005761 000002
3906 0:1536 005304

```

%
MOV R1,R2 ;SET UP REGADR
ADD #SSR,R2
MOV #1,STMP7 ;START WITH COUNT = 1
1$: MOV #BIT11,(R1) ;CLR THE DH11
MOV #100,R5 ;TEST COUNT SHOULD BE 64(10) MINUS
SUB STMP7,R5 ;THE NO. OF CHARS READ
MOV #100,R3 ;COUNTER USED TO FILL SILO
2$: MOV #1000,R4 ;INIT TIMER
BIS #BIT15,(R2) ;SET SILO MAINT. BIT
3$: DEC R4 ;STALL TO ALLOW SILO TO LOAD
BNE 3$
BIC #BIT15,(R2) ;CLEAR THE SILO MAINT BIT
DEC R3 ;COUNT ONE CHAR LOADED
BNE 2$ ;BR UNTIL ALL LOADED

MOV STMP7,R3 ;INIT COUNTER FOR READING SILO
4$: MOV #1000,R4 ;INIT TIMER
TST NRC(R1) ;READ THE SILO
5$: DEC R4 ;GIVE IT TIME TO SETTLE

```


3935	011644	012711	004000	1S:	MOV	#BIT11,(R1)	:CLEAR THE DH11
3936	011650	016705	167342		MOV	STMP7,R5	:SAVE IT IN R5
3937	011654	010512			MOV	R5,(R2)	:SET ALARM LEVEL IN SSR
3938	011656	005205			INC	R5	:LOAD ONE MORE THAN FILL LEVEL
3939	011660	052712	100000	2S:	BIS	#BIT15,(R2)	:SET SILO MAINT. TO LOAD A CHAR
3940	011664	012703	001000		MOV	#1000,R3	:INIT STALL TIMER
3941	011670	005303		3S:	DEC	R3	:WAIT FOR SILO TO SETTLE
3942	011672	001376			BNE	3S	:BR TIL R3 GOES TO 000000
3943	011674	042712	100000		BIC	#BIT15,(R2)	:CLR THE SILO MAINT BIT
3944	011700	005305			DEC	R5	:COUNT ONE LOADED
3945	011702	105711			TSTB	(R1)	:CHAR AVAIL SET YET
3946	011704	100412			BMI	4S	:BR IF IT IS
3947	011706	005705			TST	R5	:SHOULD IT BE ??
3948	011710	001363			BNE	2S	:BR IF NOT
3949							
3950	011712	004767	012416		JSR	PC,SAPS	:SAVE THE ERROR PSW
3951	011716	004767	000042		JSR	PC,5S	:GO SET UP S/B DATA
3952	011722	004767	010750		JSR	PC,SUER2A	:GO SET UP ERROR INFO
3953	011726	104006			ERROR	6	:SILO ALARM LEVEL FAILED AT SELECTED COUNT
3954	011730	000426			BR	6S	:GO CHECK NEXT COUNT
3955							
3956	011732	005705		4S:	TST	R5	:SHOULD IT HAVE BEEN SET (CHAR AVAIL)
3957	011734	001424			BEQ	6S	:BR IF YES
3958							
3959	011736	004767	012372		JSR	PC,SAPS	:SAVE THE ERROR PSW
3960	011742	022705	000001		CMR	#1,R5	:IS IT OFF BY ONLY ONE ??
3961	011746	001417			BEQ	6S	:BR IF YES - WE'LL ALLOW HIM THIS
3962	011750	004767	000010		JSR	PC,5S	:GO SET UP S/B DATA
3963	011754	004767	010716		JSR	PC,SUER2A	:GO SET UP ERROR INFO
3964	011760	104006			ERROR	6	:SILO ALARM LEVEL FAILED
3965	011762	000411			BR	6S	:GO CHECK NEXT COUNT
3966							
3967	011764	011203		5S:	MOV	(R2),R3	:GET HAS DATA
3968	011766	016704	167224		MOV	STMP7,R4	:SET UP THE S/B DATA
3969	011772	005204			INC	R4	
3970	011774	000304			SWAB	R4	
3971	011776	105004			CLRB	R4	
3972	012000	156704	167212		BISB	STMP7,R4	
3973	012004	000207			RTS	PC	:RETURN TO SET UP AND REPORT ERROR
3974							
3975	012006	005767	167204	6S:	TST	STMP7	:COUNT AT ZERO
3976	012012	001002			BNE	7S	:BR IF NOT
3977	012014	000261			SEC		:SET THE "C" BIT
3978	012016	000401			BR	8S	:GO SET UP COUNT
3979	012020	000241		7S:	CLC		:CLEAR THE "C" BIT
3980	012022	006167	167170	8S:	ROL	STMP7	:SHIFT POWER OF TWO BIT
3981	012026	032767	000100		BIT	#BIT6,STMP7	:DONE ALL POWERS ??
3982	012034	001703			BEQ	1S	:BR IF NOT

M16

MAINDEC-11-DZDMM-A
DZDMM.P11 T40

MACY11 27(663) 12-DEC-75 08:41 PAGE 79-1
VERIFY STORAGE OVERFLOW - NON MAINT. MODE - ALL LINES

SEQ 0206

```

(1)
3987 012046 012767 012062 167034 X      MOV      #25,SLPERR      ;SET UP ERROR RETURN
3988 012054 004767 010744 1S:      JSR      PC,SELIN     ;GO SELECT A LINE NO. TO TEST
3989 012060 000535          BR        TST41       ;BR IF DONE ALL SELECTED LINES
3990 012062 012711 004000 2S:      MOV      #BIT11,(R1)  ;CLEAR THE DM11
3991 012066 010102          MOV      R1,R2       ;MAKE REGADR = DEVAR
3992 012070 116711 013050          MOV      LINE,(R1)   ;SELECT A LINE
3993 012074 005061 000006          CLR      CAR(R1)    ;SET UP CURRENT ADDRESS
3994 012100 012761 177677 J00010      MOV      #65,BCR(R1) ;SET UP BYTE COUNT
3995 012106 012761 033500 000004      MOV      #33500,LPR(R1);SET UP LPR: 9600 BAUD,5 BIT CHARS
3996 012114 016761 012272 000012      MOV      LINMSK,BAR(R1);ACTIVATE THE SELECTED LINE
3997
3998 012122 012767 000001 013042          MOV      #1,TIMEA    ;INIT TIMERS
3999 012130 005067 013040          CLR      TIMEB
4000 012134 032711 040000 3S:      BIT      #BIT.4,(R1) ;STORAGE OVERFLOW YET ??
4001 012140 001024          BNE     4S          ;BR IF YES YOU SHOULD GET IT
4002 012142 004767 012004          JSR      PC,TIMEIT   ;CALL TIMER
4003 012146 000772          BR        3S        ;BR IF NO TIME OUT
4004
4005 012150 004767 012160          JSR      PC,SAPS     ;GO SAVE PSM
4006 012154 012704 040000          MOV      #BIT14,R4  ;SET UP S/B DATA
4007 012160 156704 012760          BISB    LINE,R4
4008 012164 011103          MOV      (R1),R3    ;SET UP WAS DATA
4009 012166 042703 137760          BIC     #137760,R3  ;CLEAR UNINTERESTING BITS
4010 012172 004767 010500          JSR      PC,SUER2A   ;GO SET UP ERROR INFO
4011 012176 004567 010714          JSR      RS,SUNUM    ;PUT LINE NO. IN MESSAGE
4012 012202 025144          LINE
4013 012204 031563          EMS7+44
4014 012206 104057          ERROR  57
4015 012210 000721          BR        1S        ;STORAGE OVERFLOW FAILED TO SET
4016                                     ;GO TRY NEXT LINE
4017 012212 016167 000002 166776 4S:      MOV      NRC(R1),STMP7 ;READ THE SILO
4018 012220 016167 000002 166770          MOV      NRC(R1),STMP7 ;READ IT AGAIN
4019 012226 012705 001000          MOV      #1000,R5   ;INIT STALL COUNTER
4020 012232 005305          DEC     R5          ;COUNT TIMER
4021 012234 001376          BNE     41S        ;BR IF NO TIMEOUT
4022 012236 032711 040000          BIT      #BIT14,(R1) ;DID OVERFLOW GO AWAY ?
4023 012242 001420          BEQ     5S        ;BR IF YES
4024
4025 012244 004767 012064          JSR      PC,SAPS     ;GO SAVE THE PSM
4026 012250 005004          CLR      R4        ;SET UP S/B DATA
4027 012252 156704 012666          BISB    LINE,R4
4028 012256 011103          MOV      (R1),R3    ;SET UP WAS DATA
4029 012260 042703 137760          BIC     #137760,R3  ;CLEAR UNINTERESTING BITS
4030 012264 004767 010406          JSR      PC,SUER2A   ;GO SET UP ERROR INFO
4031 012270 004567 010622          JSR      RS,SUNUM    ;PUT LINE NO. IN MSG
4032 012274 025144          LINE
4033 012276 031563          EMS7+44
4034 012300 104057          ERROR  57
4035 012302 000664          BR        1S        ;STORAGE BIT FAILED TO CLEAR
4036                                     ;GO TRY NEXT LINE
4037 012304 122761 000077 000017 5S:      CMPB    #77,SSR+1(R1) ;WAS IT REALLY 65. ??
4038 012312 001660          BEQ     1S
4039

```

4040	012314	004767	012014
4041	012320	012704	037400
4042	012324	062702	000016
4043	012330	016103	000016
4044	012334	004767	010336
4045	012340	004567	010552
4046	012344	025144	
4047	012346	031563	
4048	012350	104057	
4050	012352	000640	

```

JSR PC,SAPS
MOV #37400,R4
ADD #SSR,R2
MOV SSR(R1),R3
JSR PC,SUER2A
JSR RS,SUNUM
LINE
EM57+44
ERROR 57
BR 15

```

```

:GO SAVE PSW
:SET UP S/B DATA
:SET UP REGADR
:SAVE WAS DATA
:GO SET UP ERROR INFO
:PUT LINE NO. IN MSG

:READING SILO FAILED TO DEC SSR OR
:STORAGE OVFL SET AT WRONG COUNT
:GO TRY NEXT LINE

```


TX CLOCK LINE "N" SIGNALS ON UART PIN 40

Line No.	Address 1	Address 2	Address 3	Address 4	Label	Operation	Comment
(1)							
4055	012364	012767	012414	166516	%	MOV #25,SLPERR	:SET UP ERROR LOOP RETURN
4056	012372	004767	010426		15:	JSR PC,SELIN	:GO SELECT A LINE TO TEST
4057	012376	000534				BR TST42	:BR IF TESTED ALL SELECTED LINES
4058	012400	012767	002100	166610		MOV #2100,STMP7	:INIT T1 START WITH LOWEST SPEED
4059	012406	012767	177777	012562		MOV #-1,TIMEC	:INIT RELATIVE TIME CHECKER
4060	012414	012711	004000		25:	MOV #BIT11,(R1)	:CLEAR THE DM11
4061	012420	156711	012520		35:	BISB LINE,(R1)	:SELECT IT IN THE SCR
4062	012424	012761	177775	000010		MOV #-3,BCR(R1)	:SET BYTE COUNT TO XFER 3 CHARS
4063	012432	005061	000006			CLR CAR(R1)	:GET TEST DATA STARTING AT LOC. 0
4064	012436	016761	166554	000004		MOV STMP7,LPR(R1)	:SELECT A XMIT SPEED
4065	012444	016761	011742	000012		MOV LINASK,BAR(R1)	:ACTIVATE THE TRANSMITTER
4066							
4067	012452	012767	000001	012512		MOV #1,TIMEA	:INIT TIMER A
4068	012460	005067	012510			CLR TIMEB	:INIT TIMER B
4069	012464	005711			45:	TST (R1)	:XMITTR DONE SET YET ?
4070	012466	100437				BMI 55	:BR IF YES
4071	012470	004767	011456			JSR PC,TIMEIT	:CALL THE TIMER
4072	012474	000773				BR 45	:TIMER ROUTINE WILL MOVE RETURN PC
4073							:AROUND THIS BRANCH IF TIME OUT OCCURS
4074							
4075	012476	016767	166514	166474		MOV STMP7,STMP0	:SAVE AND SET UP THE SPEED CODE
4076	012504	000367	166470			SWAB STMP0	
4077	012510	006267	166464			ASR STMP0	
4078	012514	006267	166460			ASR STMP0	
4079	012520	042767	177760	166452		BIC #177760,STMP0	
4080	012526	011103				MOV (R1),R3	:GET THE WAS DATA
4081	012530	042703	000200			BIC #BIT07,R3	:CLEAR UNINTERESTING BITS
4082	012534	010102				MOV R1,R2	:MAKE REGADR = DEVADR
4083	012536	012704	100000			MOV #BIT15,R4	:SET UP S/B DATA
4084	012542	156704	012376			BISB LINE,R4	
4085	012546	004767	010124			JSR PC,SUER2A	:GO SET UP ERROR INFO
4086	012552	004567	010340			JSR RS,SUNUM	:GO SET LINE NO. IN MSG
4087	012556	025144				LINE	
4088	012560	031076				EM50+53	
4089	012562	104053				ERROR 53	:TIMED OUT WAITING FOR XMIT DONE
4090	012564	000426				BR 85	:GO TEST NEXT SPEED
4091							
4092	012566	016703	012402		55:	MOV TIMEB,R3	:GET THE WAS COUNT
4093	012572	016704	012400			MOV TIMEC,R4	:GET LASTR CHECK COUNT
4094	012576	020304				CMP R3,R4	:COMPARE RELATIVE TIMES
4095	012600	103420				BLO 85	:BR IF THIS SPEED FASTER THAN LAST
4096							:SPEED TESTED
4097							
4098	012602	004767	011526		75:	JSR PC,SAPS	:SAVE THE ERROR PSM
4099	012606	016702	166404			MOV STMP7,R2	:GET SPEED CODE AND RIGHT JUSTIFY
4100	012612	000302				SWAB R2	
4101	012614	006202				ASR R2	
4102	012616	006202				ASR R2	
4103	012620	042702	177760			BIC #177760,R2	:STRIP AWAY ALL JUNK
4104	012624	004767	010046			JSR PC,SUER2A	:GO SET UP ERROR INFO
4105	012630	004567	010262			JSR RS,SUNUM	:GO PUT LINE NO. IN MSG

E01

MAINDEC-11-DZDMM-A
DZDMM.A.P11 T41

MACY11 27(663) 12-DEC-75 08:41 PAGE 80-2
TRANSMITTER TIMING TEST - ALL LINES - ALL SPEEDS

SEQ 0210

4106	012634	025144				LINE		
4107	012636	025441				EM17+41		
4108	012640	104017				ERROR	17	; TRANSMITTER SPEED INCORRECT
4109								
4110	012642	016767	012326	012326	BS:	MOV	TIMER, TIMEC	; SET UP NEW CHECK TIMER COUNT
4111	012650	062767	002100	166340		ADD	#2100, STMP7	; GENERATE NEXT SPEED
4112	012656	022767	035600	166332		CMP	#35600, STMP7	; DONE ALL SPEEDS ?
4113	012664	001253				BNE	25	; BR IF NOT
4114	012666	000641				BR	15	; GO TEST NEXT LINE
4115								

```

4118 ;*****
(3) ;*TEST 42 RECEIVER TIMING TEST - ALL LINES - ALL SPEEDS
(3) ;*****
(2) 012670 000004 TST42: SCOPE
(1) 012672 012767 000042 166334 MOV #STN-1,STESTN ;;SET TEST NUMBER IN MAIL BOX
4119 .REN X
(1) TEST ABSTRACT:
(1) *****

```

THIS TEST IS IDENTICAL TO TEST 40 EXCEPT IT WAITS FOR "DATA READY" TO CHECK RECEIVER TIMING. THE SEQUENCE IS SIMILAR AND THE SAME TIMERS ARE USED FOR ERROR CHECKING.

ERRORS:

1. [ERROR 54] IS CALLED TO REPORT RCVR TIMEOUT ERRORS
2. [ERROR 20] IS CALLED TO REPORT RCVR TIMING ERRORS

SYNC: M7277 SH3 INIT A H EF2

DEBUG: (SAME AS TEST 40)

KEY LOGIC: (SAME AS TEST 40 PLUS)

M7288 SH5,7,9,11 RX CLOCK NM L SIGNALS
M7280 BUF'DA LINE "N" UART PIN 19
RX CLOCK LINE "N" UART PIN 17

```

(1) X
(1) 4120 012700 012767 012730 166202 MOV #25,SLPERR ;SET UP ERROR LOOP RETURN
(1) 4121 012706 004767 010112 15: JSR PC,SELIN ;GO SELECT A LINE TO TEST
(1) 4122 012712 000532 BR TST43 ;BR IF TESTED ALL SELECTED LINES
(1) 4123 012714 012767 002100 166274 MOV #2100,STMP7 ;INIT TO START WITH LOWEST SPEED
(1) 4124 012722 012767 177777 012246 MOV #-1,TIMEC ;INIT RELATIVE TIME CHECKER
(1) 4125 012730 012711 004000 25: MOV #BIT11,(R1) ;CLEAR THE DH11
(1) 4126 012734 156711 012204 35: BISB LINE,(R1) ;SELECT IT IN THE SCR
(1) 4127 012740 012761 177777 000010 MOV #-1,BCR(R1) ;SET BYTE COUNT TO XFER 1 CHAR
(1) 4128 012746 005061 000006 CLR CAR(R1) ;GET TEST DATA STARTING AT LOC. 0
(1) 4129 012752 016761 166240 000004 MOV STMP7,LPR(R1) ;SELECT A XMIT SPEED
(1) 4130 012760 016761 011426 000012 MOV LINMSK,BAR(R1) ;ACTIVATE THE TRANSMITTER
(1) 4131
(1) 4132 012766 012767 000001 012176 MOV #1,TIMEA ;INIT TIMER A
(1) 4133 012774 005067 012174 CLR TIMEB ;INIT TIMER B
(1) 4134 013000 105711 45: TSTB (R1) ;RCVR DONE YET ??
(1) 4135 013002 100435 BMI SS ;BR IF YES
(1) 4136 013004 004767 011142 JSR PC,TIMEIT ;CALL THE TIMER
(1) 4137 013010 000773 BR #45 ;TIMER ROUTINE WILL MOVE RETURN PC
(1) 4138 ; AROUND THIS BRANCH IF TIME OUT OCCURS
(1) 4139
(1) 4140 013012 016767 166200 166160 MOV STMP7,STMP0 ;SAVE AND SET UP THE SPEED CODE

```

4141	013020	006367	166154		ASL	STMP0		
4142	013024	006367	166150		ASL	STMP0		
4143	013030	000367	166144		SWAB	STMP0		
4144	013034	042767	177760	166136	BIC	#177760,STMP0		
4145	013042	011103			MOV	(R1),R3	:GET THE WAS DATA	
4146	013044	010102			MOV	R1,R2	:MAKE REGADR = DEVADR	
4147	013046	012704	100200		MOV	#BIT15+BIT07,R4	:SET UP S/B DATA	
4148	013052	156704	012066		BISB	LINE,R4		
4149	013056	004767	007614		JSR	PC,SUER2A	:GO SET UP ERROR INFO	
4150	013062	004567	010030		JSR	RS,SUNUM	:GO SET LINE NO. IN MSG	
4151	013066	025144			LINE			
4152	013070	026720			EM22+51			
4153	013072	104054			ERROR	54	:TIMED OUT WAITING FOR CHAR AVAIL	
4154	013074	000426			BR	85	:GO TEST NEXT SPEED	
4155								
4156	013076	016703	012072	55:	MOV	TIMEB,R3	:GET THE WAS COUNT	
4157	013102	016704	012070		MOV	TIMEC,R4	:GET THE CHECK COUNT	
4158	013106	020304			CMP	R3,R4	:COMPARE RELATIVE TIMES	
4159	013110	103420			BLO	85	:BR IF TIME INDICATES THIS SPEED FASTER :THAN LAST SPEED	
4160								
4161								
4162	013112	004767	011216	75:	JSR	PC,SAPS	:SAVE THE ERROR PSW	
4163	013116	016702	166074		MOV	STMP7,R2	:GET SPEED CODE AND RIGHT JUSTIFY	
4164	013122	006302			ASL	R2		
4165	013124	006302			ASL	R2		
4166	013126	000302			SWAB	R2		
4167	013130	042702	177760		BIC	#177760,R2	:STRIP AWAY ALL JUNK	
4168	013134	004767	007536		JSR	PC,SUER2A	:GO SET UP ERROR INFO	
4169	013140	004567	007752		JSR	RS,SUNUM	:GO PUT LINE NO. IN MSG	
4170	013144	025144			LINE			
4171	013146	026600			EM20+36			
4172	013150	104020			ERROR	20	:RECEIVER SPEED INCORRECT	
4173								
4174	013152	016767	012016	012016	85:	MOV	TIMEB,TIMEC	:SET UP NEW CHECK TIMER COUNT
4175	013160	062767	002100	166030	ADD	#2100,STMP7	:GENERATE NEXT SPEED	
4176	013166	022767	035600	166022	CMP	#35600,STMP7	:DONE ALL SPEEDS ?	
4177	013174	001255			BNE	25	:BR IF NOT	
4178	013176	000643			BR	15	:GO TEST NEXT LINE	
4179								

NB1 LPR 00 H Fh?

```

(1)
(1)
4184 013210 012767 013236 165672 X
4185 013216 004767 007602 15:
4186 013222 000511
4187 013224 012705 025160
4188 013230 005002
4189 013236 012567 165760 25:
4190 013236 012711 004000 35:
4191 013242 156711 011676
4192 013246 012761 177777 000010
4193 013254 012761 001216 000006
4194 013262 012761 033500 000004
4195 013270 050261 000004
4196 013274 156761 011112 000012
4197
4198 013302 012767 000001 011662
4199 013310 005067 011660
4200 013314 105711 45:
4201 013316 100424
4202 013320 004767 010626
4203 013324 000773
4204
4205
4206 013326 004767 011002
4207 013332 011103
4208 013334 042703 177560
4209 013340 012704 000200
4210 013344 156704 011574
4211 013350 004767 007322
4212 013354 004567 007536
4213 013360 025144
4214 013362 026720
4215 013364 104055
4216 013366 000422
4217
4218 013370 016103 000002 55:
4219 013374 012704 000200
4220 013400 156704 011540
4221 013404 000304
4222 013406 156704 165604
4223 013412 020304
4224 013414 001407
4225
4226 013416 004767 007250
4227 013422 004567 007470
4228 013426 025144
4229 013430 026761
4230 013432 104023
4231
4232 013434 005202 65:
4233 013436 022702 000004
4234 013442 001273
4235 013444 000664

```

```

MOV #35,SLPERR ;SET UP ERROR LOOP RETURN
JSR PC,SELIN ;GO SELECT A LINE TO TEST
BR TST44 ;BR IF DONE ALL SELECTED LINES
MOV #DATA2,R5 ;GET POINTER TO DATA TABLE
CLR R2 ;INIT R2 TO START AT CHAR LENGTH OF 5 BITS
MOV (R5)+,STMP7 ;PUT TEST CHAR IN XMIT BUFFER
MOV #BIT11,(R1) ;CLEAR THE DMI1
BISB LINE,(R1) ;SELECT THE LINE
MOV #-1,BCR(R1) ;SET BYTE COUNT TO -1
MOV #STMP7,CAR(R1) ;SET CURRENT ADDRESS REG
MOV #33500,LPR(R1) ;SET BAUD RATE TO 9600
BIS R2,LPR(R1) ;SELECT CHAR LENGTH
BISB LINMSK,BAR(R1) ;ACTIVATE THE SELECTED LINE

MOV #1,TIMEA ;INIT TIMER A
CLR TIMEB ;INIT TIMER B
TSTB (R1) ;RCVR DONE YET ??
BMI 55 ;BR IF YES
JSR PC,TIMEIT ;CALL THE TIMER
BR 45 ;TIMER ROUTINE WILL MOVE RETURN PC
;AROUND THIS BRANCH IF TIME OUT OCCURS

JSR PC,SAPS ;SAVE THE ERROR PSM
MOV (R1),R3 ;GET THE SCR
BIC #177560,R3 ;CLEAR UNINTERESTING BITS
MOV #200,R4 ;SET UP S/B DATA
BISB LINE,R4
JSR PC,SUER2A ;GO SET UP ERROR INFO
JSR RS,SUNUM ;GO SET LINE NO. IN MSG
LINE
EM22+51
ERROR 55 ;CHAR AVAIL FAILED TO SET ON TIME
BR 65 ;GO TEST NEXT CHAR LENGTH

MOV NRC(R1),R3 ;GET THE WAS DATA
MOV #200,R4 ;SET UP THE S/B DATA IN R4
BISB LINE,R4
SWAB R4
BISB STMP7,R4
CMP R3,R4 ;WAS THE RCVD DATA CORRECT ??
BEQ 65 ;BR IF YES

JSR PC,SUER2 ;GO SET UP THE ERROR INFO
JSR RS,SUNUM ;GO PUT LINE NO. IN MSG
LINE
EM23+36
ERROR 23 ;DATA COMPARE ERROR

INC R2 ;DO NEXT CHAR LENGTH ON SELECTED LINE
CMP #4,R2 ;HAVE WE DONE ALL FOUR CHAR LENGTHS ??
BNE 25 ;BR IF NOT
BR 15 ;GO DO NEXT LINE

```

J01

MAINDEC-11-OZDMM-A
OZDMM.A.P11 T44

MACY11 27(663) 12-DEC-75 08:41 PAGE 83
SINGLE LINE DATA TEST - ALL LINES

SEO 0215

4238
(3)
(3)
(2)
(1)
4239
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)

*TEST 44 SINGLE LINE DATA TEST - ALL LINES

013446 000004
013450 012767 000044 165556

TST44: SCOPE
MOV 8STN-1,STESTN ;;SET TEST NUMBER IN MAIL BOX
REM %

TEST ABSTRACT:

THIS TEST TRANSMITS AND RECEIVES A BINARY COUNT PATTERN
(000 - 377) ON ALL SELECTED LINES. THE TEST SEQUENCE IS AS
FOLLOWS:

1. SET UP THE ERROR LOOP RETURN
2. GO SELECT A LINE NO. TO TEST - IF DONE ALL SELECTED LINES THEN GO TO TEST 44.
3. CLEAR THE DHI1 AND PRIME THE SELECTED LINE TO XMIT TO XMIT 256. CHARS AT 9600. BAUD - 8 BIT CHARS.
4. SET UP R5 TO POINT TO RCVR CORE BUFFER.
5. ACTIVATE THE SELECTED XMITTER.
6. WAIT FOR "CHAR AVAIL" TO SET BEFORE READING THE SILO. IF RCVR TIMEOUT REPORT ERROR AND RESTART AT STEP 2.
7. IF NO TIMEOUT READ THE SILO AND STORE THE WORD IN THE RCVR CORE BUFFER - WHEN THE BUFFER IS FULL GO TO STEPB IF NOT THEN GO TO STEP 6.
8. COMPARE THE XMIT AND RCVR CORE IMAGE BUFFERS AND REPORT ALL DATA COMPARE ERRORS.
9. CHECK THE "BAR" "BCR" AND "CAR" REGISTERS FOR CORRECT CONTENTS - REPORT ALL ERRORS.
10. GO TO STEP 2

NOTE: THIS TEST USES THE MAINT. BIT (SCRD9=1) TO TURN THE DATA AROUND INSTEAD OF THE MAINT CONNECTORS OR MODULE.

ERRORS:

1. [ERROR 22] IS CALLED TO REPORT "DATA AVAIL" TIMEOUT
2. [ERROR 37] " " " " DATA COMPARE ERRORS
3. [ERROR 40] " " " " "BAR" REG NOT CLEARED
4. [ERROR 10] " " " " "BCR" REG NOT ALL ZEROES
5. [ERROR 7] " " " " "CAR" REG NOT UPDATED CORRECTLY

SYNC: M7277 SH3 INIT A H EF2

DEBUG:

1. IF THE FAULT AFFECTS ONE OR MORE LINES IN AN 8 LINE GROUP <15:08> OR <07:00> SWAP THE M7280 MODULES. IF THE FAULT SHIFTS SO THAT THE ERROR INDICATES DIFFERENT LINES THE PROBLEM IS MOST LIKELY THE M7280 THE SYMPTOM SHIFTED TO.

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279

- 2. IF THE FAULT GIVES DATA ERRORS BUT AFFECTS ONLY CERTAIN PATTERNS ON ONE LINE THE FAULT IS MOST LIKELY A "UART" CHIP.
- 3. IF THE FAULT GIVES DATA ERRORS BUT AFFECTS ONLY CERTAIN PATTERNS ON ALL LINES SUSPECT THE DATA PATHS EXTERNAL TO, THE M7280 MODULES.
- 4. IF THE FAULT CAUSES NO DATA ERRORS BUT GIVES "BAR", "BCR", OR "CAR" ERRORS SUSPECT THE M7278 OR M7277 MODULES RESPECTFULLY.

```

KEY LOGIC          (REFER TO TEST 43)
*****
%
1S:  MOV      #25,SLPERR      ;SET UP ERROR LOOP RETURN
     JSR      PC,SELIN       ;GO SELECT A LINE TO TEST
     BR      11$             ;BR IF ALL SELECTED LINES DONE
11$: BR      25              ;GO TEST IT
     JMP      8$             ;EXIT TEST
2S:  MOV      DHADR,R1       ;RESET DEVADR
     MOV      @BIT11,(R1)    ;CLEAR THE DH11
     BISB    LINE,(R1)      ;SET SELECT BITS IN SCR
     MOV      @TBUF,CAR(R1)  ;SET UP BUS ADDRESS REG
     MOV      @-400,BCR(R1)  ;SET UP BYTE COUNT
     MOV      @33503,LPR(R1) ;SET LINE PARAMETERS
     MOV      @RBUF,RS      ;SET UP POINTER TO INPUT DATA BUFFER
     BIS      @BIT09,(R1)   ;SET MAINT MODE BIT
     MOV      LINMSK,BAR(R1) ;ACTIVATE THE SELECTED LINE

3S:  MOV      #2,TIMEA       ;INIT TIMER A
     CLR      TIMEB         ;INIT TIMER B
     TSTB    (R1)           ;RCVR DONE YET ??
     BMI     4$             ;BR IF YES
     JSR      PC,TIMEIT     ;CALL THE TIMER
     BR      3$            ;TIMER ROUTINE WILL MOVE RETURN PC
                              ;AROUND THIS BRANCH IF TIME OUT OCCURS

4S:  JSR      PC,SAPS        ;SAVE THE ERROR PSM
     MOV      R1,R2         ;SET UP REGADR
     MOV      (R2),R3       ;GET THE WAS DATA
     BIC     @176400,R3     ;CLEAR JUNK BITS
     MOV     @1200,R4       ;SET UP S/B DATA
     BISB    LINE,R4
     JSR      PC,SUER2A     ;GO SET UP ERROR INFO
     JSR      RS,SUNUM      ;PUT LINE NO. IN MESSAGE

     LINE
     EM22+51
     ERROR  22             ;CHAR AVAIL TIMEOUT
     BR      1$            ;GO TRY NEXT LINE

4S:  MOV      NRC(R1),(R5)+  ;SAVE THE RECEIVED DATA
     CMP     @RBUF+1000,RS  ;INPUT BUFFER FULL ??
     BNE     3$            ;BR IF NOT
  
```

4280	013656	012702	033436	MOV	#TBUF,R2	;SET UP POINTER TO OUTPUT BUFFER
4281	013662	012701	032436	MOV	#RBUF,R1	;SET UP POINTER TO INPUT BUFFER
4282	013666	111204		SS: MOV	(R2),R4	;SET UP S/B DATA IN R4
4283	013670	042704	177400	BIC	#177400,R4	
4284	013674	000304		SWAB	R4	
4285	013676	156704	011242	BISB	LINE,R4	
4286	013702	152704	000200	BISB	#200,R4	
4287	013706	000304		SWAB	R4	
4288	013710	011103		MOV	(R1),R3	;GET THE WAS DATA
4289	013712	020304		CMF	R3,R4	;DATA CORRECT ??
4290	013714	001407		BEQ	6S	;BR IF YES
4291	013716	004767	006750	JSR	PC,SUER2	;GO SET UP ERROR INFO
4292	013722	004567	007170	JSR	RS,SUNUM	;PUT LINE NO. IN MESSAGE
4293	013726	025144		LINE		
4294	013730	030176		EM37+33		
4295	013732	104037		ERROR	37	;DATA COMPARE ERROR
4296	013734	005202		6S: INC	R2	;UPDATE DATA BUFFER POINTERS
4297	013736	062701	000002	ADD	#2,R1	
4298	013742	022701	033436	CMF	#RBUF+1000,R1	;COMPARED ALL 256. CHARS ??
4299	013746	001347		BNE	5S	;BR IF NOT
4300	013750	016701	010424	MOV	DHADR,R1	;RESET DEVADR
4301	013754	010102		MOV	R1,R2	;SET UP REGADR
4302	013756	062702	000012	ADD	#BAR,R2	
4303	013762	005712		TST	(R2)	;WAS THE "BAR" ALL ZEROES ??
4304	013764	001413		BEQ	7S	;BR IF YES
4305	013766	004767	010342	JSR	PC,SAPS	;SAVE THE ERROR PSM
4306	013772	011203		MOV	(R2),R3	;GET THE WAS DATA
4307	013774	005004		CLR	R4	;SET UP S/B DATA
4308	013776	004767	006674	JSR	PC,SUER2A	;GO SET UP ERROR INFO
4309	014002	004567	007110	JSR	RS,SUNUM	;PUT LINE NO. IN MESSAGE
4310	014006	025144		LINE		
4311	014010	030241		EM40+40		
4312	014012	104040		ERROR	40	; "BAR" REG NOT ALL ZEROES
4313	014014	010102		7S: MOV	R1,R2	;SET UP REGADR
4314	014016	062702	000010	ADD	#BCR,R2	
4315	014022	005712		TST	(R2)	;BYTE COUNT REG ALL ZEROES ?
4316	014024	001413		BEQ	71S	;BR IF BYTE COUNT ZERO
4317	014026	004767	010302	JSR	PC,SAPS	;SAVE THE ERROR PSM
4318	014032	011203		MOV	(R2),R3	;GET THE WAS DATA
4319	014034	005004		CLR	R4	;SET UP THE S/B DATA
4320	014036	004767	006634	JSR	PC,SUER2A	;GO SET UP ERROR INFO
4321	014042	004567	007050	JSR	RS,SUNUM	;PUT LINE NO. IN MESSAGE
4322	014046	025144		LINE		
4323	014050	026046		EM10+44		
4324	014052	104010		ERROR	10	;BYTE COUNT NOT ALL ZEROES
4325	014054	010102		71S: MOV	R1,R2	;SET UP REGADR

4334	014056	062702	000006		ADD	#CAR,R2	
4335	014062	022712	034036		CMP	#TBUF+400,(R2)	;DID "CAR" INCREMENT PROPERLY ?
4336	014066	001414			BEQ	72S	;BR IF YES
4337							
4338	014070	004767	010240		JSR	PC,SAPS	;SAVE THE ERROR PSW
4339	014074	011203			MOV	(R2),R3	;GET THE WAS DATA
4340	014076	012704	034036		MOV	#TBUF+400,R4	;SET UP S/B DATA
4341	014102	004767	006570		JSR	PC,SUER2A	;GO SET UP ERROR INFO
4342	014106	004567	007004		JSR	RS,SUNUM	;GO PUT LINE NO IN MESSAGE
4343	014112	025144			LINE		
4344	014114	025777			EM7+47		
4345	014116	104007			ERROR	7	; "CAR" NOT UPDATED CORRECTLY
4346							
4347	014120	000167	177340	72S:	JMP	1S	;GO DO NEXT LINE
4348							
4349	014124	000240		8S:	NOP		;EXIT POINT

4373	014254	100423			BMI	45		:BR IF YES
4374	014256	004767	007670		JSR	PC, TIMEIT		:CALL THE TIMER
4375	014262	000773			BR	35		:TIMER ROUTINE WILL MOVE RETURN PC
4376								:AROUND THIS BRANCH IF TIME OUT OCCURS
4377								
4378	014264	004767	010044		JSR	PC, SAPS		:SAVE THE ERROR PSW
4379	014270	011103			MOV	(R1), R3		:GET THE WAS DATA
4380	014272	012704	100200		MOV	#100200, R4		:SET UP THE S/B DATA
4381	014276	156704	010642		BISB	LINE, R4		
4382	014302	010102			MOV	R1, R2		:SET UP REGADR
4383	014304	004767	006366		JSR	PC, SUER2A		:GO SET UP ERROR INFO
4384	014310	004567	006602		JSR	RS, SUNUM		:PUT LINE NO. IN MESSAGE
4385	014314	025144			LINE			
4386	014316	026720			EM22+5!			
4387	014320	104022			ERROR	22		:TIMED OUT WAITING FOR DATA AVAIL
4388	014322	000710			BR	15		:GO TEST NEXT LINE
4389								
4390	014324	016103	000002	45:	MOV	NRC(R1), R3		:GET THE WAS DATA
4391	014330	020304			CMF	R3, R4		:CORRECT DATA RECEIVED ??
4392	014332	001704			BEQ	15		:BR IF YES
4393								
4394	014334	004767	007774		JSR	PC, SAPS		:SAVE THE ERROR PSW
4395	014340	010102			MOV	R1, R2		:SET UP THE REGADR
4396	014342	062702	000002		ADD	#NRC, R2		
4397	014346	004767	006324		JSR	PC, SUER2A		:GO SET UP ERROR INFO
4398	014352	004567	006540		JSR	RS, SUNUM		:PUT LINE NO. IN MESSAGE
4399	014356	025144			LINE			
4400	014360	027712			EM33+40			
4401	014362	104333			ERROR	33		:INCORRECT DATA OR PARITY ERROR
4402	014364	000667			BR	15		:GO TEST NEXT LINE

4405
(3)
(3)
(2) 014366 000004
(1) 014370 012767 000046 164636
4406
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)

*TEST 46 MULTI-LINE PARITY DATA TEST - ALL SELECTED LINES

↑ST46: SCOPE
MOV #STN-1,STESTN ;;SET TEST NUMBER IN MAIL BOX

REM X
TEST ABSTRACT:

THIS TEST VERIFIES ALL SELECTED LINES CAN TRANSMIT AND RECEIVE
A BINARY COUNT PATTERN WHEN RUN CONCURRENTLY. ALL CHAR LENGTHS (5, 6, 7,
AND 8 BITS) ARE TESTED WITH BOTH EVEN AND ODD PARITY CHECKING SPECIFIED.
THE TEST ACTUALLY INCLUDES EIGHT SUB-TESTS - THE PARAMETERS FOR EACH
SUB-TEST RETRIEVED FROM A TABLE TAGGED "PRTYTB:". REFER TO THE FLOW
CHARTS IN THE PROGRAM DOCUMENT FOR A DETAILED DESCRIPTION OF THE TEST
SEQUENCES.

ERRORS:

- 1. [ERROR 41] IS CALLED TO REPORT FALSE RECEIVER INTRIS.
- 2. [ERROR 42] IS CALLED TO REPORT SILO OVERFLOW ERRORS
- 3. [ERROR 34] IS CALLED TO REPORT PARITY/DATA ERRORS
- 4. [ERROR 35] IS CALLED TO REPORT TEST TIMEOUTS

SYNC: (NONE)

DEBUG: (REFER TO TEST 45)

KEY LOGIC: (REFER TO TEST 45)

4407 014376 012767 014414 164504
4408 014404 012705 024462
4409 014410 005067 164602
4410 014414 162705 000004
4411 014420 005367 164572
4412 014424 022705 024516
4413 014430 001456
4414 014432 012706 001100
4415 014436 016701 007736
4416 014442 012567 164546
4417 014446 012567 164540
4418 014452 005267 164540
4419 014456 012711 004000
4420 014462 004767 006612
4421 014466 016167 000004 164504
4422 014474 016767 007710 164504
4423 014502 004767 007612
4424 014506 016702 007670
4425 014512 012722 014604

X
MOV #15,SLPERR ;SET UP THE ERROR LOOP RETURN
MOV #PRTYTB+4,RS ;SET UP POINTER TO TEST PARAMETERS
CLR \$TMP7 ;START WITH SUB TEST #00
15: SUB #4,RS ;RESET POINTER FOR ERROR LOOPS
DEC \$TMP7 ;RESET SUB TEST # FOR ERROR LOOP
25: CMP #PRTYTB+40,RS ;DONE ALL 8. SUB TESTS ??
BEQ 21\$;BR IF YES
MOV #STACK,SP ;RESET STACK POINTER FOR ERROR LOOPS
MOV DHADR,R1 ;RESET DEVADR FOR ERROR LOOPS
MOV (RS)+,\$TMP6 ;GET THE BYTE COUNT PARAMETER
MOV (RS)+,\$TMP5 ;GET THE LINE PARAMETERS
INC \$TMP7 ;GENERATE NEW SUB-TEST NO.
MOV #BIT11,(R1) ;CLEAR THE DH11
JSR PC,SUPPAR ;GO SET UP PARAMETER!
MOV LPR(R1),\$TMP0 ;SAVE CURRENT LINE PARAMETERS
MOV LINSEL,\$TMP3 ;SAVE SELECTED LINES PARAMETER
JSR PC,CHPS2 ;GO LOCK OUT INTRIS
MOV DHVCT,R2 ;SET UP THE VECTOR
MOV #35,(R2)+ ;GO TO 35 ON RCVR INTERRUPT


```

4426 014516 116712 010416          MOVB   DHRVL (R2)
4427 014522 012711 000100          MOV    #100, (R1)          ;ENABLE CHAR AVAIL INTERRUPTS
4428 014526 016767 007656 010122    MOV    LINSEL, LINACT      ;FLAG ALL SELECTED LINES ACTIVE
4429 014534 016761 007650 000012    MOV    LINSEL, BAR(R1)    ;ACTIVATE ALL SELECTED LINES
4430 014542 116767 164334 164434    MOVB   $STNM, STMP2      ;SAVE THE TEST NO.
4431 014550 042767 177400 164426    BIC    #177400, STMP2
4432 014556 004767 007522          JSR    PC, CHPS1          ;GO CLEAR PSW
4433 014562 000167 000176          JNP    7$                ;GO WAIT FOR INTERRUPTS
4434
4435 014566 012706 001100          21$:  MOV    #STACK, SP      ;RESTORE THE SP
4436 014572 004767 007506          JSR    PC, CHPS1          ;GO CLEAR PSW
4437 014576 004767 007322          JSR    PC, RESTRP        ;RESTORE TRAP CATCHER
4438 014602 000536          BR     TST47              ;GO TO NEXT TEST
4439
4440          ;RECEIVER INTERRUPT SERVICE ROUTINE
4441
4442 014604 105711          3$:  TSTB   (R1)            ;CHAR AVAIL SET
4443 014606 100404          BMI    4$                ;BR IF YES
4444
4445 014610 012711 004000          MOV    #BIT11, (R1)      ;CLEAR OUT THE DH11
4446 014614 104041          ERROR  41                ;RCVR FALSE INTERRUPT - CHAR AVAIL NOT SET
4447 014616 000702          BR     2$                ;GO TRY NEXT SUB TEST
4448
4449 014620 032711 040000          4$:  BIT    #BIT14, (R1)    ;SILO OVERFLOW ??
4450 014624 001404          BEQ    5$                ;BR IF NOT
4451
4452 014626 012711 004000          MOV    #BIT11, (R1)      ;CLEAR OUT THE DH11
4453 014632 104042          ERROR  42                ;SILO OVERFLOW ERROR
4454 014634 000673          BR     2$                ;GO TRY NEXT SUB TEST
4455
4456 014636 016103 000002          5$:  MOV    NRC(R1), R3      ;GET THE WAS DATA
4457 014642 010302          MOV    R3, R2            ;EXTRACT AND SAVE LINE NO.
4458 014644 000302          SWAB   R2
4459 014646 042702 177760    BIC    #177760, R2
4460 014652 010267 164332    MOV    R2, STMP4
4461 014656 006302          ASL    R2                ;GENERATE TABLE OFFSETR
4462 014660 026203 032436    CMP    REUF(R2), R3      ;CORRECT DATA RECEIVED ??
4463 014664 001426          BEQ    6$                ;BR IF YES
4464
4465 014666 004767 007442          JSR    PC, SAPS          ;SAVE THE ERROR PSW
4466 014672 012711 004000          MOV    #BIT11, (R1)      ;CLEAR OUT THE DH11
4467 014676 016204 032436          MOV    REUF(R2), R4      ;SET UP S/B DATA
4468 014702 062701 000002    ADD    #NRC, R1          ;SET UP WAS ADDRESS
4469 014706 062702 032436          ADD    #RBUF, R2        ;SET UP S/B ADDRESS
4470 014712 004767 005760          JSR    PC, SUER2A        ;GO SET UP ERROR INFO
4471 014716 004567 006174          JSR    RS, SUNUM        ;PUT LINE NO. IN MESSAGE
4472 014722 001210          STMP4
4473 014724 027766          EM34+51
4474 014726 004567 006164          JSR    RS, SUNUM        ;PUT SUBTEST NO. IN MESSAGE
4475 014732 001216          STMP7
4476 014734 030004          EM34+67
4477 014736 104034          ERROR  34                ;PARITY DATA COMPARE ERROR
4478 014740 000631          BR     2$                ;GO TRY NEXT SUBTEST
4479

```

```

4480 014742 105262 032436      6S:   INCB   RBUF(R2)      :GENERATE NEW RCVD DATA
4481 014746 005262 024660      :   INC    MULTB(R2)   :COUNT ONE BYTES RECEIVED
4482 014752 001003      :   BNE    61S        :BR IF NOT DONE
4483 014754 046267 024616 007674 :   BIC    LINBIT(R2),LINACT :FLAG THIS LINE DONE
4484 014762 000002      61S:   RTI      :RETURN TO WAIT ROUTINE
4485
4486      :WAIT ROUTINE
4487
4488 014764 012767 000002 010200 7S:   MOV    #2,TIMEA     :INIT TIMER A
4489 014772 005067 010176      :   CLR    TIMEB      :INIT TIMER B
4490 014776 005761 000012      8S:   TST    BAR(R1)     :ALL LINES DONE XMITTING ??
4491 015002 001413      :   BEQ    9S        :BR IF YES
4492 015004 004767 007142      :   JSR    PC,TIMEIT  :CALL THE TIMER
4493 015010 000772      :   BR     8S        :TIMER ROUTINE WILL MOVE RETURN PC
4494      :AROUND THIS BRANCH IF TIME OUT OCCURS
4495
4496 015012 016167 000012 164166      :   MOV    BAR(R1),STMP3 :SAVE THE ACTIVE LINES FLAG
4497 015020 012711 004000      :   MOV    #BIT11,(R1)  :CLEAR OUT THE DH11
4498 015024 104035      :   ERROR  35         :TIMED OUT WAITING FOR TRANSMITTERS TO FINISH
4499 015026 000167 177372      :   JMP    2S        :GO TRY NEXT SUBTEST
4500
4501
4502 015032 012767 000001 010132 9S:   MOV    #1,TIMEA     :INIT TIMER A
4503 015040 005067 010130      :   CLR    TIMEB      :INIT TIMER B
4504 015044 005767 007606      10S:  TST    LINACT      :ALL CHARS RECEIVED ?
4505 015050 001411      :   BEQ    11S       :BR IF YES
4506 015052 004767 007074      :   JSR    PC,TIMEIT  :CALL THE TIMER
4507 015056 000772      :   BR     10S      :TIMER ROUTINE WILL MOVE RETURN PC
4508      :AROUND THIS BRANCH IF TIME OUT OCCURS
4509
4510 015060 016767 007572 164120      :   MOV    LINACT,STMP3 :SET UP ACTIVE LINE PARAMETER
4511 015066 012711 004000      :   MOV    #BIT11,(R1)  :CLEAR OUT THE DH11
4512 015072 104035      :   ERROR  35         :SILO EMPTY TIMEOUT
4513 015074 000167 177324      11S:  JMP    2S        :GO TRY NEXT SUB TEST

```


(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)

7402 "OR" GATE CHIPS E38 OR E41
74157 MUX CHIPS E39 OR 342
E35 - PIN 2 STUCK LOW

M7289

SM4

AE GO L EK1
AE SCAN MUX E22 PIN 10
SAMPLE STATUS H E21-12

M7288

SM5, 7, 9, 11

AE ENABLE "NM" H CONTROL FLOPS
74174 CHIPS PIN 15

4517	015110	012767	015172	163772	%	MOV	#25,SLPERR	:SET UP ERROR LOOP RETURN
4518	015116	005067	164064			CLR	STMP3	:INIT I/O DATA FLAG
4519	015122	012711	004000			MOV	#BIT11,(R1)	:CLEAR THE DH11
4520	015126	012705	024516			MOV	#AETAB,RS	:GET POINTER TO AUTO ECHO DATA TABLE
4521	015132	005067	007256		7S:	CLR	LMSK1	:INIT BIT TEST MARKER
4522	015136	000261				SEC		:SET "C" BIT FOR MARKER
4523	015140	000401				BR	13S	:GO SHIFT MASK
4524	015142	000241			1S:	CLC		:INIT THE "C" BIT
4525	015144	006167	007244		13S:	ROL	LMSK1	:SHIFT BIT MARKER
4526	015150	001407				BEQ	12S	:BR IF DONE ALL LINES
4527	015152	012504				MOV	(RS)+,R4	:SET UP THE S/B DATA
4528	015154	036767	007234	007226		BIT	LMSK1,LINSEL	:TEST THIS LINE ?
4529	015162	001767				BEQ	1S	:BR IF NOT
4530	015164	004767	005634		11S:	JSR	PC,SELINE	:GO SELECT A LINE TO TEST
4531	015170				12S:			
(1)	015170	000522				BR	6S	:BR IF ALL SELECTED LINES TESTED
4532	015172	004767	006000		2S:	JSR	PC,CLCABC	:GO CLEAR "CAR" AND "BCR" MEMORIES
4533	015176	116711	007742			MOV	LINE,(R1)	:SET SELECT BITS IN SCR REG
4534	015202	012761	177777	000010		MOV	#-1,BCR(R1)	:SET UP TO XFER ONE CHAR
4535	015210	010561	000006			MOV	RS,CAR(R1)	:SET UP THE BUS ADDRESS REG
4536	015214	162761	000002	000006		SUB	#2,CAR(R1)	:CORRECT BUS ADDRESS
4537	015222	012767	000100	163766		MOV	#100,STMP7	:COUNT 64 CHARS TO BE RECEIVED IN AUTO ECHO
4538	015230	005067	163760			CLR	STMP6	:INIT CHAR COUNTER
4539	015234	012761	133503	000004		MOV	#133503,LPR(R1)	:SET UP LINE PARAMETER REG
4540	015242	016761	007144	000012		MOV	LINMSK,BAR(R1)	:ACTIVATE THE LINE
4541								
4542	015250	012767	000002	007714		MOV	#2,TIMEA	:INIT TIMER A
4543	015256	005067	007712			CLR	TIMEB	:INIT TIMERB
4544	015262	105711			3S:	TSTB	(R1)	:CHAR AVAIL SET ??
4545	015264	100427				BMI	4S	:BR IF YES
4546	015266	004767	006660			JSR	PC,TIMEIT	:CALL THE TIMER
4547	015272	000773				BR	3S	:TIMER ROUTINE WILL MOVE RETURN PC
4548								:AROUND THIS BRANCH IF TIME OUT OCCURS
4549								
4550	015274	004767	007034			JSR	PC,SAPS	:SAVE THE ERROR PSM
4551	015300	005061	000004			CLR	LPR(R1)	:TURN OFF AUTO ECHO MODE
4552	015304	010102				MOV	R1,R2	:MAKE REGADR = DEVRDR
4553	015306	011103				MOV	(R1),R3	:GET THE WAS DATA
4554	015310	042703	100000			BIC	#BIT15,R3	:CLEAR JUNK BIT
4555	015314	012774	000200			MOV	#200,R4	:SET UP S/B DATA
4556	015320	156704	007620			BISB	LINE,R4	

4557	015324	004767	005346		JSR	PC,SUER2A		:GO SET UP ERROR INFO
4558	015330	004567	005562		JSR	RS,SUNUM		:GO SET LINE NO. IN MSG
4559	015334	025144			LINE			
4560	015336	027116			EM24+35			
4561	015340	104024			ERROR	24		:DATA AVAIL FAILED TO SET ON TIME
4562	015342	000677			BR	18		:GO TRY NEXT LINE
4563								
4564	015344	005267	163644	45:	INC	STMP6		:COUNT ONE CHAR RECVD
4565	015350	016103	000002		MOV	MRC(R1),R3		:GET THE WAS DATA
4566	015354	020304			CMP	R3,R4		:WAS CHAR AUTO ECHOED CORRECTLY ?
4567	015356	001417			BEG	58		:BR IF YES
4568								
4569	015360	004767	006750		JSR	PC,SAPS		:SAVE THE ERROR PSM
4570	015364	005061	000004		CLR	LPR(R1)		:DISABLE AUTO ECHO
4571	015370	010102			MOV	R1,R2		:SET UP REGADR
4572	015372	062702	000002		ADD	8RAC,R2		
4573	015376	004767	005274		JSR	PC,SUER2A		:GO SET UP ERROR INFO
4574	015402	004567	005510		JSR	RS,SUNUM		:PUT LINE NO. IN ERROR MSG
4575	015406	025144			LINE			
4576	015410	027116			EM24+35			
4577	015412	104024			ERROR	24		:CHAR AUTO ECHOED INCORRECTLY
4578	015414	000652			BR	18		:GO TRY NEXT LINE
4579								
4580	015416	005367	163574	55:	DEC	STMP7		:COUNT ONE CHAR READ OUT OF 64
4581	015422	003317			BGT	38		:BR IF NOT LAST ONE
4582	015424	100646			BMI	18		:BR IF LAST ONE READ
4583	015426	042761	100000 000004		BIC	8BIT15,LPR(R1)		:DISABLE AUTO ECHO
4584	015434	000712			BR	38		:GO READ LAST CHAR
4585								
4586	015436	005167	163544	65:	COM	STMP3		:TOGGLE I/O FLAG
4587	015442	001406			BEG	TST50		:BR IF DONE BOTH I/O DATA
4588	015444	005067	007474		CLR	LINE		:INIT LINE NO TO 00
4589	015450	012705	024556		MOV	8AETAB0,R5		:SET POINTER TO 0'S TABLE
4590	015454	000167	177452		JMP	78		:REPEAT TEST FOR ZERO PATTERNS
4591								

4615	015564	010561	000006		MOV	R5,CAR(R1)	:SET BUS ADDR TO XMIT TEST CHAR	
4616	015570	162761	000002	000006	SUB	R2,CAR(R1)	:CORRECT THE ADDRESS	
4617	015576	012761	177777	000010	MOV	R-1,BCR(R1)	:XMIT ONE CHAR ON THIS LINE	
4618	015604	012761	133503	000004	MOV	R133503,LPR(R1)	:DO IT AT 9600 BAUD/8 BITS	
4619	015612	116767	163264	163364	MOV	STSTNH,STMP2	:SAVE THE TEST NO.	
4620	015620	042767	177400	163356	BIC	R177400,STMP2		
4621	015626	046767	006550	007022	BIC	LINMSK,LINACT	:MAKE THIS LINE APPEAR INACTIVE	
4622	015634	016761	006550	000012	MOV	LINSEL,BAR(R1)	:ACTIVATE ALL SELECTED TRANSMITTERS	
4623								
4624	015642	012767	000002	007322	21S:	MOV	R2,TIMEA	:INIT TIMER A
4625	015650	005067	007320			CLR	TIMEB	:INIT TIMER B
4626	015654	016103	000002		3S:	MOV	RAC(R1),R3	:GET THE WAS DATA
4627	015660	100414				BHI	4S	:BR IF YES
4628	015662	004767	006264			JSR	PC,TIMEIT	:CALL THE TIMER
4629	015666	000772				BR	3S	:TIMER ROUTINE WILL MOVE RETURN PC
4630								:AROUND THIS BRANCH IF TIME OUT OCCURS
4631								
4632	015670	016167	000004	163302		MOV	LPR(R1),STMP0	:SAVE THE CURRENT "LPR"
4633	015676	004567	005214			JSR	R5,SUNUM	:PUT LINE NO. IN MESSAGE
4634	015702	025144				LINE		
4635	015704	027607				EM32+3S		
4636	015706	104032				ERROR	32	:AUTO ECHO TIMEOUT
4637	015710	000700				BR	1S	:GO TRY NEXT LINE
4638								
4639	015712	010304			4S:	MOV	R3,R4	:EXTRACT LINE NUMBER OF RCVD CHAR
4640	015714	000304				SWAB	R4	
4641	015716	042704	177760			BIC	R177760,R4	
4642	015722	010402				MOV	R4,R2	:SAVE IT IN R2
4643	015724	006302				ASL	R2	:GENERATE TABLE INDEX IN R2
4644	015726	126704	007212			CHPB	LINE,R4	:IS THIS THE A.E. TEST LINE ??
4645	015732	001426				BEQ	5S	:BR IF YES
4646								
4647	015734	026203	032436			CMP	RBUF(R2),R3	:RCVD DATA CORRECT ??
4648	015740	001447				BEQ	6S	:BR IF IT WAS
4649								
4650	015742	004767	006366			JSR	PC,SAPS	:SAVE THE ERROR PSW
4651	015746	010467	163240			MOV	R4,STMP5	:SAVE THE LINE NUMBER
4652	015752	016204	032436			MOV	RBUF(R2),R4	:SET UP S/B DATA
4653	015756	062702	032436			ADD	RBUF,R2	:SET UP S/B ADDRESS
4654	015762	012701	177703			MOV	R177703,R1	:SET UP THE WAS ADDRESS
4655	015766	004767	004704			JSR	PC,SUER2A	:GO SET UP ERROR INFO
4656	015772	004567	005120			JSR	R5,SUNUM	:PUT LINE NO. IN MESSAGE
4657	015776	001212				STMP5		
4658	016000	027452				EM31+4S		
4659	016002	104031				ERROR	31	:NON-ECHO DATA COMPARE ERROR
4660	016004	000167	177502			JMP	1S	:GO TRY NEXT LINE
4661								
4662								
4663	016010	020367	163202		5S:	CMP	R3,STMP7	:CHAR ECHOED OK ??
4664	016014	001427				BEQ	7S	:BR IF YES
4665								
4666	016016	004767	006312			JSR	PC,SAPS	:SAVE THE ERROR PSW
4667	016022	012702	001216			MOV	RSTMP7,R2	:SAVE THE S/B ADDRESS
4668	016026	016704	163164			MOV	STMP7,R4	:SAVE THE S/B DATA

4669	016032	012701	177703		MOV	8177703,R1	:SAVE THE WAS ADDRESS
4670	016036	004767	004634		JSR	PC,SUER2A	:GO SET UP ERROR INFO
4671	016042	004567	005050		JSR	RS,SUNUM	:GO SET UP LINE NO. IN MESSAGE
4672	016046	025144			LINE		
4673	016050	027452			EM31+45		
4674	016052	104031			ERROR	31	:AUTO ECHO LINE DATA ERROR
4675	016054	000167	177432		JMP	18	:GO TRY NEXT LINE
4676							
4677	016060	105262	032436	6S:	INCB	RBUF(R2)	:GENERATE NEXT EXPECTED DATA ON THIS LINE
4678	016064	001266			BNE	218	:BR IF ITS NOT BACK TO 000
4679	016066	046267	024616	006562	BIC	LINBIT(R2),LINACT	:INDICATE THIOS LINE DONE 256 BYTES
4680	016074	005767	006556	7S:	TST	LINACT	:ALL LINES INACTIVE
4681	016100	001260			BNE	218	:BR IF NOT
4682	016102	042761	100000	000004	BIC	%BIT15,LPR(R1)	:TURN OFF THE A.E. BIT
4683	016110	105761	000017		TSTB	SSR+1(R1)	:SILO EMPTY ??
4684	016114	001002			BNE	88	:BR IF NOT
4685	016116	000167	177370		JMP	18	:GO TEST NEXT LINE
4686	016122	000167	177514	8S:	JMP	218	:GO EMPTY IT

4689
(3)
(3)
(2)
(1)
4690
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)

016126 000004
016130 012767 000051 163076

```

*****
*TEST 51 AUTO ECHO TEST 3 - ALL LINES
*****
†ST51: SCOPE
      MOV #STN-1,STESTN ;;SET TEST NUMBER IN MAIL BOX
      REM X
TEST ABSTRACT:
*****

```

THIS TEST IS IDENTICAL TO TEST 47 EXCEPT ALL SELECTED LINES ARE ACTIVATED CONCURRENTLY RATHER THAN ONE AT A TIME AND ONLY THE ALL 1'S DATA IS USED.

ERRORS:

- 1. [ERROR 36] IS CALLED TO REPORT "DATA AVAIL" TIMEOUTS
- 2. [ERROR 31] IS CALLED TO REPORT A.E. DATA ERRORS

SYNC: M7277 SH4 LOAD BAR LB+HB L CN2

DEBUG:

REFER TO TEST 47

KEY LOGIC:

REFER TO TEST 47

4691 016136 012767 016144 162744 X
 4692 016144 012711 004000 15:
 4693 016150 012705 000020
 4694 016154 012702 024516
 4695 016160 012703 032376
 4696 016164 010261 000006 25:
 4697 016170 012761 177777 000010
 4698 016176 012761 131403 000004
 4699 016204 005023
 4700 016206 062702 000002
 4701 016212 005211
 4702 016214 005305
 4703 016216 001362
 4704 016220 116767 162656 162756
 4705 016226 042767 177400 162750
 4706 016234 016767 006150 006414
 4707 016242 016761 006142 000012
 4708
 4709 016250 012767 000002 006714
 4710 016256 005067 006712
 4711 016262 105711 38:

```

X
MOV #15,SLPERR ;SET UP THE ERROR LOOP RETURN
15: MOV #BIT11,(R1) ;CLEAR OUT THE DH11
MOV #20,R5 ;INIT COUNTER TO SET UP 16. LINES
MOV #AETAB,R2 ;SET UP POINTER TO AUTO ECHO TEST DATA
MOV #RCNT,R3 ;R3 POINTS TO TABLE OF CHAR COUNTERS
25: MOV R2,CAR(R1) ;SET UP BUS ADDRESS REG
MOV #-1,BCR(R1) ;SET UP BYTE COUNT REG
MOV #131403,LPR(R1) ;SET UP LINE PARAMETERS
CLR (R3)+ ;CLEAR A COUNTER
ADD #2,R2 ;UPDATE POINTERS
INC (R1) ;SELECT NEXT LINE
DEC R5 ;COUNT ONE DONE
BNE 25 ;BR TILL 16. DONE
MOVB STSTN,STMP2 ;SAVE THE TEST NO.
BIC #177400,STMP2
MOV LINSEL,LINACT ;SET FLAG TO INDICATE ALL 16. ACTIVE
MOV LINSEL,BAR(R1) ;ACTIVATE ALL XMITTERS

MOV #2,TIMEA ;INIT TIMER A
38: CLR TIMEB ;INIT TIMERB
TSTB (R1) ;CHAR AVAIL SET YET ?

```

```

4712 016264 100410      BMI      45      ;BR IF YES
4713 016266 004767 005660 JSR      PC,TIMEIT ;CALL THE TIMER
4714 016272 000773      BR       35      ;TIMER ROUTINE WILL MOVE RETURN PC
4715                                     ;AROUND THIS BRANCH IF TIME OUT OCCURS
4716
4717 016274 016167 000004 162676 MOV      LPR(R1),STMP0 ;SAVE THE "LPR" REG
4718 016302 104036      ERROR   36      ;DATA AVAILABLE TIMEOUT
4719 016304 000453      BR       TST52    ;EXIT TEST ON ERROR
4720
4721 016306 016103 000002      4S:    MOV      NRC(R1),R3    ;GET THE WAS DATA
4722 016312 010302      MOV      R3,R2    ;BUILD AND SAVE LINE NO.
4723 016314 000302      SWAB    R2
4724 016316 042702 177760      BIC     #177760,R2
4725 016322 010267 162666      MOV      R2,STMP6 ;SAVE THE LINE NO.
4726 016326 006302      ASL     R2        ;GENERATE TABLE OFFSET
4727 016330 005262 032376      INC     RCNT(R2)  ;COUNT THE CHARACTER
4728 016334 020362 024516      CMP     R3,AETAB(R2) ;IS THE DATA CORRECT ??
4729 016340 001420      BEQ    55        ;BR IF YES
4730
4731 016342 004767 005766      JSR     PC,SAPS   ;SAVE THE ERROR PSW
4732 016346 016204 024516      MOV     AETAB(R2),R4 ;GET THE S/B DATA
4733 016352 062702 024516      ADD     #AETAB,R2  ;GENERATE S/B ADDRESS
4734 016356 062701 000002      ADD     #NRC,R1   ;GENERATE THE WAS ADDRESS
4735 016362 004767 004310      JSR     PC,SUER2A ;GO SET UP ERROR INFO
4736 016366 004567 004524      JSR     RS,SUNUM  ;PUT LINE NO. IN MESSAGE
4737 016372 001214      STMP6  EM31+45
4738 016374 027452      ERROR  31
4739 016376 104031      ERROR  31      ;DATA COMPARE ERROR
4740 016400 000415      BR     TST52    ;EXIT TEST ON ERROR
4741
4742 016402 022762 000100 032376 5S:    CMP     #100,RCNT(R2) ;DONE 64. CHARS ON THIS LINE ?
4743 016410 001324      BNE    35      ;BR IF NOT
4744 016412 016711 162576      MOV     STMP6 (R1) ;SELECT LINE IN SCR REG
4745 016416 042761 100000 000004 BIC     #BIT15,LPR(R1) ;TURN OFF A.E. BIT
4746 016424 046267 024616 006224 BIC     LINBIT(R2),LINACT ;ALL LINES INACTIVE ??
4747 016432 001313      BNE    35      ;BR IF NOT

```

4750
(3)
(3)
(2) 016434 000004
(1) 016436 012767 000052 162570
4751
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)

```

;*****
;TEST 52 BREAK BIT TEST - ALL LINES
;*****
TST52: SCOPE
MOV #STN-1,STESTN ;;SET TEST NUMBER IN MAIL BOX
REM X
TEST ABSTRACT:
*****
    
```

THIS TEST VERIFIES THAT THE "BREAK" FEATURE WORKS PROPERLY FOR ALL SELECTED LINES. THE TEST SEQUENCE IS AS FOLLOWS:

1. SET UP THE ERROR LOOP RETURN
2. RETRIEVE THE CORRECT S/B DATA FROM THE "BREAK" DATA TABLE AND UPDATE THE POINTER.
3. GO SELECT A LINE TO TEST - GO TO TEST 52 IF DONE ALL SELECTED LINES
4. RESET THE DH11 AND CLEAR THE "CAR" AND "BCR" MEMORIES.
5. PRIME SELECTED LINE TO OUTPUT TWO "NULL" CHARS TO CLEAR UART
6. ACTIVATE THE SELECTED LINE
7. WAIT FOR SILO TO RECEIVE TWO NULLS - IF TIMEOUT REPORT ERROR AND RESTART AT STEP 2
8. IF NO TIMEOUT CLEAR THE SELECTED DH11 AND RESELECT LINE NO.
9. PRIME SELECTED LINE TO OUT PUT 256. CHARS.
10. SET THE SELECTED LINE'S BREAK BIT
11. ACTIVATE THE SELECTED LINE
12. WAIT FOR "BAR" REG TO CLEAR -F TIMEOUT REPORT ERROR AND RESTART AT STEP 2
13. IF NO TIMEOUT VERIFY THAT THE SILO RECEIVED ONLY ONE CHAR- IF NOT REPORT ERROR AND RESTART AT STEP 2
14. IF SILO RECEIVED ONLY ONE CHAR VERIFY THAT IT WAS A "BREAK" CHAR - IF NOT REPORT ERROR - AND RESTART AT STEP 2

ERRORS:

1. (ERROR 25) IS CALLED TO REPORT ALL ERRORS

SYNC: M7277 SH4 LOAD BCR H FU1

DEBUG:

1. IF ALL LINES FAILED SUSPECT THAT THE M7277 IS NOT GENERATING THE BREAK CONTROL REG LOAD SIGNAL.
2. IF ONLY ONE LINE FAILS SUSPECT THE BREAK CONTROL LOGIC ON THE M7278

KEY SIGNALS:

M7277 SH4 LOAD BCR H FU1

```

(1)
(1)
(1)
(1)
(1)
(1)
M7278 SHS THRU SH8
74175 REGISTER CHIPS E51, E38, E67, E60
7400 DRIVERS E45, E46, E75, E76
X
4752 016444 012767 016524 162436 MOV #25,SLPERR ;SET UP ERROR LOOP RETURN
4753 016452 012705 024720 MOV #BRKTAB,RS ;SET UP POINTER TO BREAK DATA TABLE
4754 016456 005067 005732 CLR LMSK1 ;INIT BIT TEST MASK
4755 016462 000261 SEC ;SET BIT MARKER IN "C"
4756 016464 000401 BR 125 ;GO SHIFT MASK
4757 016466 000241 15: CLC ;INIT THE "C" BIT
4758 016470 006167 005720 125: ROL LMSK1 ;SHIFT TEST MARKER
4759 016474 001411 BEQ 115 ;BR IF ALL LINES DONE
4760 016476 012504 MOV (R5)+,R4 ;GET TEST DATA FOR THIS LINE
4761 016500 036767 005710 005702 BIT LMSK1,LINSEL ;LINE SELECTED ?
4762 016506 001767 BEQ 15 ;BR IF NOT
4763 016510 004767 004310 JSR PC,SELINE ;GO SELECT A LINE TO TEST
4764 016514 000401 BR 115 ;BR IF DONE ALL SELECTED LINES
4765 016516 000402 BR 25 ;GO TEST THE SELECTED LINE
4766 016520 000167 000454 115: JMP 95 ;GO EXIT TEST
4767 016524 012711 004000 25: MOV #BIT11,(R1) ;CLEAR THE DM11
4768 016530 004767 004442 JSR PC,CLCABC ;GO CLR THE "CAR" AND "BCR" MEMORIES
4769 016534 116711 006404 MOV#B LINE,(R1) ;SELECT THE LINE
4770
4771 016540 012761 025200 000006 MOV #TNUL, CAR(R1) ;SET UP TO OUTPUT TWO NULL CHARS
4772 016546 012761 177776 000010 MOV #2,BCR(R1) ;SET BYTE COUNT TO 2
4773 016554 012761 033503 000004 MOV #33503,LPR(R1) ;SET UP LINE PARAMETERS
4774 016562 016761 005624 000012 MOV LINMSK,BAR(R1) ;ACTIVATE SELECTED LINE
4775
4776 016570 012767 000001 006374 MOV #1,TIMEA ;INIT TIMER A
4777 016576 005067 006372 CLR TIMEB ;INIT TIMER B
4778 016602 122761 000002 000017 35: CHPB #2,SSR+1(R1) ;TWO CHARS RECEIVED ??
4779 016610 001432 BEQ 45 ;BR IF YES
4780 016612 004767 005334 JSR PC,TIMEIT ;CALL THE TIMER
4781 016616 000771 BR 35 ;TIMER ROUTINE WILL MOVE RETURN PC
4782 ;AROUND THIS BRANCH IF TIME OUT OCCURS
4783
4784 016620 004767 005510 JSR PC,SAPS ;SAVE THE ERROR PSM
4785 016624 010467 162352 MOV R4,STMP1 ;SAVE S/B DATA
4786 016630 010102 MOV R1,R2 ;SET UP REGADR
4787 016632 062702 000016 ADD #55R,R2
4788 016636 011203 MOV (R2),R3 ;GET THE WAS DATA
4789 016640 042703 100377 BIC #100377,R3 ;CLEAR JUNK
4790 016644 012704 000002 MOV #2,R4 ;SET UP S/B DATA
4791 016650 000304 SWAB R4
4792 016652 004767 004020 JSR PC,SUER2A ;GO SET UP ERROR INFO
4793 016656 004567 004234 JSR RS,SUNUM ;GO PUT LINE NO. IN MESSAGE
4794 016662 025144 LINE
4795 016664 027155 EM25+34
4796 016666 016704 162310 MOV STMP1,R4 ;RESTORE S/B DATA
4797 016672 104025 ERROR 25 ;TIMED OUT WAITING FOR TWO NULLS
4798 016674 000674 BR 15 ;GO TRY NEXT LINE
4799

```

```

4800 016676 012711 004000      4S:  MOV      #BIT11,(R1)      ;CLEAR THE INTERFACE
4801 016702 116711 006236      MOV      LINE,(R1)      ;SELECT THE LINE
4802 016706 012761 033436 000006  MOV      #BUF,CAR(R1)   ;SET UP BUS ADDRESS REG FOR XMITTR
4803 016714 012761 177400 000010  MOV      #400,BCR(R1)  ;SET BYTE COUNT TO XMIT 256(10) CHARS
4804 016722 012761 033503 000004  MOV      #33503,LPR(R1) ;SET UP LINE PARAMETERS
4805 016730 016761 005456 000014  MOV      LINMSK,BKR(R1) ;SET BREAK BIT FOR ACTIVE LINE
4806 016736 016761 005450 000012  MOV      LINMSK,BAR(R1) ;ACTIVATE THE SELECTED LINE
4807
4808 016744 012767 000005 006220  MOV      #5,TIMEA      ;INIT TIMER A
4809 016752 005067 006216      CLR      TIMEB        ;INIT TIMER B
4810 016756 005761 000012      5S:  TST      BAR(R1)      ;BAR BIT CLEARED ??
4811 016762 001426      BEQ      #S           ;BR IF 0 YES
4812 016764 004767 005162      JSR      PC,TIMEIT    ;CALL THE TIMER
4813 016770 000772      BR       #S           ;TIMER ROUTINE WILL MOVE RETURN PC
4814                                     ;AROUND THIS BRANCH IF TIME OUT OCCURS
4815
4816 016772 004767 005336      JSR      PC,SAPS      ;SAVE THE ERROR PSW
4817 016776 010467 162200      MOV      R4,STMP1     ;SAVE THE S/B DATA
4818 017002 010102      MOV      R1,R2        ;SET UP REGADR
4819 017004 062702 000012      ADD      #BAR,R2
4820 017010 011203      MOV      (R2),R3
4821 017012 005004      CLR      R4
4822 017014 004767 003656      JSR      PC,SUER2A    ;GO SET UP ERROR INFO
4823 017020 004567 004072      JSR      R5,SUNUM     ;PUT LINE NO IN MESSAGE
4824 017024 025144      LINE
4825 017026 027155      EM25+34
4826 017030 016704 162146      MOV      STMP1,R4     ;RESTORE THE S/B DATA
4827 017034 104025      ERROR    25          ;BAR BIT FAILED TO CLEAR
4828 017036 000613      BR       #S           ;GO TRY NEXT LINE
4829
4830 017040 122761 000001 000017  6S:  CNPB     #1,SSR+1(R1) ;ONE CHAR RECEIVED ?
4831 017046 001430      BEQ      #S           ;BR IF YES
4832
4833 017050 004767 005260      JSR      PC,SAPS      ;SAVE THE ERROR PSW
4834 017054 010467 162122      MOV      R4,STMP1     ;SAVE THE S/B DATA
4835 017060 010102      MOV      R1,R2        ;SET UP REGADR
4836 017062 062702 000016      ADD      #SSR,R2
4837 017066 011203      MOV      (R2),R3
4838 017070 042703 100377      BIC      #100377,R3   ;GET THE WAS DATA
4839 017074 012704 000001      MOV      #1,R4        ;CLEAR JUNK
4840 017100 000304      SWAB     R4           ;SET UP S/B DATA
4841 017102 004767 003570      JSR      PC,SUER2A    ;GO SET UP ERROR INFO
4842 017106 004567 004004      JSR      R5,SUNUM     ;GO PUT LINE NO. IN MESSAGE
4843 017112 025144      LINE
4844 017114 027155      EM25+34
4845 017116 016704 162060      MOV      STMP1,R4     ;RESTORE THE S/B DATA
4846 017122 104025      ERROR    25          ;FAILED TO RECEIVE THE ONE CHAR
4847 017124 000167 177336      JMP      #S           ;GO TRY NEXT LINE
4848
4849
4850 017130 016103 000002      7S:  MOV      NRC(R1),R3   ;GET THE WAS DATA
4851 017134 020304      CMP      R3,R4        ;WAS IT A BREAK CHAR ?
4852 017136 001002      BNE     #S           ;BR IF NOT CORRECT
4853 017140 000167 177322      JMP      #S           ;GO TEST NEXT LINE

```


4874	017232	004767	003740		JSR	PC, CLCABC	:GO CLR THE "CAR" AND "BCR" MEMORIES
4875	017236	156711	005702		BISB	LINE, (R1)	:SELECT THE LINE
4876	017242	012761	033436	000006	MOV	#STBUF, CAR(R1)	:POINT TO XMIT BUFFER
4877	017250	012761	177400	000010	MOV	#-400, BCR(R1)	:XMIT 256(10) CHARS
4878	017256	012761	073503	000004	MOV	#73503, LPR(R1)	:SET UP THE LINE PARAMETERS
4879	017264	016761	005122	000012	MOV	LINYSK, BAR(R1)	:ACTIVATE THE SELECTED LINE
4880							
4881	017272	012767	000001	005672	MOV	#1, TIMEA	:INIT TIME A
4882	017300	005067	005670		CLR	TIMEB	:INIT TIME B
4883	017304	005761	000012	3S:	TST	BAR(R1)	:WAIT FOR XMITR TO FINISH
4884	017310	001423			BEG	4S	:BR IF XMITR FINISHED
4885	017312	004767	004634		JSR	PC, TIMEIT	:CALL TIMER
4886	017316	000772			BR	3S	:TIMER WILL MOVE RETURN PC AROUND :THIS BRANCH IF TIMEOUT OCCURS
4887							
4888							
4889	017320	004767	005010		JSR	PC, SAPS	:SAVE THE ERROR PSM
4890	017324	016103	000012		MOV	BAR(R1), R3	:GET THE WAS DATA
4891	017330	010102			MOV	R1, R2	:SET UP REGADR
4892	017332	062702	000012		ADD	#BAR, R2	
4893	017336	005004			CLR	R4	:SET UP NEW S/B DATA
4894	017340	004767	003332		JSR	PC, SUER2A	:GO SET UP THE ERROR INFO
4895	017344	004567	003546		JSR	RS, SUNUM	:PUT LINE NO. IN MESSAGE
4896	017350	025144			LINE		
4897	017352	027220			EM26+37		
4898	017354	104026			ERROR	26	:BAR BIT FAILED TO CLEAR ON TIME
4899	017356	000720			BR	1S	:GO TRY NEXT LINE
4900							
4901	017360	105711		4S:	TSTB	(R1)	:CHAR AVAIL SET ??
4902	017362	100316			BPL	1S	:BR IF NOT IT SHOULDN'T BE
4903							
4904	017364	004767	004744		JSR	PC, SAPS	:SAVE THE ERROR PSM
4905	017370	010102			MOV	R1, R2	:SET UP REGADR
4906	017372	011103			MOV	(R1), R3	:GET WAS DATA
4907	017374	042703	100000		BIC	#BIT15, R3	:CLEAR JUNK BIT
4908	017400	116704	005540		MOV	LINE, R4	:SET UP S/B DATA
4909	017404	004767	003266		JSR	PC, SUER2A	:GO SETUP ERROR INFO
4910	017410	004567	003502		JSR	RS, SUNUM	:PUT LINE NO. IN MSG
4911	017414	025144			LINE		
4912	017416	027220			EM26+37		
4913	017420	104026			ERROR	26	:HALF DUPLEX FAILED TO BLIND RECVR
4914	017422	000676			BR	1S	:GO SELECT NEXT LINE

4917
(3)
(3)
(2) 017424 000004
(1) 017426 012767 000054 161600
4918

: #TEST 54 VERIFY THAT OVERRUN CAN SET PROPERLY - ALL LINES

↑TST54: SCOPE
MOV #STN-1,STESTN ;;SET TEST NUMBER IN MAIL BOX

REM X
TEST ABSTRACT:

THIS TEST VERIFIES THAT "OVERRUN" SETS PROPERLY FOR ALL LINES
THAT ARE SELECTED FOR TEST WHEN THE OVERRUN CONDITION IS FORCED BY THE
PROGRAM. THE TEST SEQUENCE IS AS FOLLOWS:

1. SET UP THE ERROR LOOP RETURN
2. SELECT A LINE NO. TO TEST - IF DONE ALL LINES GO TO
END OF PASS HANDLER.
3. PRIME THE SELECTED LINE TO XMIT 68. CHARS
4. ACTIVATE THE SELECTED LINE
5. WAIT FOR "XMIT DONE" TO SET - IF TIMEOUT REPORT ERROR
AND RESTART AT STEP 2
6. IF NO TIMEOUT READ 65. CHARS FROM THE SILO AND VERIFY THAT
"OVERRUN" IS SET ON THE LAST WORD READ
7. IF NOT REPORT ERROR AND RESTART AT STEP 2

ERRORS:

1. [ERROR 50] IS CALLED TO REPORT "XMIT DONE " TIMEOUTS
2. [ERROR 56] IS CALLED TO REPORT "OVERRUN" ERROR

SYNC: M7277 SH3 INIT A H EF2

DEBUG:

1. IF FAULT APPEARS ON ONLY ONE LINE SUSPECT UART MODULE
FOR THE APPROPRIATE LINE IN QUESTION.
2. IF FAULT APPEARS ON ALL LINES SUSPECT THE M7279 MODULE

KEY LOGIC:

M7279	SH1	MASTER OR H	E12-9
	SH2	MEMORY CHIP (3341)	E13-11
M7280	SH2	UC1 OR 2 MASTER OR	EN2
	SH2-5	UART PIN 15 (BUF OR LINE NN)	

%

4919 017434 012767 017450 161446
4920 017442 004767 003356

MOV #25,SLPERR ;SET UP ERROR LOOP RETURN
JSR PC,SELINE ;GO SELECT A LINE # TO TEST

4921	017446	000512			BR	ENDA	:: BR IF DONE ALL SELECTED LINES	
4922	017450	012711	004000	2S:	MOV	#BIT11,(R1)	:: CLEAR OUT THE DM11	
4923	017454	116711	005464		MOV	LINE,(R1)	:: SELECT THE LINE TO TEST	
4924	017460	012761	033436	000006	MOV	#TBUF,CAR(R1)	:: SET UP CURRENT ADDRESS	
4925	017466	012761	177674	000010	MOV	#-68,,BCR(R1)	:: SET UP BYTE COUNT REG	
4926	017474	012761	033503	000004	MOV	#33503,LPR(R1)	:: DO IT AT 9600 BAUD - 8 BITS	
4927	017502	016761	004704	000012	MOV	LINMSK,BAR(R1)	:: ACTIVATE THE SELECTED LINE	
4928								
4929	017510	012767	000001	005454	MOV	#1,TIMEA	:: INIT TIMERS A AND B	
4930	017516	005067	005452		CLR	TIMEB		
4931	017522	005711		3S:	TST	(R1)	:: TRANSMITTER DONE ??	
4932	017524	100425			BMI	45	:: BR IF YES	
4933	017526	004767	004420		JSR	PC,TIMEIT	:: CALL TIMER	
4934	017532	000773			BR	35	:: BR IF NO TIMEOUT	
4935								
4936	017534	004767	004574		JSR	PC,SAPS	:: GO SAVE PSW	
4937	017540	011103			MOV	(R1),R3	:: GET THE WAS DATA	
4938	017542	042703	077760		BIC	#77760,R3	:: CLEAR UNINTERESTING BITS	
4939	017546	116704	005372		MOV	LINE,R4	:: SET UP S/B DATA	
4940	017552	052704	100000		BIS	#BIT15,R4		
4941	017556	010102			MOV	R1,R2	:: SET UP REGADR	
4942	017560	004767	003112		JSR	PC,SUER2A	:: GO SET UP ERROR INFO	
4943	017564	004567	003326		JSR	RS,SUNUM	:: PUT LINE NO. IN MESSAGE HEADER	
4944	017570	025144			LINE			
4945	017572	031076			EM50+53			
4946	017574	104050			ERROR	50	:: REPORT XMIT DONE TIME OUT	
4947	017576	000721			BR	18	:: GO TRY NEXT LINE	
4948								
4949	017600	012767	000101	161374	4S:	MOV	#65,,STMP1	:: SET UP TO READ 65. WORDS FROM SILO
4950	017606	116704	005332		MOV	LINE,R4	:: SET UP S/B DATA	
4951	017612	000304			SWAB	R4		
4952	017614	152704	000101		BIS	#65,R4		
4953	017620	052704	140000		BIS	#BIT15+BIT14,R4	:: PUT IN OVERRUN AND VALID DATA BITS	
4954	017624	016103	000002	5S:	MOV	NRC(R1),R3	:: GET WAS DATA FROM SILO	
4955	017630	005367	161346		DEC	STMP1	:: COUNT ONE WORD READ	
4956	017634	001373			BNE	55	:: BR TIL 65. READ	
4957	017636	020304			CMPS	R3,R4	:: WAS DATA AND OVERRUN CORRECT ??	
4958	017640	001700			BEQ	18	:: BR IF YES TRY NEXT SELECTED LINE	
4959								
4960	017642	004767	004466		JSR	PC,SAPS	:: GO SAVE PSW	
4961	017646	010102			MOV	R1,R2	:: SET UP REGADR	
4962	017650	062702	000002		ADD	#NRC,R2		
4963	017654	004767	003016		JSR	PC,SUER2A	:: GO SET UP ERROR INFO	
4964	017660	004567	003232		JSR	RS,SUNUM	:: GO PUT LINE NO. IN MSG HDR	
4965	017664	025144			LINE			
4966	017666	031514			EM56+42			
4967	017670	104051			ERROR	56	:: OVERRUN OR DATA INCORRECT	
4968	017672	000663			BR	18	:: GO TEST NEXT SELECTED LINE	
4969								

```

4972 017674 000004      ENDA:  SCOPE
4973 017676 012767 000240 000054  MOV      #240,SEOP      ;NOP THE SCOPE AT THE BEGINNING OF EOP
4974 017704 005267 005232      INC      DHNUM        ;GENERATE NEW DH11 NUMBER
4975 017710 062767 000002 005232  ADD      #2,ADPTR     ;UPDATE THE TABLE POINTERS
4976 017716 062767 000002 005226  ADD      #2,VCPTR
4977 017724 062767 000002 005222  ADD      #2,BAPTR
4978 017732 006367 004446      ASL      SELMSK      ;SHIFT MARKER TO TEST NEXT DH11
4979 017736 001410      BEQ      SEOP        ;BR IF TESTED ALL SELECTED DH11'S
4980 017740 036767 004440 004440  BIT      SELMSK,DHSEL ;IS THIS DH11 SELECTED ?
4981 017746 001752      BEQ      ENDA        ;BR IF NOT
4982 017750 105067 161126      CLR      $STSNM     ;INIT TEST NUMBER
4983 017754 000167 162632      JMP      RSTRTA     ;GO TEST THIS DH11
4984
(1) ;*****
(1)
(1) .SBTTL  END OF PASS ROUTINE
(1)
(1) ;*INCREMENT THE PASS NUMBER (SPASS)
(1) ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
(1) ;*IF THERES A MONITOR GO TO IT
(1) ;*IF THERE ISN'T JUMP TO START2
(1)
(1) SEOP:
(1) 017760      SCOPE
(1) 017760 000004      CLR      $STSNM     ;;ZERO THE TEST NUMBER
(1) 017762 005067 161114      CLR      $TIMES     ;;ZERO THE NUMBER OF ITERATIONS
(1) 017766 005067 161226      INC      SPASS      ;;INCREMENT THE PASS NUMBER
(1) 017772 005267 161240      BIC      #100000,SPASS ;;DON'T ALLOW A NEG. NUMBER
(1) 017776 042767 100000 161232  DEC      (PC)+      ;;LOOP?
(1) 020004 005327      SEOPCT: .WORD      1
(1) 020006 000001      BGT      $DOAGN     ;;YES
(1) 020010 003022      MOV      (PC)+,2(PC)+ ;;RESTORE COUNTER
(1) 020012 012737      SENDCT: .WORD      1
(1) 020014 000001      SEOPCT
(1) 020016 020006      TYPE      $SENDMG   ;;TYPE "END PASS #"
(1) 020020 104400 020062      MOV      SPASS,-(SP) ;;SAVE SPASS FOR TYPEOUT
(1) 020024 016746 161206      TYPDS    ;;GO TYPE--DECIMAL ASCII WITH SIGN
(1) 020030 104404      TYPE      ,SENULL   ;;TYPE A NULL CHARACTER
(1) 020032 104400 020077      $GET42:
(1) 020036      MOV      #42,R0     ;;GET MONITOR ADDRESS
(1) 020042 001405      BEQ      $DOAGN     ;;BRANCH IF NO MONITOR
(1) 020044 000005      RESET    ;;CLEAR THE WORLD
(1) 020046 004710      SENDAD: JSR      PC,(R0) ;;GO TO MONITOR
(1) 020050 000240      NOP      ;;SAVE ROOM
(1) 020052 000240      NOP      ;;FOR
(1) 020054 000240      NOP      ;;ACT11
(1) 020056      SDOAGN:
(1) 020056 000137 002506      JMP      #START2    ;;RETURN
(1) 020062 005015 047105 020104 SENDMG: .ASCIZ  <15><12>/END PASS #/
(1) 020070 040520 051523 021440
(1) 020076      $NULL: .BYTE  -1,-1,0 ;;NULL CHARACTER STRING
(1) 020077      377      377      000
4985
(1) ;*****

```

```

(1) .SBTTL SCOPE HANDLER ROUTINE
(1)
(1) ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
(1) ;*AND LOAD THE TEST NUMBER(STSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
(1) ;*AND LOAD THE ERROR FLAG (SERFLG) INTO DISPLAY<15:08>
(1) ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
(1) ;*SW14=1 LOOP ON TEST
(1) ;*SW11=1 INHIBIT ITERATIONS
(1) ;*SW09=1 LOOP ON ERROR
(1) ;*SW08=1 LOOP ON TEST IN SMR<7:0>
(1) ;*CALL
(1) ;* SCOPE ;;SCOPE=IOT
(1)
(1) $SCOPE:
(3) 020102 005067 005036 CLR LINE ;INIT THE LINE NO. TO ZERO
(3) 020106 016701 004266 MOV DHADR,R1 ;SET UP DEVRDR IN R1
(1) 020112 032777 040000 161016 15: BIT #BIT14,2SMR ;LOOP ON PRESENT TEST?
(1) 020120 001114 BNE SOVER ;YES IF SW14=1
(1) ;*****START OF CODE FOR THE XOR TESTER*****
(1) 020122 000416 $XTSTR: BR 6S ;IF RUNNING ON THE "XOR" TESTER CHANGE
(1) ; THIS INSTRUCTION TO A "NOP" (NOP=240)
(1) 020124 013746 000004 MOV 2#ERRVEC,-(SP) ;SAVE THE CONTENTS OF THE ERROR VECTOR
(1) 020130 012737 020150 000004 MOV #5S,2#ERRVEC ;SET FOR TIMEOUT
(1) 020136 005737 177060 TST 2#177060 ;TIME OUT ON XOR?
(1) 020142 012637 000004 MOV (SP)+,2#ERRVEC ;RESTORE THE ERROR VECTOR
(1) 020146 000463 BR $SVLAD ;GO TO THE NEXT TEST
(1) 020150 022626 5S: CMP (SP)+,(SP)+ ;CLEAR THE STACK AFTER A TIME OUT
(1) 020152 012637 000004 MOV (SP)+,2#ERRVEC ;RESTORE THE ERROR VECTOR
(1) 020156 000423 BR 7S ;LOOP ON THE PRESENT TEST
(1) 020160 6S:;*****END OF CODE FOR THE XOR TESTER*****
(1) 020160 032777 000400 160750 BIT #BIT08,2SMR ;LOOP ON SPEC. TEST?
(1) 020166 001404 BEQ 2S ;BR IF NO
(1) 020170 127767 160742 160704 CMPCB 2SMR,STSTNM ;ON THE RIGHT TEST? SMR<7:0>
(1) 020176 001465 BEQ SOVER ;BR IF YES
(1) 020200 105767 160677 2S: TSTB SERFLG ;HAS AN ERROR OCCURRED?
(1) 020204 001421 BEQ 3S ;BR IF NO
(1) 020206 126767 160703 160667 CMPCB SERMAX,SERFLG ;MAX. ERRORS FOR THIS TEST OCCURRED?
(1) 020214 101015 BHI 3S ;BR IF NO
(1) 020216 032777 001000 160712 BIT #BIT09,2SMR ;LOOP ON ERROR?
(1) 020224 001404 BEQ 4S ;BR IF NO
(1) 020226 016767 160656 160652 7S: MOV $LPERR,$LPADR ;SET LOOP ADDRESS TO LAST SCOPE
(1) 020234 000446 BR SOVER
(1) 020236 105067 160641 4S: CLRB SERFLG ;ZERO THE ERROR FLAG
(1) 020242 005067 160752 CLR $TIMES ;CLEAR THE NUMBER OF ITERATIONS TO MAKE
(1) 020246 000415 BR 1S ;ESCAPE TO THE NEXT TEST
(1) 020250 032777 004000 160660 3S: BIT #BIT11,2SMR ;INHIBIT ITERATIONS?
(1) 020256 001011 BNE 1S ;BR IF YES
(1) 020260 005767 160752 TST $PASS ;IF FIRST PASS OF PROGRAM
(1) 020264 001406 BEQ 1S ; INHIBIT ITERATIONS
(1) 020266 005267 160612 INC $ICNT ;INCREMENT ITERATION COUNT
(1) 020272 026767 160722 160604 CMPCB $TIMES,$ICNT ;CHECK THE NUMBER OF ITERATIONS MADE
(1) 020300 002024 BGE SOVER ;BR IF MORE ITERATION REQUIRED
(1) 020302 012767 000001 160574 1S: MOV #1,$ICNT ;REINITIALIZE THE ITERATION COUNTER
(1) 020310 016767 000052 160702 MOV $MXCNT,$TIMES ;SET NUMBER OF ITERATIONS TO DO

```

```

(1) 020316 105267 160560
(1) 020322 116767 160554 160704
(1) 020330 011667 160552
(1) 020334 011667 160550
(1) 020340 005067 160556
(1) 020344 112767 000001 160543
(1) 020352 016777 160524 160560
(1) 020360 016716 160522
(1) 020364 000002
(1) 020366 000010

```

```

SSVLAD: INCB $STSTM          ;; COUNT TEST NUMBERS
        MOV  $STSTM,$STSTM  ;; SET TEST NUMBER IN APT MAILBOX
        MOV  (SP),SLPADR    ;; SAVE SCOPE LOOP ADDRESS
        MOV  (SP),SLPERR    ;; SAVE ERROR LOOP ADDRESS
        CLR  $ESCAPE        ;; CLEAR THE ESCAPE FROM ERROR ADDRESS
        MOV  $1,$SERMAX     ;; ONLY ALLOW ONE(1) ERROR ON NEXT TEST
SOVER:  MOV  $STSTM,$DISPLAY ;; DISPLAY TEST NUMBER
        MOV  $SLPADR,(SP)   ;; FUDGE RETURN ADDRESS
        RTI                  ;; FIXES PS
SMXCNT: 10                  ;; MAX. NUMBER OF ITERATIONS
;*****

```

4986

.SBTTL ERROR HANDLER ROUTINE

```

;#THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
;#SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
;#AND GO TO SERRTYP ON ERROR
;#THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
;#SW15=1 HALT ON ERROR
;#SW13=1 INHIBIT ERROR TYPEOUTS
;#SW09=1 LOOP ON ERROR
;#CALL   ERROR N ;;ERROR=ENT AND N=ERROR ITEM NUMBER

```

```

(1) 020370
(1) 020370 105267 160507
(1) 020374 001775
(1) 020376 016777 160500 160534
(1) 020404 005267 160502
(1) 020410 011667 160502
(1) 020414 162767 000002 160474
(1) 020422 117767 160470 160464
(1) 020430 032777 020000 160500
(1) 020436 001004
(1) 020440 004767 000072
(1) 020444 104400 001225
(1) 020450
(1) 020450 122767 000001 160572
(1) 020456 001007
(1) 020460 116767 160430 000004
(1) 020466 004767 001146
(1) 020472 000
(1) 020473 000
(1) 020474 000777
(1) 020476 005777 160434
(1) 020502 100001
(1) 020504 000000
(1) 020506 032777 001000 160422
(1) 020514 001402
(1) 020516 016716 160366
(1) 020522 005767 160474
(1) 020526 001402
(1) 020530 016716 160466
(1) 020534

```

```

ERROR: 7S: INCB $SERFLG          ;; SET THE ERROR FLAG
        BEQ  7S             ;; DON'T LET THE FLAG GO TO ZERO
        MOV  $STSTM,$DISPLAY ;; DISPLAY TEST NUMBER AND ERROR FLAG
        INC  $ERTTL         ;; INC THE ERROR COUNT
        MOV  (SP),SERRPC    ;; GET ADDRESS OF ERROR INSTRUCTION
        SUB  $2,$SERRPC     ;; STRIP AND SAVE THE ERROR ITEM CODE
        MOV  $SERRPC,$ITEMB ;; SKIP TYPEOUT IF SET
        BIT  $BIT13,$SMR    ;; SKIP TYPEOUTS
        BNE  20S           ;; GO TO USER ERROR ROUTINE
        JSR  PC,SERRTYP
        TYPE ,SCRLF

20S:    CMPB  $APTENV,$ENV   ;; RUNNING IN APT MODE
        BNE  2S             ;; NO SKIP APT ERROR REPORT
        MOV  $ITEMB,$21S    ;; SET ITEM NUMBER AS ERROR NUMBER
        JSR  PC,$SATY4     ;; REPORT FATAL ERROR TO APT

21S:    .BYTE 0
        .BYTE 0

22S:    BR   22S           ;; APT ERROR LOOP
23S:    TST  $SMR          ;; HALT ON ERROR
        BPL  3S           ;; SKIP IF CONTINUE
        HALT              ;; HALT ON ERROR!

3S:    BIT  $BIT09,$SMR    ;; LOOP ON ERROR SWITCH SET?
        BEQ  4S           ;; BR IF NO
        MOV  $SLPERR,(SP)  ;; FUDGE RETURN FOR LOOPING
        TST  $ESCAPE       ;; CHECK FOR AN ESCAPE ADDRESS
        BEQ  5S           ;; BR IF NONE
        MOV  $ESCAPE,(SP)  ;; FUDGE RETURN ADDRESS FOR ESCAPE

5S:

```



```

(1) #OCTAL (ASCII) NUMBER AND TYPE IT.
(1) #STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
(1) #CALL:
(1) #      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
(1) #      TYPOS    ;;CALL FOR TYPEOUT
(1) #      .BYTE   N              ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
(1) #      .BYTE   M              ;;M=1 OR 0
(1) #                               ;;1=TYPE LEADING ZEROS
(1) #                               ;;0=SUPPRESS LEADING ZEROS
(1) #STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
(1) #STYPOS OR STYPOC
(1) #CALL:
(1) #      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
(1) #      TYPON   ;;CALL FOR TYPEOUT
(1) #STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
(1) #CALL:
(1) #      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
(1) #      TYPOC   ;;CALL FOR TYPEOUT
(1) 020672 017646 000000          STYPOS: MOV      2(SP),-(SP)      ;; PICKUP THE MODE
(1) 020676 116667 000001 000211  MOVB    1(SP),SOFILL    ;; LOAD ZERO FILL SWITCH
(1) 020704 112667 000207          MOVB    (SP)+,SOMODE+1  ;; NUMBER OF DIGITS TO TYPE
(1) 020710 062716 000002          ADD     #2,(SP)        ;; ADJUST RETURN ADDRESS
(1) 020714 000406          BR      STYPON
(1) 020716 112767 000001 000171  STYPOC: MOVB    #1,SOFILL    ;; SET THE ZERO FILL SWITCH
(1) 020724 112767 000006 000165  MOVB    #6,SOMODE+1    ;; SET FOR SIX(6) DIGITS
(1) 020732 112767 000005 000154  STYPON: MOVB    #5,SOCNT    ;; SET THE ITERATION COUNT
(1) 020740 010346          MOV     R3,-(SP)      ;; SAVE R3
(1) 020742 010446          MOV     R4,-(SP)      ;; SAVE R4
(1) 020744 010546          MOV     R5,-(SP)      ;; SAVE R5
(1) 020746 116704 000145          MOVB    SOMODE+1,R4    ;; GET THE NUMBER OF DIGITS TO TYPE
(1) 020752 005404          NEG     R4
(1) 020754 062704 000006          ADD     #6,R4         ;; SUBTRACT IT FOR MAX. ALLOWED
(1) 020760 110467 000132          MOVB    R4,SOMODE     ;; SAVE IT FOR USE
(1) 020754 116704 000125          MOVB    SOFILL,R4     ;; GET THE ZERO FILL SWITCH
(1) 020770 016606 000012          MOV     12(SP),R5    ;; PICKUP THE INPUT NUMBER
(1) 020774 005003          CLR     R3           ;; CLEAR THE OUTPUT WORD
(1) 020776 006105          1S:    ROL     R5     ;; ROTATE MSB INTO "C"
(1) 021000 000404          BR      3S          ;; GO DO MSB
(1) 021002 006105          2S:    ROL     R5     ;; FORM THIS DIGIT
(1) 021004 006105          ROL     R5
(1) 021006 006105          ROL     R5
(1) 021010 010503          MOV     R5,R3
(1) 021012 006103          3S:    ROL     R3     ;; GET LSB OF THIS DIGIT
(1) 021014 105367 000076          DECB   SOMODE        ;; TYPE THIS DIGIT?
(1) 021020 100016          BPL    7S           ;; BR IF NO
(1) 021022 042703 177770          BIC    #177770,R3    ;; GET RID OF JUNK
(1) 021026 001002          BNE    4S           ;; TEST FOR 0
(1) 021030 005704          TST    R4           ;; SUPPRESS THIS 0?
(1) 021032 001403          BEQ    5S           ;; BR IF YES
(1) 021034 005204          4S:    INC    R4     ;; DON'T SUPPRESS ANYMORE 0'S
(1) 021036 052703 000060          BIS    #'0,R3      ;; MAKE THIS DIGIT ASCII

```

(1)	021042	052703	000040	5S:	BIS	#' R3	:: MAKE ASCII IF NOT ALREADY
(1)	021044	110367	000040		MOV	R3, BS	:: SAVE FOR TYPING
(1)	021052	104400	021112		TYPE	BS	:: GO TYPE THIS DIGIT
(1)	021056	105367	000032	7S:	DECB	\$OCNT	:: COUNT BY 1
(1)	021062	003347			BGT	2S	:: BR IF MORE TO DO
(1)	021064	002402			BLT	6S	:: BR IF DONE
(1)	021066	005204			INC	R4	:: INSURE LAST DIGIT ISN'T A BLANK
(1)	021070	000744			BR	2S	:: GO DO THE LAST DIGIT
(1)	021072	012605		6S:	MOV	(SP)+, R5	:: RESTORE R5
(1)	021074	012604			MOV	(SP)+, R4	:: RESTORE R4
(1)	021076	012603			MOV	(SP)+, R3	:: RESTORE R3
(1)	021100	016666	000002 000004		MOV	2(SP), 4(SP)	:: SET THE STACK FOR RETURNING
(1)	021106	012616			MOV	(SP)+, (SP)	
(1)	021110	000002			RTI		:: RETURN
(1)	021112	000		8S:	.BYTE	0	:: STORAGE FOR ASCII DIGIT
(1)	021113	000			.BYTE	00	:: TERMINATOR FOR TYPE ROUTINE
(1)	021114	000		\$OCNT:	.BYTE	00	:: OCTAL DIGIT COUNTER
(1)	021115	000		\$OFILL:	.BYTE	0	:: ZERO FILL SWITCH
(1)	021116	000000		\$OMODE:	.WORD	0	:: NUMBER OF DIGITS TO TYPE

4989

.SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

:: THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
 :: SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
 :: NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
 :: BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
 :: REPLACED WITH SPACES.

:: CALL:
 :: # MOV NUM, -(SP) :: PUT THE BINARY NUMBER ON THE STACK
 :: # TYPDS :: GO TO THE ROUTINE

STYPDS:
 MOV R0, -(SP) :: PUSH R0 ON STACK
 MOV R1, -(SP) :: PUSH R1 ON STACK
 MOV R2, -(SP) :: PUSH R2 ON STACK
 MOV R3, -(SP) :: PUSH R3 ON STACK
 MOV R5, -(SP) :: PUSH R5 ON STACK
 MOV #20200, -(SP) :: SET BLANK SWITCH AND SIGN
 MOV 20(SP), R5 :: GET THE INPUT NUMBER
 BPL 1S :: BR IF INPUT IS POS.
 NEG R5 :: MAKE THE BINARY NUMBER POS.
 MOV #'-, 1(SP) :: MAKE THE ASCII NUMBER NEG.
 1S: CLR R0 :: ZERO THE CONSTANTS INDEX
 MOV #SDBLK, R3 :: SETUP THE OUTPUT POINTER
 MOV #', (R3)+ :: SET THE FIRST CHARACTER TO A BLANK
 2S: CLR R2 :: CLEAR THE BCD NUMBER
 MOV \$DTBL(R0), R1 :: GET THE CONSTANT
 3S: SUB R1, R5 :: FORM THIS BCD DIGIT
 BLT 4S :: BR IF DONE
 INC R2 :: INCREASE THE BCD DIGIT BY 1
 BR 3S
 4S: ADD R1, R5 :: ADD BACK THE CONSTANT
 TST R2 :: CHECK IF BCD DIGIT=0

(1)	021120						
(1)	021120	010046					
(3)	021122	010146					
(3)	021124	010246					
(3)	021126	010346					
(3)	021130	010546					
(1)	021132	012746	020200				
(1)	021136	016605	000020				
(1)	021142	100004					
(1)	021144	005405					
(1)	021146	112766	000055 000001				
(1)	021154	005000		1S:	CLR	R0	
(1)	021156	012703	021334		MOV	#SDBLK, R3	
(1)	021162	112723	000040		MOV	#', (R3)+	
(1)	021166	005002		2S:	CLR	R2	
(1)	021170	016001	021324		MOV	\$DTBL(R0), R1	
(1)	021174	160105		3S:	SUB	R1, R5	
(1)	021176	002402			BLT	4S	
(1)	021200	005202			INC	R2	
(1)	021202	000774			BR	3S	
(1)	021204	060105		4S:	ADD	R1, R5	
(1)	021206	005702			TST	R2	


```

(1) 021210 001002      BNE      5$      ; FALL THROUGH IF 0
(1) 021212 105716      TSTB     (SP)    ; STILL DOING LEADING 0'S?
(1) 021214 100407      BMI      7$      ; BR IF YES
(1) 021216 106316      5$: ASLB     (SP)    ; MSD?
(1) 021220 103003      BCC      6$      ; BR IF NO
(1) 021222 116663      MOVB     1(SP),-1(R3) ; YES--SET THE SIGN
(1) 021230 052702      6$: BIS      0'D,R2 ; MAKE THE BCD DIGIT ASCII
(1) 021234 052702      7$: BIS      0'R2   ; MAKE IT A SPACE IF NOT ALREADY A DIGIT
(1) 021240 110223      MOVB     R2,R3)+ ; PUT THIS CHARACTER IN THE OUTPUT BUFFER
(1) 021242 005720      TST      (R0)+  ; JUST INCREMENTING
(1) 021244 020027      CMP      R0,#10 ; CHECK THE TABLE INDEX
(1) 021250 002746      BLT      2$      ; GO DO THE NEXT DIGIT
(1) 021252 003002      BGT      8$      ; GO TO EXIT
(1) 021254 010502      MOV      R5,R2   ; GET THE LSD
(1) 021256 000764      BR       6$      ; GO CHANGE TO ASCII
(1) 021260 105726      8$: TSTB     (SP)+ ; WAS THE LSD THE FIRST NON-ZERO?
(1) 021262 100003      BPL      9$      ; BR IF NO
(1) 021264 116663      MOVB     -1(SP),-2(R3) ; YES--SET THE SIGN FOR TYPING
(1) 021272 105013      9$: CLRB     (R3)   ; SET THE TERMINATOR
(3) 021274 012605      MOV      (SP)+,R5 ; POP STACK INTO R5
(3) 021276 012603      MOV      (SP)+,R3 ; POP STACK INTO R3
(3) 021300 012602      MOV      (SP)+,R2 ; POP STACK INTO R2
(3) 021302 012601      MOV      (SP)+,R1 ; POP STACK INTO R1
(3) 021304 012600      MOV      (SP)+,R0 ; POP STACK INTO R0
(1) 021306 104400      TYPE     SDBLK   ; NOW TYPE THE NUMBER
(1) 021312 016666      MOV      2(SP),4(SP) ; ADJUST THE STACK
(1) 021320 012616      MOV      (SP)+,(SP)
(1) 021322 000002      RTI                      ; RETURN TO USER
(1) 021324 023420      SDBLK: 10000.
(1) 021326 001750      1000.
(1) 021330 000144      100.
(1) 021332 000012      10.
(1) 021334 000004      SDBLK: .BLKW 4
;*****
4990 (1) .SBTTL TYPE ROUTINE
(1) ;#ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
(1) ;#THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
(1) ;#NOTE1: SNULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
(1) ;#NOTE2: SFILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
(1) ;#NOTE3: SFILLC CONTAINS THE CHARACTER TO FILL AFTER.
(1) ;
(1) ;#CALL:
(1) ;#1) USING A TRAP INSTRUCTION
(1) ; TYPE ,MESADR ; MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
(1) ;#OR
(1) ; TYPE
(1) ; MESADR
(1) ;
(1) 021344 105767 157605 STYPE: TSTB STPFLG ; IS THERE A TERMINAL?
(1) 021350 100002 BPL IS ; BR IF YES
(1) 021352 000000 HALT ; HALT HERE IF NO TERMINAL

```

```

(1) 021354 000430          BR      3$          ;; LEAVE
(1) 021356 010046          1$: MOV      RO,-(SP)  ;; SAVE RO
(1) 021360 017600 000002  MOV      22(SP),RO  ;; GET ADDRESS OF ASCIZ STRING
(1) 021364 122767 000001 157656  CMPB    #APTENV,SENV ;; RUNNING IN APT MODE
(1) 021372 001011          BNE     62$        ;; NO GO CHECK FOR APT CONSOLE
(1) 021374 132767 000100 157647  BITB    #APTPOOL,SENVM ;; SPOOL MESSAGE TO APT
(1) 021402 001405          BEQ     62$        ;; NO GO CHECK FOR CONSOLE
(1) 021404 010067 000004  MOV      RO,61$    ;; SETUP MESSAGE ADDRESS FOR APT
(1) 021410 004767 000214  JSR     PC,SATY3   ;; SPOOL MESSAGE TO APT
(1) 021414 000000          .WORD   0          ;; MESSAGE ADDRESS
(1) 021416 132767 000040 157625  61$: BITB    #APTCSUP,SENVM ;; APT CONSOLE SUPPRESSED
(1) 021424 001003          BNE     60$        ;; YES, SKIP TYPE OUT
(1) 021426 112046          2$: MOVB   (RO)+,-(SP) ;; PUSH CHARACTER TO BE TYPED ONTO STACK
(1) 021430 001005          BNE     4$         ;; BR IF IT ISN'T THE TERMINATOR
(1) 021432 005726          TST    (SP)+      ;; IF TERMINATOR POP IT OFF THE STACK
(1) 021434 012600          60$: MOV      (SP)+,RO ;; RESTORE RO
(1) 021436 062716 000002  3$: ADD     #2,(SP)  ;; ADJUST RETURN PC
(1) 021442 000002          RTI                    ;; RETURN
(1) 021444 122716 000011  4$: CMPB    #HT,(SP)  ;; BRANCH IF <HT>
(1) 021450 001426          BEQ     8$         ;; BRANCH IF NOT <CRLF>
(1) 021452 122716 000200  5$: CMPB    #TCRLF,(SP) ;; BRANCH IF NOT <CRLF>
(1) 021456 001004          BNE     5$        ;; POP <CR><LF> EQUIV
(1) 021460 005726          TST    (SP)+      ;; TYPE A CR AND LF
(1) 021462 104400          TYPE                    ;;
(1) 021464 001225          SCRLF                    ;;
(1) 021466 000757          BR      2$        ;; GET NEXT CHARACTER
(1) 021470 004767 000056  5$: JSR     PC,STYPEC  ;; GO TYPE THIS CHARACTER
(1) 021474 126726 157454  6$: CMPB    #FILLC,(SP)+ ;; IS IT TIME FOR FILLER CHARS.?
(1) 021500 001352          BNE     2$        ;; IF NO GO GET NEXT CHAR.
(1) 021502 016746 157444  2$: MOV      #NULL,-(SP) ;; GET # OF FILLER CHARS. NEEDED
(1) 021506 105366 000001  7$: DECB    1(SP)    ;; AND THE NULL CHAR.
(1) 021512 002770          BLT    6$         ;; DOES A NULL NEED TO BE TYPED?
(1) 021514 004767 000032  6$: JSR     PC,STYPEC  ;; BR IF NO--GO POP THE NULL OFF OF STACK
(1) 021520 105367 000072  7$: DECB    #CHARCNT ;; GO TYPE A NULL
(1) 021524 000770          BR      7$        ;; DO NOT COUNT AS A COUNT
(1)                                ;; LOOP
(1)                                ;HORIZONTAL TAB PROCESSOR
(1) 021526 112716 000040  8$: MOVB   #40,(SP)  ;; REPLACE TAB WITH SPACE
(1) 021532 004767 000014  9$: JSR     PC,STYPEC  ;; TYPE A SPACE
(1) 021536 132767 000007 000052  BITB    #7,#CHARCNT ;; BRANCH IF NOT AT
(1) 021544 001372          BNE     9$        ;; TAB STOP
(1) 021546 005726          TST    (SP)+      ;; POP SPACE OFF STACK
(1) 021550 000726          BR      2$        ;; GET NEXT CHARACTER
(1) 021552 105777 157370  STYPEC: TSTB   #STPS  ;; WAIT UNTIL PRINTER IS READY
(1) 021556 100375          BPL    STYPEC     ;;
(1) 021560 116677 000002 157362  MOVB   2(SP),#STPB ;; LOAD CHAR TO BE TYPED INTO DATA REG.
(1) 021566 122766 000015 000002  CMPB    #15,2(SP)  ;; BRANCH IF
(1) 021574 001003          BNE     1$        ;; NOT <CR>
(1) 021576 105067 000014          CLRB   #CHARCNT  ;;
(1) 021602 000406          BR      STYPEX    ;; EXIT
(1) 021604 122766 000012 000002  1$: CMPB    #12,2(SP) ;; BRANCH IF
(1) 021612 002002          BGE    STYPEX    ;; <LF>

```

```

(1) 021614 105227          INCB      (PC)+      ;; INC SPACE
(1) 021616 000000          $CHARCNT: WORD 0      ;; COUNT
(1) 021620 000207          $TYPEX: RTS      PC
(1)                                ;; EQUATES
(1)                                TAT=11
(1)                                TCRLF=200
(1)
4991 ;*****
(1)                                .SBTTL  APT COMMUNICATIONS ROUTINE
(1) 021622 112767 000001 000236 $SATY1: MOV      #1, SFFLG      ;TO REPORT FATAL ERROR
(1) 021630 112767 000001 000226 $SATY3: MOV      #1, SMFLG      ;TO TYPE A MESSAGE
(1) 021636 000403          BR          SATYC
(1) 021640 112767 000001 000220 $SATY4: MOV      #1, SFFLG      ;TO ONLY REPORT FATAL ERROR
(2) 021646          $SATYC:
(3) 021646 010046          MOV      R0, -(SP)      ;; PUSH R0 ON STACK
(3) 021650 010146          MOV      R1, -(SP)      ;; PUSH R1 ON STACK
(1) 021652 105767 000206          TSTB     SMFLG          ;; SHOULD TYPE A MESSAGE?
(1) 021656 001450          BEQ      5$            ;; IF NOT: BR
(1) 021660 122767 000001 157362          CMPB     @APTENV, SENV   ;; OPERATING UNDER APT?
(1) 021666 001031          BNE     3$            ;; IF NOT: BR
(1) 021670 132767 000100 157353          BITB     @APTPOOL, SENVM ;; SHOULD SPOOL MESSAGES?
(1) 021676 001425          BEQ      3$            ;; IF NOT: BR
(1) 021700 017600 000004          MOV      @4(SP), R0      ;; GET MESSAGE ADDR.
(1) 021704 062766 000002 000004          ADD      #2, 4(SP)      ;; BUMP RETURN ADDR.
(1) 021712 005767 157312 1$:          TST      $MSGTYPE      ;; SEE IF DONE W/ LAST XMISSION?
(1) 021716 001375          BNE     1$            ;; IF NOT: WAIT
(1) 021720 010067 157320          MOV      R0, $MSGAD      ;; PUT ADDR IN MAILBOX
(1) 021724 105720 2$:          TSTB     (R0)+          ;; FIND END OF MESSAGE
(1) 021726 001376          BNE     2$            ;;
(1) 021730 166700 157310          SUB      $MSGAD, R0      ;; SUB START OF MESSAGE
(1) 021734 006200          ASR      R0            ;; GET MESSAGE LGTH IN WORDS
(1) 021736 010067 157304          MOV      R0, $MSGGLT     ;; PUT LENGTH IN MAILBOX
(1) 021742 012767 000004 157260          MOV      #4, $MSGTYPE    ;; TELL APT TO TAKE MSG.
(1) 021750 000413          BR          5$
(1) 021752 017667 000004 000016 3$:          MOV      @4(SP), 4$      ;; PUT MSG ADDR IN JSR LINKAGE
(1) 021760 062766 000002 000004          ADD      #2, 4(SP)      ;; BUMP RETURN ADDRESS
(3) 021766 016746 156004          MOV      177776, -(SP)   ;; PUSH 177776 ON STACK
(1) 021772 004767 177346          JSR      PC, $TYPE      ;; CALL TYPE MACRO
(1) 021776 000000          .WORD    0
(1) 022000          5$:
(1) 022000 105767 000062          10$: TSTB     SFFLG          ;; SHOULD REPORT FATAL ERROR?
(1) 022004 001416          BEQ      12$          ;; IF NOT: BR
(1) 022006 005767 157236          TST      SENV          ;; RUNNING UNDER APT?
(1) 022012 001413          BEQ      12$          ;; IF NOT: BR
(1) 022014 005767 157210          11$: TST      $MSGTYPE      ;; FINISHED LAST MESSAGE?
(1) 022020 001375          BNE     11$          ;; IF NOT: WAIT
(1) 022022 017667 000004 157202          MOV      @4(SP), $FATAL  ;; GET ERROR #
(1) 022030 062766 000002 000004          ADD      #2, 4(SP)      ;; BUMP RETURN ADDR.
(1) 022036 005267 157166          INC      $MSGTYPE      ;; TELL APT TO TAKE ERROR
(1) 022042 105067 000020          12$: CLRB     SFFLG          ;; CLEAR FATAL FLAG
(1) 022046 105067 000013          CLRB     $LFLG          ;; CLEAR LOG FLAG
(1) 022052 105067 000006          CLRB     SMFLG          ;; CLEAR MESSAGE FLAG
(3) 022056 012601          MOV      (SP)+, R1      ;; POP STACK INTO R1

```

```

(3) 022060 012600          MOV      (SP)+,R0          ;; POP STACK INTO R0
(1) 022062 0002C7          RTS      PC              ;; RETURN
(1)
(1) 022064      000          SMFLG: .BYTE 0          ;; MESSG. FLAG
(1) 022065      000          $LFLG: .BYTE 0          ;; LOG FLAG
(1) 022066      000          $FFLG: .BYTE 0          ;; FATAL FLAG
(1)
(1)      022070          .EVEN
(1)      000200          APTSIZE=200
(1)      000001          APTENV=001
(1)      000100          APTSPool=100
(1)      000040          APTCSUP=040
4992 *****
(1)
(1)      .SBTTL TTY INPUT ROUTINE
(1)
(1)      ;; THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
(1)      ;; CALL:
(1)      ;;      R0CHR          ;; INPUT A SINGLE CHARACTER FROM THE TTY
(1)      ;;      RETURN HERE    ;; CHARACTER IS ON THE STACK
(1)
(1) 022070 011646          SRDCHR: MOV      (SP),-(SP)      ;; PUSH DOWN THE PC
(1) 022072 016666 000004 000002  MOV      4(SP),2(SP)      ;; SAVE THE PS
(1) 022100 105777 157036  IS:      TSTB      2(STK)      ;; WAIT FOR
(1) 022104 100375          BPL      15              ;; A CHARACTER
(1) 022106 117766 157032 000004  MOVB     2(STK),4(SP)      ;; READ THE TTY
(1) 022114 042766 177600 000004  BIC      8(C<177>),4(SP)  ;; GET RID OF JUNK IF ANY
(1) 022122 000002          RTI                    ;; GO BACK TO USER
(1) *****
(1)      ;; THIS ROUTINE WILL INPUT A STRING FROM THE TTY
(1)      ;; CALL:
(1)      ;;      RDLIN          ;; INPUT A STRING FROM THE TTY
(1)      ;;      RETURN HERE    ;; ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
(1)      ;;      ;;            ;; TERMINATOR WILL BE A BYTE OF ALL 0'S
(1)
(1) 022124 010346          SRDLIN: MOV      R3, -(SP)      ;; SAVE R3
(1) 022126 012703 022232  IS:      MOV      #TTYIN,R3      ;; GET ADDRESS
(1) 022132 022703 022242  2S:      CMP      #TTYIN+8.,R3      ;; BUFFER FULL?
(1) 022136 101405          BLOS     45              ;; BR IF YES
(1) 022140 104405          R0CHR          ;; GO READ ONE CHARACTER FROM THE TTY
(1) 022142 112613          MOVB     (SP)+,(R3)      ;; GET CHARACTER
(1) 022144 122713 000177  CMPB     8(177),(R3)      ;; IS IT A RUBOUT
(1) 022150 001003          BNE     3S              ;; SKIP IF NOT
(1) 022152 104400 001224  4S:      TYPE     8QUES      ;; TYPE A '?'
(1) 022156 000763          BR      15              ;; CLEAR THE BUFFER AND LOOP
(1) 022160 111367 000044  3S:      MOVB     (R3),9S      ;; ECHO THE CHARACTER
(1) 022164 104400 022230          TYPE     9S
(1) 022170 122723 000015  CMPB     8(15),(R3)+    ;; CHECK FOR RETURN
(1) 022174 001356          BNE     2S              ;; LOOP IF NOT RETURN
(1) 022176 105063 177777  CLRB     -1(R3)          ;; CLEAR RETURN (THE 15)
(1) 022202 104400 001226          TYPE     8LF           ;; TYPE A LINE FEED
(1) 022206 012603          MOV      (SP)+,R3      ;; RESTORE R3
(1) 022210 011646          MOV      (SP),-(SP)     ;; ADJUST THE STACK AND PUT ADDRESS OF THE
(1) 022212 016666 000004 000002  MOV      4(SP),2(SP)    ;; FIRST ASCII CHARACTER ON IT

```

```
(1) 022220 012766 022232 000004      MOV      #STTYIN,4(SP)
(1) 022226 000002                      RTI      ;: RETURN
(1) 022230 000          9S: .BYTE 0          ;: STORAGE FOR ASCII CHAR. TO TYPE
(1) 022231 000          .BYTE 0          ;: TERMINATOR
(1) 022232 000010      STTYIN: .BLKB 8. ;: RESERVE 8 BYTES FOR TTY INPUT
4993 ;:*****
(1) .SBTTL READ AN OCTAL NUMBER FROM THE TTY
(1) ;:THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
(1) ;:CHANGE IT TO BINARY.
(1) ;:THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
(1) ;:OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
(1) ;:FOLLOWED BY A CARRIAGE RETURN-LINE FEED, THE COMPLETE NUMBER MUST
(1) ;:THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
(1) ;:CALL:
(1) ;:      RDOCT          ;: READ AN OCTAL NUMBER
(1) ;:      RETURN HERE   ;: LOW ORDER BITS ARE ON TOP OF THE STACK
(1) ;:                  ;: HIGH ORDER BITS ARE IN SHIOCT
(1) 022242 011646 000004 000002 SRDOCT: MOV      (SP),-(SP) ;: PROVIDE SPACE FOR THE
(1) 022244 016666          MOV      4(SP),2(SP) ;: INPUT NUMBER
(3) 022252 010046          MOV      R0,-(SP)    ;: PUSH R0 ON STACK
(3) 022254 010146          MOV      R1,-(SP)    ;: PUSH R1 ON STACK
(3) 022256 010246          MOV      R2,-(SP)    ;: PUSH R2 ON STACK
(1) 022260 104406          1S: RDLIN ;: READ AN ASCII LINE
(1) 022262 012600          MOV      (SP)+,R0    ;: GET ADDRESS OF 1ST CHARACTER
(1) 022264 010067 000100  MOV      R0,5S      ;: AND SAVE IT
(1) 022270 005001          CLR      R1          ;: CLEAR DATA WORD
(1) 022272 005002          CLR      R2
(1) 022274 112046          2S: MOV8B  (R0)+,-(SP) ;: PICKUP THIS CHARACTER
(1) 022276 001420          BEQ      3S          ;: IF ZERO GET OUT
(1) 022300 122716 000060  CMPB    #'0,(SP)    ;: MAKE SURE THIS CHARACTER
(1) 022304 003026          BGT      4S          ;: IS AN OCTAL DIGIT
(1) 022306 122716 000067  CMPB    #'7,(SP)
(1) 022312 002423          BLT      4S
(1) 022314 006301          ASL     R1           ;: #2
(1) 022316 006102          ROL     R2
(1) 022320 006301          ASL     R1           ;: #4
(1) 022322 006102          ROL     R2
(1) 022324 006301          ASL     R1           ;: #8
(1) 022326 006102          ROL     R2
(1) 022330 042716 177770  BIC     #'C7,(SP)   ;: STRIP THE ASCII JUNK
(1) 022334 062601          ADD     (SP)+,R1    ;: ADD IN THIS DIGIT
(1) 022336 000756          BR      2S          ;: LOOP
(1) 022340 005726          3S: TST     (SP)+     ;: CLEAN TERMINATOR FROM STACK
(1) 022342 010166 000012  MOV     R1,12(SP)   ;: SAVE THE RESULT
(1) 022346 010267 000026  MOV     R2,SHIOCT
(3) 022352 012602          MOV     (SP)+,R2    ;: POP STACK INTO R2
(3) 022354 012601          MOV     (SP)+,R1    ;: POP STACK INTO R1
(3) 022356 012600          MOV     (SP)+,R0    ;: POP STACK INTO R0
(1) 022360 000002          RTI      ;: RETURN
(1) 022362 005726          4S: TST     (SP)+     ;: CLEAN PARTIAL FROM STACK
(1) 022364 105010          CLRB    (R0)        ;: SET A TERMINATOR
```

(1) 022366 104400
(1) 022370 000000
(1) 022372 104400 001224
(1) 022376 000730
(1) 022400 000000
4994

SS: TYPE WORD 0 ;;TYPE UP THRU THE BAD CHAR.
TYPE SQUES ;;?" "CR" & "LF"
BR 1S ;;TRY AGAIN
SHIOCT: .WORD 0 ;;HIGH ORDER BITS GO HERE

.SBTTL TRAP DECODER

;;THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
;;AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
;;OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
;;GO TO THAT ROUTINE.

(1) 022402 010046
(1) 022404 016600 000002
(1) 022410 005740
(1) 022412 111000
(1) 022414 006300
(1) 022416 016000 022424
(1) 022422 000200

STRAP: MOV RO, -(SP) ;;SAVE RO
MOV 2(SP), RO ;;GET TRAP ADDRESS
TST -(RO) ;;BACKUP BY 2
MOV (RO), RO ;;GET RIGHT BYTE OF TRAP
ASL RO ;;POSITION FOR INDEXING
MOV STRPAD(RO), RO ;;INDEX TO TABLE
RTS RO ;;GO TO ROUTINE

.SBTTL TRAP TABLE

;;THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
;;BY THE "TRAP" INSTRUCTION.

ROUTINE

(3) 022424
(3) 022424 021344
(3) 022426 020716
(3) 022430 020672
(3) 022432 020732
(3) 022434 021120
(3) 022436 022070
(3) 022440 022124
(3) 022442 022242
4995

STRPAL: STYPE ;;CALL=TYPE TRAP+0(104400) TTY TYPEOUT ROUTINE
STYPOC ;;CALL=TYPOC TRAP+1(104401) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
STYPOS ;;CALL=TYPOS TRAP+2(104402) TYPE OCTAL NUMBER (NO LEADING ZEROS)
STYPON ;;CALL=TYPON TRAP+3(104403) TYPE OCTAL NUMBER (AS PER LAST CALL)
STYPLD ;;CALL=TYPDS TRAP+4(104404) TYPE DECIMAL NUMBER (WITH SIGN)
SRDCHR ;;CALL=RDCHR TRAP+5(104405) TTY TYPEIN CHARACTER ROUTINE
SRDLIN ;;CALL=RDLIN TRAP+6(104406) TTY TYPEIN STRING ROUTINE
SRDOCT ;;CALL=RDOCT TRAP+7(104407) READ AN OCTAL NUMBER FROM TTY

.SBTTL POWER DOWN AND UP ROUTINES

:POWER DOWN ROUTINE

(1) 022444 012737 022572 000024
(1) 022452 012737 000340 000026
(3) 022460 010046
(3) 022462 010146
(3) 022464 010246
(3) 022466 010346
(3) 022470 010446
(3) 022472 010546
(1) 022474 010667 000076
(1) 022500 012737 022512 000024

SPWRDN: MOV #SILLUP, 2#PWRVEC ;;SET FOR FAST UP
MOV #340, 2#PWRVEC+2 ;;PRIO:7
MOV RO, -(SP) ;;PUSH RO ON STACK
MOV R1, -(SP) ;;PUSH R1 ON STACK
MOV R2, -(SP) ;;PUSH R2 ON STACK
MOV R3, -(SP) ;;PUSH R3 ON STACK
MOV R4, -(SP) ;;PUSH R4 ON STACK
MOV R5, -(SP) ;;PUSH R5 ON STACK
MOV SP, \$SAVR6 ;;SAVE SP
MOV #SPWRUP, 2#PWRVEC ;;SET UP VECTOR

```

(1) 022506 000000          HALT
(1) 022510 000776          BR      .-2          ;;HANG UP
(1)
(1)
(1)      :POWER UP ROUTINE
(1) 022512 016706 000060  $PWRUP: MOV    $SAVR6,SP      ;;GET SP
(1) 022516 005067 000054          CLR    $SAVR6          ;;WAIT LOOP FOR THE TTY
(1) 022522 005267 000050  15:   INC    $SAVR6          ;;WAIT FOR THE INC
(1) 022526 001375          BNE    15              ;;OF WORD
(3) 022530 012605          MOV    (SP)+,R5        ;;POP STACK INTO R5
(3) 022532 012604          MOV    (SP)+,R4        ;;POP STACK INTO R4
(3) 022534 012603          MOV    (SP)+,R3        ;;POP STACK INTO R3
(3) 022536 012602          MOV    (SP)+,R2        ;;POP STACK INTO R2
(3) 022540 012601          MOV    (SP)+,R1        ;;POP STACK INTO R1
(3) 022542 012600          MOV    (SP)+,R0        ;;POP STACK INTO R0
(1) 022544 012737 022444 000024  MOV    $SPWRDN,$PWRVEC ;;SET UP THE POWER DOWN VECTOR
(1) 022552 012737 000340 000026  MOV    $340,$PWRVEC+2 ;;PRIO:7
(1) 022560 104400          TYPE          ;;REPORT THE POWER FAILURE
(1) 022562 022600  $PWRMG: .WORD  $POWER          ;;POWER FAIL MESSAGE POINTER
(1) 022564 012716          MOV    (PC)+,(SP)    ;;RESTART AT RSTRTA
(1) 022566 002612  $PWRAD: .WORD  RSTRTA          ;;RESTART ADDRESS
(1) 022570 000002          RTI
(1) 022572 000000  $ILLUP: HALT
(1) 022574 000776          BR      .-2          ;;THE POWER UP SEQUENCE WAS STARTED
(1) 022576 000000  $SAVR6: 0              ;;BEFORE THE POWER DOWN WAS COMPLETE
(1) 022600 005015 047520 042527  $POWER: .ASCIZ <15><12>"POWER" ;;PUT THE SP HERE
(1) 022606 000122          .EVEN

```

```

4998 :*****
4999 :COMMON DH11 SERVICE ROUTINES
5000 :*****
5001
5002 :THIS ROUTINE IS CALLED DURING START UP TO LOAD THE XMITTER
5003 :OUTPUT BUFFER WITH A BINARY COUNT TEST PATTERN
5004
5005 022610 012701 033436 LDTBF1: MOV #TBUF,R1 ;POINT TO START OF BUFFER
5006 022614 005002 CLR R2 ;INIT DATA BYTE GENERATOR
5007 022616 110221 1S: MOVB R2,(R1)+ ;LOAD ONE CHAR
5008 022620 005202 INC R2 ;GENERATE NEXT CHAR
5009 022622 022702 000400 CMP #400,R2 ;LOADED 256(10) BYTES
5010 022626 001373 BNE 1S ;BR IF NOT
5011 022630 000207 RTS PC ;RETURN TO START TESTING
5012
5013 :THIS ROUTINE SETS UP THE ERROR INFORMATION REQUIRED BY ANY TEST
5014 :USING A "DH1" HEADER
5015
5016 022632 004767 001476 SUER1: JSR PC,SAPS ;SAVE THE ERROR PSM
5017 022636 116700 156240 MOVB STSTNM,R0 ;SAVE THE TEST NO.
5018 022642 010067 156312 MOV R0,SREG0 ;SAVE THE TEST NO. FOR ERROR PRINT
5019 022646 010167 156310 MOV R1,SREG1 ;SAVE THE DH1 ADDR
5020 022652 010267 156306 MOV R2,SREG2 ;SAVE THE REG ADDRESS
5021 022656 010667 156312 MOV R6,SREG6 ;SAVE THE SP
5022 022662 062767 000002 156304 ADD #2,SREG6 ;CORRECT FOR CALLING JSR
5023 022670 000207 RTS PC ;RETURN TO CALLING ROUTINE
5024
5025 :THIS ROUTINE IS CALLED BY THOSE TESTS USING A "DH2" HEADER TO
5026 :SAVE THE ERROR INFORMATION IN "DT2"
5027
5028 022672 004767 001436 SUER2: JSR PC,SAPS ;SAVE THE ERROR PSM
5029 022676 116700 156200 SUER2A: MOVB STSTNM,R0 ;GET THE TEST NO.
5030 022702 010067 156252 MOV R0,SREG0 ;SAVE THE REGISTERS-TEST#
5031 022706 010167 156250 MOV R1,SREG1 ;SAVE THE DH ADDRESS
5032 022712 010267 156246 MOV R2,SREG2 ;SAVE THE REGISTER ADDRESS
5033 022716 010367 156244 MOV R3,SREG3 ;SAVE THE HAS DATA
5034 022722 010467 156242 MOV R4,SREG4 ;SAVE THE S/B DATA
5035 022726 010667 156242 MOV R6,SREG6 ;SAVE THE STACK POINTER
5036 022732 062767 000002 156234 ADD #2,SREG6 ;CORRECT FOR CALLING JSR
5037 022740 000207 RTS PC ;RETURN TO REPORT ERROR
5038
5039 :THIS ROUTINE IS CALLED TO SET UP ERROR INFORMATION FOR THE
5040 :BUS ERROR AND RSVD INSTR ERROR ROUTINES
5041
5042 022742 010067 156212 SUER3: MOV R0,SREG0 ;SAVE THE REGS
5043 022746 010167 156210 MOV R1,SREG1
5044 022752 010267 156206 MOV R2,SREG2
5045 022756 000207 RTS PC ;RETURN TO REPORT ERROR
5046
5047 :THIS ROUTINE IS CALLED TO SET UP ERROR INFORMATION FOR THE
5048 :CAR/BCR MEMORY PATTERNS TESTS
5049
5050 022760 005067 156214 SUER4: CLR STMP0 ;SAVE THE LINE NO. WRITTEN
5051 022764 116767 002156 156206 MOVB LINEA,STMP0
    
```



```

5052 022772 116700 156104      MOVB   STSTNM,R0      ;SAVE THE TEST NUMBER
5053 022776 010067 156156      MOV    R0,SREG0      ;SAVE THE REGISTER INFORMATION
5054 023002 010167 156154      MOV    R1,SREG1
5055 023006 010267 156152      MOV    R2,SREG2
5056 023012 010367 156150      MOV    R3,SREG3
5057 023016 010467 156146      MOV    R4,SREG4
5058 023022 000207                RTS    PC              ;RETURN TO PATTERNS TEST

```

;THIS ROUTINE IS CALLED TO SELECT A NEW LINE NO. BASED ON THE
;VALUE OF THE LINE SELECTION PARAMETER

;CALLING SEQUENCE:

```

;JSR   PC,SELINE      ;CALL THE ROUTINE
;BR    IS              ;EXIT BRANCH-ROUTINE MOVES THE RETURN
                        ;PC AROUND THIS BR IF MORE LINES ARE
                        ;YET TO BE TESTED

```

```

5070 023024 105767 002115      SELINE: TSTB   LINE+1      ;FIRST TIME THROUGH FOR ANY TEST ?
5071 023030 001010                BNE    IS                ;BR IF NOT
5072 023032 105167 002107      COMB   LINE+1            ;SET ENTRY FLAG
5073 023036 012767 000001 001346  MOV    #1,LINMSK        ;INIT SELECT TEST MASK TO TEST LINE 00
5074 023044 105067 002074      CLRB  LINE              ;START WITH LINE #00
5075 023050 000405                BR     2S                ;GO TEST FOR LINE #00
5076 023052 105267 002066      IS:   INCB  LINE         ;GENERATE NEW LINE NO.
5077 023056 006367 001330      ASL   LINMSK           ;SHIFT SELECT MASK TO TEST NXT LINE
5078 023062 001407                BEQ   3S                ;RETURN TO EXIT BRANCH - ALL LINES DONE
5079 023064 036767 001322 001316 2S:   BIT   LINMSK,LINSEL  ;IS THE LINE SELECTED FOR TEST ??
5080 023072 001767                BEQ   IS                ;BR IF NOT
5081 023074 062716 000002      ADD   #2,(SP)          ;MOVE RETURN PC AROUND EXIT BRANCH
5082 023100 000402                BR     4S                ;RETURN TO TEST SELECTED LINE
5083 023102 005067 002036      3S:   CLR   LINE         ;INIT ENTRY FLAG AND LINE NO. TO 000
5084 023106 142777 000017 001264 4S:   BICB  #17,20HADR      ;INIT LINE SELECT BITS IN "SCR"
5085 023114 000207                RTS    PC              ;RETURN TO CALLING TEST

```

;THIS ROUTINE IS CALLED TO CONVERT EITHER THE "DH" NUMBER OR THE
;"LINE" NUMBER TO TWO ASCII CHARACTERS AND MOVE THEM INTO A
;PARTICULAR MESSAGE BUFFER FOR ERROR REPORTING

;CALLING SEQUENCE

```

;JSR   R5,SUNUM       ;CALL TO THIS ROUTINE
;ADDR1 ;ADDRESS OF THE NUMBER TO BE CONVERTED
;ADDR2 ;ADDRESS OF THE MSG BUFFER SLOT

```

```

5097 023116                SUNUM: MOV    R0,-(SP)      ;: PUSH R0 ON STACK
(2) 023116 010046                MOV    R1,-(SP)      ;: PUSH R1 ON STACK
(2) 023120 010146                MOV    R2,-(SP)      ;: PUSH R2 ON STACK
(2) 023122 010246                MOV    (R5)+,R0      ;: GET ADDRESS OF NUMBER
5098 023124 012500                MOV    (R5)+,R1      ;: GET MSG BUFFER ADDR
5099 023126 012501                MOV    (R0),R0       ;: GET NO. TO BE CONVERTED
5100 023130 111000                MOV    R0,R2         ;: SAVE IT IN R2
5101 023132 010002                ASR    R2             ;: SHIFT MSD TO LSD POSITION
5102 023134 006202

```

```

5103 023136 006202 ASR R2
5104 023140 006202 ASR R2
5105 023142 042702 177770 BIC #177770,R2 ;CLR JUNK BITS
5106 023146 062702 000060 ADD #60,R2 ;MAKE IT ASCII
5107 023152 110221 MOV# R2,(R1)+ ;PUT IT IN MSG BUFFER
5108 023154 042700 177770 BIC #177770,R0 ;CLR JUNK FROM LSD
5109 023160 062700 000060 ADD #60,R0 ;MAKE IT ASCII
5110 023164 110011 MOV# R0,(R1) ;PUT LSD IN THE BUFFER
5111 023166 012602 MOV (SP)+,R2 ;POP STACK INTO R2
5112 (2) 023170 012601 MOV (SP)+,R1 ;POP STACK INTO R1
5113 (2) 023172 012600 MOV (SP)+,R0 ;POP STACK INTO R0
5114 023174 000205 RTS R5 ;RETURN TO CALLER

;THIS ROUTINE IS CALLED TO CLEAR THE "CAR" AND "BCR" MEMORIES
;IT ASSUMES THAT THE ADDRESS OF THE "SCR" IS IN R1
5115
5116
5117 023176 005067 156014 CLCABC: CLR STMP7 ;INIT A COUNTER
5118 023202 116711 156010 IS: MOV# STMP7,(R1) ;SELECT A LINE
5119 023206 005061 000006 CLR CAR(R1) ;CLEAR A CAR LOCATION
5120 023212 005061 000010 CLR BCR(R1) ;CLEAR A BCR LOCATION
5121 023216 005267 155774 INC STMP7 ;GENERATE NEW LINE NO.
5122 023222 022767 000020 155766 CMP #20,STMP7 ;DONE ALL LINES?
5123 023230 001364 BNE IS ;BR IF NOT
5124 023232 142711 000017 BICB #17,(R1) ;SET "SCR" TO SELECT LINE 00
5125 023236 000207 RTS PC ;RETURN TO CALLER

;THIS ROUTINE IS CALLED TO LOAD THE "BCR" MEMORY WITH ALL ONES
;IT ASSUMES THAT THE ADDRESS OF THE SCR IS IN R1
5126
5127
5128
5129
5130 023240 005067 155752 LD8CR: CLR STMP7 ;INIT A COUNTER
5131 023244 116711 155746 IS: MOV# STMP7,(R1) ;SELECT A LINE
5132 023250 012761 177777 000010 MOV #-1,BCR(R1) ;LOAD BCR LOC. WITH 177777
5133 023256 005267 155734 INC STMP7 ;GENERATE NEXT LINE NO.
5134 023262 022767 000020 155726 CMP #20,STMP7 ;DONE ALL LINES?
5135 023270 001365 BNE IS ;BR IF NOT
5136 023272 142711 000017 BICB #17,(R1) ;SET "SCR" TO SELECT LINE 00
5137 023276 000207 RTS PC ;RETURN TO CALLER

;THIS ROUTINE CALLED TO SET UP FOR PARITY TESTS
5138
5139
5140
5141 023300 012767 000020 155702 SUPPAR: MOV #20,STMP4 ;SET UP FOR 16. LINES
5142 023306 105011 CLR# (R1) ;INIT SCR TO START AT LINE 00
5143 023310 005002 CLR R2 ;INIT INDEX REGISTER FOR RBUF (EVEN)
5144 023312 012703 000200 MOV #200,R3 ;SET UP CONSTANT
5145 023316 012704 000001 MOV #1,R4 ;INIT INDEX REG FOR RBUF (ODD)
5146 023322 012761 033436 000006 IS: MOV #RBUF,CAR(R1) ;LOAD BUS ADDRESS REWG
5147 023330 016761 155660 000010 MOV STMP6,BCR(R1) ;LOAD BYTE COUNT REG
5148 023336 016761 155650 000004 MOV STMP5,LPR(R1) ;LOAD LINE PARAMETERS
5149 023344 105062 032436 CLR# RBUF(R2) ;INIT DATA BYTE IN RBUF TO START AT 000
5150 023350 110364 032436 MOV# R3,RBUF(R4) ;SET CONSTANT IN HIGH BYTE
5151 023354 005211 INC (R1) ;SELECT NEXT LINE
5152 023356 005203 INC R3 ;GENERATE NEW CONSTANT
5153 023360 062702 000002 ADD #2,R2 ;UPDATE POINTERS TO RBUF (EVEN/ODD)
5154 023364 062704 000002 ADD #2,R4

```

```

5155 023370 005367 155614      DEC      STMP4      ;COUNT ONE LINE SETUP
5156 023374 001352              BNE      15          ;BR TILL ALL 16. SET UP
5157 023376 012704 024660      MOV      #MULPTB,R4 ;SET UP TABLE POINTER
5158 023402 016724 155606      2S:     MOV      STMP6,(R4)+ ;SET UP BYTE COUNT ENTRY
5159 023406 022704 024720      CMP      #MULPTB+40,R4 ;SET UP ALL COUNTS ?
5160 023412 001373              BNE      2S          ;BR IF NOT
5161 023414 105011              CLRB     (R1)        ;INIT SCR TO SELECT LINE 00
5162 023416 000207              RTS      PC          ;RETURN TO PARITY TEST
5163
5164 ;THIS ROUTINE IS USED TO ACCEPT INPUT PARAMETERS FROM THE CONSOLE
5165 ;TELETYPE
5166
5167 023420 104400      INPARA:  TYPE
5168 023422 032306              VCMC
5169 023424 104407              RDOCT
5170 023426 012600      MOV      (SP)+,RO
5171 023430 001412              BEQ      3S
5172 023432 022700 000004      CMP      #4,RO
5173 023436 001404              BEQ      2S
5174 023440 022700 000010      CMP      #10,RO
5175 023444 001404              BEQ      3S
5176 023446 000764              BR       INPARA
5177 023450 012700 000010      2S:     MOV      #10,RO
5178 023454 000402              BR       4S
5179 023456 012700 000020      3S:     MOV      #20,RO
5180 023462 000207      4S:     RTS      PC
5181
5182
5183
5184 023464 012700 177777      INPARC:  MOV      #-1,RO
5185 023470 000167 156452              JMP      BEGINA
5186
5187 023474 012767 177777 001374 INPARX:  MOV      #-1,VCFLG
5188 023502 000167 156440              JMP      BEGINA
5189
5190
5191 023506 104400      INPAR:   TYPE
5192 023510 031663              INMSG1
5193 023512 104407              RDOCT
5194 023514 012601      MOV      (SP)+,R1
5195 023516 001403              BEQ      INPAR1
5196 023520 004767 000106      JSR      PC,CHKADR
5197 023524 000770              BR       INPAR
5198
5199 023526 104400      INPAR1:  TYPE
5200 023530 031727              INMSG2
5201 023532 104407              RDOCT
5202 023534 012601      MOV      (SP)+,R1
5203 023536 001403              BEQ      INPAR3
5204 023540 004767 000150      JSR      PC,CHKVCT
5205 023544 000770              BR       INPAR1
5206
5207 023546 104400      INPAR3:  TYPE
5208 023550 031776              INMSG3
5209 023552 104407              RDOCT

```

```

;COUNT ONE LINE SETUP
;BR TILL ALL 16. SET UP
;SET UP TABLE POINTER
;SET UP BYTE COUNT ENTRY
;SET UP ALL COUNTS ?
;BR IF NOT
;INIT SCR TO SELECT LINE 00
;RETURN TO PARITY TEST

;THIS ROUTINE IS USED TO ACCEPT INPUT PARAMETERS FROM THE CONSOLE
;TELETYPE

INPARA: TYPE
VCMC
RDOCT
"ASK FOR NO. WORDS BETWEEN VECTORS"
READ OCTAL NO. FM TTY
GET THE NO. HE TYPED
BR IF HE TYPED <CR>
FOUR WORDS BETWEEN VECTORS ?
BR IF YES
8. WORDS BETWEEN VECTORS ??
BR IF YES
ASK ALL OVER AGAIN
SET UP CONSTANT IN RO FOUR 4 WORDS
CONTINUE
SET UP CONSTANT FOR 8. WORDS
RETURN TO CALLER

INPARC: MOV #-1,RO
;SET FLAG IN RO
;GO ASK FOR SELECT PARAMETER

INPARX: MOV #-1,VCFLG
;SET SETUP FLAG
;GO START UP

INPAR: TYPE
INMSG1
RDOCT
;ASK FOR DEVICE ADDRESS
;READ IN WHAT IS TYPED
;GET THE NO. HE TYPED
;BR IF DEFAULT
;GO CHECK VALIDITY OF THE ADDR
;ERROR BRANCH

INPAR1: TYPE
INMSG2
RDOCT
;ASK FOR VECTOR ADDRESS
;READ IN WHAT HE TYPES
;GET THE ADDRESS
;BR IF DEFAULT
;GO CHECK VALIDITY OF VECTOR
;ERROR BRANCH

INPAR3: TYPE
INMSG3
RDOCT
;ASK FOR DEVICE SELECTION PARAMETER
;READ IN WHAT HE TYPES

```

```

5209 023554 012601      MOV      (SP)+,R1      ;GET THE SELECT PARAMETER
5210 023556 001402      BEQ      INPAR4      ;BR IF DEFAULT
5211 023560 010167 000622      MOV      R1,DHSEL    ;SET UP DH11 SELECTION PARAMETER
5212
5213 023564 104400      INPAR4: TYPE          ;ASK FOR LINE SELECT PARAMETER
5214 023566 032174      INMSG6
5215 023570 104407      RDOCT
5216 023572 012601      MOV      (SP)+,R1    ;GET WHAT HE TYPES
5217 023574 001403      BEQ      1$          ;GET PARAMETER
5218 023576 010167 000606      MOV      R1,LINSEL  ;BR IF DEFAULT
5219 023602 000403      BR       2$          ;SET UP LINE SELECT PARAMETER
5220 023604 012767 177777 000576 1$:      MOV      #-1,LINSEL ;CONTINUE
5221 023612 005777 155320 2$:      TST     2$SWR      ;SET UP DEFAULT (ALL LINES)
5222 023616 100003      BPL     EXPAR        ;HALT AFTER SET UP ??
5223 023620 104400      TYPE          ;BR IF NOT
5224 023622 032236      INMSG7        ;TYPE CONTINUE MESSAGE PRIOR TO HALTING
5225 023624 000000      HALT
5226 023626 000167 156654      EXPAR: JMP      START2 ;DEPRESS CONTINUE TO RESUME TESTING
5227
5228
5229
5230
5231
5232
5233
5234
5235
5236
5237 023632 020127 160020      CHKADR: CMP      R1,#160020 ;IS ADDRESS ABOVE OR EQUAL TO LOW LIMIT
5238 023634 002001      BGE     1$          ;BR IF YES
5239 023640 000422      BR      4$          ;BR IF NOT
5240 023642 020127 160420 1$:      CMP      R1,#160420 ;IS IT BELOW THE HIGH LIMIT?
5241 023646 002401      BLT     2$          ;BR IF YES
5242 023650 000416      BR      4$          ;BR IF NOT
5243 023652 032701 000017 2$:      BIT      #17,R1      ;CORRECT BOUNDARY ?
5244 023656 001013      BNE     4$          ;BR IF NOT
5245 023660 062716 000002      ADD     #2,(SP)     ;MOVE RETURN PC AROUND ERROR BRANCH
5246 023664 012702 024776      MOV     #DHADTB,R2  ;POINT TO BEGIN OF ADDR TABLE
5247 023670 010122 000020 3$:      MOV     R1,(R2)+    ;SETR UP A TABLE ENTRY
5248 023672 062701 025036      ADD     #20,R1      ;GENERATE NEXT DH11 ADDR
5249 023676 022702      CMP     #DHADTB+40,R2 ;END OF TABLE ?
5250 023702 001372      BNE     3$          ;BR IF NOT
5251 023704 000402      BR      5$          ;RETURN TO INPUT ROUTINES
5252 023706 104400 4$:      TYPE          ;TELL HIM HE GOOFED
5253 023710 032047      INMSG4
5254 023712 000207 5$:      RTS     PC          ;RETURN TO INPUT ROUTINES
5255
5256
5257
5258
5259
5260
5261
5262 023714 020127 000300      CHKVCT: CMP      R1,#300 ;IS ADDRESS ABOVE OR EQUAL TO LOW LIMIT
5263 023720 002001      BGE     1$          ;BR IF YES
5264 023722 000421      BR      4$          ;BR IF NOT
5265 023724 020127 001000 1$:      CMP      R1,#1000  ;IS IT BELOW THE HIGH LIMIT?
5266 023730 002401      BLT     2$          ;BR IF YES
5267 023732 000415      BR      4$          ;BR IF NOT
5268 023734 032701 000007 2$:      BIT      #7,R1      ;CORRECT BOUNDARY ?
5269 023740 001012      BNE     4$          ;BR IF NOT
5270 023742 062716 000002      ADD     #2,(SP)     ;MOVE RETURN PC AROUND ERROR BRANCH
5271 023746 012702 025036      MOV     #DHVCTB,R2  ;POINT TO BEGIN OF VECTOR TABLE
5272 023752 010122 000001 3$:      MOV     R1,(R2)+    ;SETR UP A TABLE ENTRY
5273 023754 060001      ADD     R0,R1      ;GENERATE NEXT DH11 ADDR
5274 023756 022702 025076      CMP     #DHVCTB+40,R2 ;END OF TABLE ?
5275 023762 001373      BNE     3$          ;BR IF NOT
5276 023764 000402      BR      5$          ;RETURN TO INPUT ROUTINES

```

```

5263 023766 104400          4S:   TYPE           ;TELL HIM HE GOOFED
5264 023770 032120          INMSG5
5265 023772 000207          5S:   RTS           PC      ;RETURN TO INPUT ROUTINES
5266
5267          ;THESE TWO ROUTINES SERVICE UNEXPECTED BUS ERROR AND RSVD INSTR TRAPS
5268
5269 023774 012767 000340 155176 BUSER:  MOV      #340,STMP0      ;SAVE THE PSW
5270 024002 010667 155166          MOV      SP,SREG6      ;SAVE THE SP
5271 024006 012601          MOV      (SP)+,R1      ;GET THE TRAP PC
5272 024010 012602          MOV      (SP)+,R2      ;GET THE TRAP PSW
5273 024012 116700 155064          MOV      STSTN,R0      ;GET TEST NO.
5274 024016 012706 001100          MOV      #STACK,SP     ;RESET THE STACK POINTER
5275 024022 004767 176714          JSR      PC,SUER3      ;GO SET UP ERROR INFO
5276 024026 012767 024036 :55054 MOV      #1$,SLPERR     ;ALWAYS COME BACK TO 1$
5277 024034 104027          ERROR   27            ;UNEXPECTED BUS ERROR TRAP
5278 024036 000005          1S:   RESET          ;PREPARE TO RESTART
5279 024040 004767 000240          JSR      PC,CHPS1      ;GO CLEAR PSW
5280 024044 000167 156532          JMP      REST1         ;GO RESTART THE PROGRAM
5281
5282 024050 012767 000340 155122 RESERR: MOV      #340,STMP0      ;SAVE THE PSW
5283 024056 010667 155112          MOV      SP,SREG6      ;SAVE THE SP
5284 024062 012601          MOV      (SP)+,R1      ;GET THE TRAP PC
5285 024064 012602          MOV      (SP)+,R2      ;GET THE TRAP PSW
5286 024066 116700 155010          MOV      STSTN,R0      ;GET TEST NO.
5287 024072 012706 001100          MOV      #STACK,SP     ;RESET THE STACK POINTER
5288 024076 004767 176640          JSR      PC,SUER3      ;GO SET UP ERROR INFO
5289 024102 012767 024112 155000 MOV      #1$,SLPERR     ;ALWAYS COME BACK TO 1$
5290 024110 104030          ERROR   30            ;UNEXPECTED RSVD INSTR ERROR TRAP
5291 024112 000005          1S:   RESET          ;PREPARE TO RESTART
5292 024114 004767 000164          JSR      PC,CHPS1      ;GO CLEAR PSW
5293 024120 000167 156456          JMP      REST1         ;GO RESTART THE PROGRAM
5294
5295          ;THIS ROUTINE IS CALLED WHEN A TEST NEEDS TO RESTORE THE TRAP
5296          ;CATCHER IN THE DH11 VECTOR
5297
5298 024124 016703 000252          RESTRP: MOV      DHVCT,R3      ;GET VECTOR ADDRESS
5299 024130 010313          MOV      R3,(R3)      ;RESTORE THE TRAP CATCHER
5300 024132 062723 000002          ADD      #2,(R3)+
5301 024136 005023          CLR      (R3)+
5302 024140 010313          MOV      R3,(R3)
5303 024142 062723 000002          ADD      #2,(R3)+
5304 024146 005023          CLR      (R3)+
5305 024150 000207          RTS           PC      ;RETURN TO CALLING TEST
5306
5307          ;THIS ROUTINE CALLED BY ANY TEST THAT NEEDS A TIMING WAIT LOOP
5308          ;"TIMEA" IS INITIALIZED BY THE CALLING ROUTINE TO THE MINIMUM REQUIRED
5309          ;VALUE AND "TIMEB" IS CLEARED TO 000000. IF A TIME OUT OCCURS THIS
5310          ;ROUTINE WILL MOVE THE RETURN PC AROUND THE "LOOP" BRANCH BACK IN
5311          ;THE ROUTINE THAT CALLED IT TO ALLOW REPORTING AN ERROR MESSAGE
5312
5313 024152 005267 001016          TIMEIT: INC      TIMEB      ;COUNT B
5314 024156 001005          BNE     1$            ;BR IF NOT ZERO
5315 024160 005367 001006          DEC      TIMEA      ;COUNT TIME A
5316 024164 001002          BNE     1$            ;BR IF NO TIMEOUT
    
```

```

5317 024166 062716 000002          ADD    #2,(SP)      ;MOVE RETURN PC TO ALLOW ERROR REPORT
5318 024172 000207          IS:    RTS         PC      ;RETURN TO THE CALLING TEST
5319
5320
5321
5322
5323
5324
5325
5326
5327
5328
5329
5330
5331
5332
5333
5334
5335
5336
5337
5338
5339
5340
5341
5342
5343
5344
5345
5346
5347
5348
5349
5350
5351
5352
5353
5354
5355
5356
5357
5358
5359
5360
5361
5362
5363
5364
5365
5366
5367
5368
5369
5370
    024174 012767 000020 155012 SETALL: MOV    #20,STMP6      ;SET UP SIXTEEN LINES
    024202 005002          CLR    R2           ;INIT A TABLE INDEX REG
    024204 012703 000200          MOV    #200,R3      ;SET UP TO GENERATE HI BYTE OF EXPECTED DATA
    024210 012704 000001          MOV    #1,R4        ;SET UP INDEX REG TO ODD BYTES
    024214 005011          CLR    (R1)         ;START WITH LINE 00
    024216 012761 033436 000006 IS:    MOV    #TBUF,CAR(R1) ;SET UP BUS ADDR REG
    024224 012761 177400 000010          MOV    #-400,BCR(R1);SET UP BYTE COUNT REG
    024232 012761 031403 000004          MOV    #31403,LPR(R1);SET UP FOR 4800 BAUD/8 BIT CHARS
    024240 105062 032436          CLRB  RBUF(R2)      ;START WITH DATA CHAR OF 000
    024244 110364 032436          MOVB  R3,RBUF(R4)   ;SET UP HIGH BYTE OF EXPECTED DATA
    024250 005211          INC    (R1)         ;GEN NEW LINE NO. IN SCR
    024252 005203          INC    R3           ;UPDATE THE POINTERS AND DATA
    024254 062702 000002          ADD    #2,R2
    024260 062704 000002          ADD    #2,R4
    024264 005367 154724          DEC    STMP6        ;COUNT ONE LINE DONE
    024270 001352          BNE    IS           ;BR TIL ALL 16 SET UP
    024272 016767 000112 000356          MOV    LINSEL,LINACT;SET SOFTWARE FLAG FOR ALL LINES ACTIVE
    024300 005011          CLR    (R1)         ;PUT SCR REG BACK TO LINE 00
    024302 000207          RTS         PC      ;RETURN TO AUTO ECHO TEST
    
```

;THIS ROUTINE IS CALLED TO SET PSM PRIORITY TO 000 IN ORDER
 ;TO BE LS111 COMPATIBLE

```

024304 012746 000000          CHPS1: MOV    #0,-(SP) ;NEW PSM
024310 012746 024316          MOV    #1$,-(SP)    ;NEW PC
024314 000002          RTI
024316 000207          IS:    RTS         PC      ;CHANGE PSM
                                ;RETURN TO CALLING TEST
    
```

;THIS ROUTINE DOES THE SAME THING EXCEPT IT SET THE PSM
 ;PRIORITY TO 340 (LEVEL 7) TO LOCK OUT INTR

```

024320 012746 000340          CHPS2: MOV    #340,-(SP);NEW PSM
024324 012746 024332          MOV    #1$,-(SP)   ;NEW PC
024330 000002          RTI
024332 000207          IS:    RTS         PC      ;CHANGE THE PSM
                                ;RETURN TO CALLING TEST
    
```

;THIS ROUTINE IS ALSO FOR LS111 COMPATIBILITY AND IT IS CALLED
 ;TO SAVE THE PSM IN "STMP0"

```

024334 005046          SAPS:  CLR    -(SP)   ;TEMP STORAGE TO SAVE PSM
024336 016746 153472          MOV    #34,-(SP)   ;SAVE TRAP VECTOR POINTER
024342 012767 024352 153464          MOV    #1$,34      ;GO TO IS ON TRAP
024350 104400          TRAP
024352 016666 000002 000006 IS:    MOV    2(SP),6(SP) ;GET PSM SAVED
024350 012716 024366          MOV    #2$, (SP)  ;GO TO 2$ ON RTI
024364 000002          RTI
5370 024366 012667 153442          2$:    MOV    (SP)+,34 ;RESTORE VECTOR
    
```

C05

MAINDEC-11-DZDMM-A MACY11 27(663) 12-DEC-75 08:41 PAGE 93-7
DZDMMR.P11 POWER DOWN AND UP ROUTINES

SEQ 0260

5371	024372	012667	154602	MOV	(SP)+,STMPD	;FINALLY SAVE PSW IN STMPD
5372	024376	000207		RTS	PC	
5373						

5375
5376
5377
5378
5379
5380
5381
5382
5383
5384
5385
5386
5387
5388
5389
5390
5391
5392
5393
5394
5395
5396
5397
5398
5399
5400
5401
5402
5403
5404
5405
5406
5407
5408
5409
5410
5411
5412
5413
5414
5415
5416
5417
5418
5419
5420
5421
5422
5423
5424
5425
5426
5427
5428

000002
000004
000006
000010
000012
000014
000016

024400 000000
024402 000000
024404 000000
024406 000003
024410 177777
024412 000000
024414 000000

024416 000004

024426 177777
024430 125252
024432 052525
024434 000000

024436 000060
024440 000300
024442 000020
024444 000100
024446 000040
024450 000200
024452 000000
024454 000000

024456 177400
024460 027363
024462 177400
024464 027323
024466 177600
024470 027362
024472 177600
024474 027322
024476 177700
024500 027361
024502 177700
024504 027321
024506 177740
024510 027360

: ADDITIONAL PROGRAM CONSTANTS AND VARIABLES

NRC=2 : INDEX CONST. TO ACCESS NEXT RCVD CHAR REG
LPR=4 : INDEX CONST. TO ACCESS LINE PARAMETER REG.
CAR=6 : INDEX CONST. TO ACCESS CURRENT ADDRESS REG.
BCR=10 : INDEX CONST. TO ACCESS BYTE COUNT REG.
BAR=12 : INDEX CONST. TO ACCESS BUFFER ACTIVE REG.
BKR=14 : INDEX CONST. TO ACCESS BREAK CONTROL REG.
SSR=16 : INDEX CONST. TO ACCESS SILO STATUS REG.

DHADR: 0 : HOLDS THE "SCR" ADDRESS OF THE DH11 UNDER TEST
DHVCT: 0 : HOLDS THE 1ST VECTOR ADDRESS OF THE DH11 UNDER TEST
SELMSK: 0 : BIT TST MARKER FOR SELECTING DH11'S
DHSEL: 3 : SPECIFIES DH11'S SELECTED FOR TEST
LINSEL: 177777 : SPECIFIES LINES TO TEST
LINMSK: 0 : MARKER USED TO TEST FOR LINES TO TEST
LMSK1: 0 : ALTERNATE MARKER TO SUPPORT THE
: SELECT LINES FEATURE
MSTCLR: .BLKW 4 : FOUR WORD ADDRESS TABLE USED BY THE TEST THAT
: CHECKS OPERATION OF "MASTER CLR"

PATRMA: 177777 : BIT PATTERNS USED WITH "CAR" AND "BCR" TESTS
125252
052525
000000 : TABLE TERMINATOR

PATRNB: 60 : BIT PATTERNS USED IN "CAR" MEM EXT BIT TEST
300
20
100
40
200
0 : TABLE TERMINATOR
0 : TABLE TERMINATOR

: THIS TABLE STORES THE BYTE COUNT AND LINE PARAMETERS FOR THE
: 8 SUBTESTS IN THE MULTILINE PARITY/DATA TEST
PRTYTB: -400 : 256 CHARS
27363 : 2400 BAUD - ODD PARITY - 8 BITS
-400 : 256 CHARS
27323 : 2400 BAUD - EVEN PARITY - 8 BITS
-200 : 128 CHARS
27362 : 2400 BAUD - ODD PARITY - 7 BITS
-200 : 128 CHARS
27322 : 2400 BAUD - EVEN PARITY - 7 BITS
-100 : 64 CHARS
27361 : 2400 BAUD - ODD PARITY - 6 BITS
-100 : 64 CHARS
27321 : 2400 BAUD - EVEN PARITY - 6 BITS
-40 : 32 CHARS
27360 : 2400 BAUD - ODD PARITY - 5 BITS


```

5429 024512 177740
5430 024514 027320
5431
5432
5433
5434
5435
5436
5437
5438
5439
5440
5441
5442
5443
5444
5445
5446
5447
5448
5449
5450
5451
5452
5453
5454
5455
5456
5457
5458
5459
5460
5461
5462
5463
5464
5465
5466
5467
5468
5469
5470
5471
5472
5473
5474
5475
5476
5477
5478
5479
5480
5481
5482

```

```

-40 ;32 CHARS
27320 ;2400 BAUD - EVEN PARITY - 5 BITS

;THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE AUTO ECHO
;TEST (ALL 1'S DATA TABLE)

AETAB: 100377 ;TEST DATA FOR LINE 00
        100777 ;TEST DATA FOR LINE 01
        101377
        101777
        102377
        102777
        103377
        103777
        104377
        104777
        105377
        105777
        106377
        106777
        107377
        107777 ;TEST DATA FOR LINE 17

```

```

;THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE AUTO ECHO
;TEST (ALL 0'S DATA TABLE)

AETAB0: 100000 ;TEST DATA FOR LINE 00
        100400 ;TEST DATA FOR LINE 01
        101000
        101400
        102000
        102400
        103000
        103400
        104000
        104400
        105000
        105400
        106000
        106400
        107000
        107400 ;TEST DATA FOR LINE 17

```

```

;THIS TABLE USED BY THE AUTO ECHO TEST 2 TO RESET ACTIVE BIT WHEN A
;LINE IS DONE

LINBIT: BIT00 ;DEACTIVATE LINE 00
        BIT01 ;DEACTIVATE LINE 01
        BIT02
        BIT03
        BIT04
        BIT05
        BIT06
        BIT07

```

5483 024636 000400
5484 024640 001000
5485 024642 002000
5486 024644 004000
5487 024646 010000
5488 024650 020000
5489 024652 040000
5490 024654 100000
5491
5492 024656 000000
5493
5494
5495
5496
5497
5498
5499 024660 000020
5500
5501
5502
5503
5504 024720 120000
5505 024722 120400
5506 024724 121000
5507 024726 121400
5508 024728 121400
5509 024730 122000
5510 024732 122400
5511 024734 123000
5512 024736 123400
5513 024738 124000
5514 024740 124400
5515 024742 124400
5516 024744 125000
5517 024746 125400
5518 024750 126000
5519 024752 126400
5520 024754 127000
5521 024756 127400
5522
5523 024760 131177
5524 024762 046600
5525 024764 177767
5526 024766 177777
5527 024770 100077
5528 024772 042200
5529 024774 030100
5530
5531
5532 024776 160020
5533 025000 160040
5534 025002 160060
5535 025004 160100
5536 025006 160120

BIT08
BIT09
BIT10
BIT11
BIT12
BIT13
BIT14
BIT15

;DEACTIVATE LINE 17

LINACT: 0

;MAINTAINS STATUS OF ACTIVE LINES
;DURING AUTO ECHO TEST 2

;THIS TABLE CONTAINS 16. COUNTERS USED BYN THE MULTI-LINE
;PARITY TEST TO KEEP TRACK OF TOTAL CHARS RECEIVED

MULPTB: .BLKN 16. ;SIXTEEN WORD COUNTERS TABLE

;THIS 16 WORD TABLE CONTAINS THE TEST DATA USED BY THE BREAK BIT
;TEST

BRKTAB: 120000 ;TEST DATA FOR LINE 00
120400 ;TEST DATA FOR LINE 01
121000
121400
121400
122000
122400
123000
123400
124000
124400
125000
125400
126000
126400
127000
127400 ;TEST DATA FOR LINE 17

RGMSK1: 131177 ;MASK TO SPECIFY R/W BITS FOR NORMAL "SCR" REG TEST
RGMSK2: 46600 ;MASK TO SPECIFY READ ONLY BITS IN "SCR" FOR NORMAL MODE TEST
RGMSK3: 177767 ;MASK TO SPECIFY R/W BITS IN "LPR"
RGMSK4: 177777 ;MASK TO SPECIFY R/W BITS IN "BKR"
RGMSK5: 100077 ;MASK TO SPECIFY R/W BITS IN "SSR"
RGMSK6: 42200 ;MASK TO SPECIFY READ ONLY BITS IN "SCR" FOR MAINT. MODE TEST
INTMSK: 30100 ;MASK USED TO SELECT INTR BITS TO TEST

;DH11 ADDRESS TABLE - THIS TABLE CONTAINS THE "SCR" ADDRESS FOR UP TO
;SIXTEEN DH11'S

DHADTB: 160020 ;ADDRESS OF FIRST DH11
160040 ;ADDRESS OF SECOND DH11
160060
160100
160120

55743
55744
55745
55746
55747
55748
55749
55750
55751
55752
55753
55754
55755
55756
55757
55758
55759
55760
55761
55762
55763
55764
55765
55766
55767
55768
55769
55770
55771
55772
55773
55774
55775
55776
55777
55778
55779
55780
55781
55782
55783
55784
55785
55786
55787
55788
55789
55790
55791
55792
55793
55794
55795
55796
55797
55798
55799
55800

025010
025012
025014
025016
025018
025020
025022
025024
025026
025028
025030
025032
025034
025036
025038
025040
025042
025044
025046
025048
025050
025052
025054
025056
025058
025060
025062
025064
025066
025068
025070
025072
025074
025076

160140
160160
160200
160220
160240
160260
160300
160320
160340
160360
160400

160140
160160
160200
160220
160240
160260
160300
160320
160340
160360
160400

; ADDRESS OF THE LAST DH11

; DH11 VECTOR TABLE - THIS TABLE CONTAINS THE VECTOR ADDRESSES FOR UP
; TO SIXTEEN DH11'S

DHVCTB: 330
350
370
410
430
450
470
510
530
550
570
610
630
650
670
710

; ADDRESS OF VECTOR FOR FIRST DH11
; ADDRESS OF VECTOR FOR SECOND DH11

; ADDRESS OF VECTOR FOR LAST DH11

VCFLG: 0

; VECTOR SET UP FLAGG

; BR PRIORITY LEVEL TABLE - THIS TABLE CONTAINS THE PRIORITY LEVELS
; FOR UP TO SIXTEEN DH11'S - THE RCVR LEVEL IS STORED IN THE LOW BYTE
; AND THE XMTTR LEVEL IN THE HIGH BYTE

BRVLV: 120240
120240
120240
120240
120240
120240
120240
120240
120240
120240
120240
120240
120240
120240
120240
120240
120240
120240
120240
120240

; BRLEVELS FOR FIRST DH11
; BR LEVELS FOR SECOND DH11

5591 025136 120240
5592 025140 000
5593 025141 000
5594 025142 000000
5595 025144 000000
5596 025146 000000
5597
5598
5599
5600 025150 000000
5601 025152 000000
5602 025154 000000
5603
5604 025156 000000
5605 025160 000037
5606 025162 000077
5607 025164 000177
5608 025166 000377
5609 025170 000000
5610 025172 000000
5611 025176 000000
5612 025200 000000
5613
5614
5615

120240 ;BR LEVELS FOR LAST DH11
DHRLVL: .BYTE 0 ;BR LEVEL FOR RCVR
DH1LVL: .BYTE 0 ;BR LEVEL FOR XMITTER
DHNUM: 0 ;CONTAINS NUMBER OF THE DH11 UNDER TEST
LINE: 0 ;CONTAINS NUMBER OF THE LINE UNDER TEST
LINEA: 0 ;LOCATION TO SAVE LINE NUMBER
;ADDRESS POINTERS TO SET UP TABLES WHEN INPUTTING PARAMETERS
ADPTR: 0 ;POINTS TO ADDRESS TABLE
VCPTR: 0 ;POINTS TO VECTOR TABLE
BRPTR: 0 ;POINTS TO BR LEVEL TABLE
TDATA1: 0 ;DATA BUFFER FOR BASIC DATA TEST
TDATA2: 37 ;TEST DATA FOR FIVE BIT CHAR
77 ;TEST DATA FOR SIX BIT CHAR
177 ;TEST DATA FOR SEVEN BIT CHAR
377 ;TEST DATA FOR EIGHT BIT CHAR
TITFLG: 0 ;FLAG TO ALLOW PRINTING TITLE ONLY ONCE
TIMEA: 0 ;GENERAL PURPOSE TIMERS
TIMEB: 0
TIMEC: 0
TNULL: 0 ;TIMER FOR TIMING TESTS
;CONTAINS TWO NULL CHARS USED BY BREAK TEST

5617
5618
5619
5620
5621
5622
5623
5624
5625
5626
5627
5628
5629
5630
5631
5632
5633
5634
5635
5636
5637

; ERROR MESSAGE INFORMATION - MESSAGE BUFFERS AND POINTERS

; INFORMATION FOR MESSAGE 1

EM1: .ASCIZ 'DH11 REGISTER REFERENCE CAUSED TIMEOUT'

DH1: .ASCIZ ' (PC) (PS) (SP) TEST DEVAOR REGADR'

.EVEN
DT1: .WORD SERRPC, STMPD, SREG6, SREG0, SREG1, SREG2, 0

; INFORMATION FOR MESSAGE 2

EM2: .ASCIZ 'SYSTEM CONTROL REGISTER ERROR'

DH2: .ASCIZ ' (PC) (PS) (SP) TEST DEVAOR REGADR HAS S/B'

.EVEN
DT2: .WORD SERRPC, STMPD, SREG6, SREG0, SREG1, SREG2, SREG3, SREG4, 0

; INFORMATION FOR MESSAGE 3

EM3: .ASCIZ 'DH11 MASTER CLEAR FAILED TO CLR SPECIFIED REG'

025303 044104 030461 051040
042522 043506 051511 042524
020122 042522 042506
042522 041516 020105
040503 051523 042105
052040 046511 7505
02123 000
040 050050 024503
020040 020040 050050
024523 020040 020040
051450 024520 020040
020040 042524 052123
020040 042040 053105
042101 020122 051040
043505 042101 000122
025330 001116 001200 001174
025336 001160 001162 001164
025344 000000
025346 054523 052123 046505
041440 047117 051124
046117 051040 043505
051511 042524 020122
051105 047522 000122
024040 041520 020051
020040 024040 051520
020051 020040 024040
050123 020051 020040
052040 051505 020124
020040 042504 040526
051104 020040 042522
040507 051104 020040
053440 051501 020040
020040 051440 041057
000
025502
001116 001200 001174
001160 001162 001164
001166 001170 000000
044104 030461 046440
051501 042524 020122
046103 040505 020122

	025546	040506	046111	042105	
	025554	052040	020117	046103	
	025562	020122	050123	041505	
	025570	043111	042511	020104	
	025576	042522	000107		
5638					
5639					
5640					; INFORMATION FOR MESSAGE 4
5641	025602	044514	042516	050040	EM4: .ASCIZ 'LINE PARAMETER REGISTER ERROR'
	025610	051101	046501	052105	
	025616	051105	051040	043505	
	025624	051511	042524	020122	
	025632	051105	047522	000122	
5642					
5643					; INFORMATION FOR MESSAGE 5
5644					
5645	025640	051102	040505	020113	EM5: .ASCIZ 'BREAK CONTROL REGISTER ERROR'
	025646	047503	052116	047522	
	025654	020114	042522	044507	
	025662	052123	051105	042440	
	025670	051122	051117	000	
5646					
5647					; INFORMATION FOR MESSAGE 6
5648					
5649	025675	123	046111	020117	EM6: .ASCIZ 'SILO STATUS REGISTER ERROR'
	025702	052123	052101	051525	
	025710	051040	043505	051511	
	025716	042524	020122	051105	
	025724	047522	000122		
5650					
5651					; INFORMATION FOR MESSAGE 7
5652					
5653	025730	052503	051122	047105	EM7: .ASCIZ 'CURRENT ADDRESS REGISTER ERROR - LINE #XX'
	025736	020124	042101	051104	
	025744	051505	020123	042522	
	025752	044507	052123	051105	
	025760	042440	051122	051117	
	025766	026440	046040	047111	
	025774	020105	054043	000130	
5654					
5655					; INFORMATION FOR MESSAGE 10
5656					
5657	026002	054502	042524	041440	EM10: .ASCIZ 'BYTE COUNTER REGISTER ERROR - LINE #XX'
	026010	052517	052116	051105	
	026016	051040	043505	051511	
	026024	042524	020122	051105	
	026032	047522	020122	020055	
	026040	044514	042516	021440	
	026046	054130	000		
5658					
5659					; INFORMATION FOR MESSAGE 11
5660					
5661	026051	125	042516	050130	EM11: .ASCIZ 'UNEXPECTED DH11 RCVR INTERRUPT'
	026056	041505	042524	020104	

026064 044104 030461 051040
026072 053103 020122 047111
026100 042524 051122 050125
026106 000124

5662
5663
5664
5665

; INFORMATION FOR MESSAGE 12

026110 047125 054105 042520
026116 052103 042105 042040
026124 030510 020061 046530
026132 052111 051124 044440
026140 052116 051105 052522
026146 052120 000

EM12: .ASCIZ 'UNEXPECTED DH11 XMITTR INTERRUPT'

5666
5667
5668
5669

; INFORMATION FOR MESSAGE 13

026151 103 040510 020122
026156 053101 044501 040514
026164 046102 020105 040506
026172 046111 042105 052040
026200 020117 042507 042516
026206 040522 042524 051040
026214 053103 020122 047111
026222 042524 051122 050125
026230 000124

EM13: .ASCIZ 'CHAR AVAILABLE FAILED TO GENERATE RCVR INTERRUPT'

5670
5671
5672
5673

; INFORMATION FOR MESSAGE 14

026232 051124 047101 046523
026240 052111 042524 020122
026246 050116 020122 047514
026254 044507 020103 051105
026262 047522 020122 020055
026270 044514 042516 021440
026276 020040 000

EM14: .ASCIZ 'TRANSMITTER NPR LOGIC ERROR - LINE # '

5674
5675
5676
5677

; INFORMATION FOR MESSAGE 15

026301 130 044515 052124
026306 020122 040506 046111
026314 042105 052040 020117
026322 047111 042524 051122
026330 050125 020124 020055
026336 044514 042516 021440
026344 020040 000

EM15: .ASCIZ 'XMITTR FAILED TO INTERRUPT - LINE # '

5678
5679
5680
5681

; INFORMATION FOR MESSAGE 16

026347 122 053103 020122
026354 040506 046111 042105
026362 052040 020117 047111
026370 042524 051122 050125
026376 000124

EM16: .ASCIZ 'RCVR FAILED TO INTERRUPT'

5682

```

5683 ;INFORMATION FOR MESSAGE 17
5684
5685 026400 051124 047101 046523 EM17: .ASCIZ 'TRANSMITTER TIMING ERROR - LINE # '
      026406 052111 042524 020122
      026414 044524 044515 043516
      026422 042440 051122 051117
      026430 026440 046040 047111
      026436 020105 020043 000040
5686 026444 024040 041520 020051 DM6: .ASCIZ ' (PC) (PS) (SP) TEST DEVADR SPEED TIMEB TIMEC'
      026452 020040 024040 051520
      026460 020051 020040 024040
      026466 050123 020051 020040
      026474 052040 051505 020124
      026502 020040 042504 040526
      026510 051104 020040 050123
      026516 042505 020104 020040
      026524 044524 042515 020102
      026532 020040 044524 042515
      026540 000103

```

```

5687 ;INFORMATION FOR MESSAGE 20
5688
5689
5690 026542 042522 042503 053111 EM20: .ASCIZ 'RECEIVER TIMING ERROR - LINE # '
      026550 051105 052040 046511
      026556 047111 020107 051105
      026564 047522 020122 020055
      026572 044514 042516 021440
      026600 020040 000

```

```

5691 ;INFORMATION FOR MESSAGE 21
5692
5693
5694 026603 122 053103 020122 EM21: .ASCIZ 'RCVR FAILED TO INTERRUPT - LINE # '
      026610 040506 046111 042105
      026616 052040 020117 047111
      026624 042524 051122 050125
      026632 020124 020055 044514
      026640 042516 021440 020040
      026646 000

```

```

5695 ;INFORMATION FOR MESSAGE 22
5696
5697
5698 026647 103 040510 020122 EM22: .ASCIZ 'CHAR AVAIL FAILED TO SET ON TIME - LINE # '
      026654 053101 044501 020114
      026662 040506 046111 042105
      026670 052040 020117 042523
      026676 020124 047117 052040
      026704 046511 020105 020055
      026712 044514 042516 021440
      026720 020040 000

```

```

5699 ;INFORMATION FOR MESSAGE 23
5700
5701
5702 026723 102 051501 041511 EM23: .ASCIZ 'BASIC DATA TEST ERROR - LINE # '
      026730 042040 052101 020101

```


	027304	020040	042524	052123	
	027312	020040	052040	050122	
	027320	041520	020040	052040	
	027326	050122	051520	000040	
5721					.EVEN
5722	027334	001116	001200	001174	DT3: .WORD SERRPC, STMP0, SREG6, SREG0, SREG1, SREG2, 0
	027342	001160	001162	001164	
	027350	000000			
5723					
5724					; INFORMATION FOR MESSAGE 30
5725					
5726	027352	047125	054105	042520	EM30: .ASCIZ 'UNEXPECTED RSVD INSTR TRAP'
	027360	052103	042105	051040	
	027366	053123	020104	047111	
	027374	052123	020122	051124	
	027402	050101	000		
5727					
5728					; INFORMATION FOR MESSAGE 31
5729					
5730	027405	101	052125	020117	EM31: .ASCIZ 'AUTO ECHO DATA COMPARE ERROR - LINE # '
	027412	041505	047510	042040	
	027420	052101	020101	047503	
	027426	050115	051101	020105	
	027434	051105	047522	020122	
	027442	020055	044514	042516	
	027450	021440	020040	000	
5731	027455	040	050050	024503	DH4: .ASCIZ ' (PC) (PS) (SP) TEST WASADR SBADR WAS S/B'
	027462	020040	020040	050050	
	027470	024523	020040	020040	
	027476	051450	024520	020040	
	027504	020040	042524	052123	
	027512	020040	053440	051501	
	027520	042101	020122	051440	
	027526	040502	051104	020040	
	027534	020040	040527	020123	
	027542	020040	020040	027523	
	027550	000102			
5732					
5733					; INFORMATION FOR MESSAGE 32
5734					
5735	027552	052501	047524	042440	EM32: .ASCIZ 'AUTO ECHO TEST TIMEOUT - LINE # '
	027560	044103	020117	042524	
	027566	052123	052040	046511	
	027574	047505	052125	026440	
	027602	046040	047111	020105	
	027610	020043	000040		
5736	027614	024040	041520	020051	DH5: .ASCIZ ' (PC) (LPRG) TEST'
	027622	020040	046050	051120	
	027630	024507	020040	052040	
	027636	051505	000124		
5737					.EVEN
5738	027642	001116	001200	001204	DT4: .WORD SERRPC, STMP0, STMP2, 0
	027650	000000			
5739					

```

5740 ;INFORMATION FOR MESSAGE 33
5741
5742 027652 040520 044522 054524 EM33: .ASCIZ 'PARITY LOGIC TEST ERROR - LINE # '
      027660 046040 043517 041511
      027666 052040 051505 020124
      027674 051105 047522 020122
      027702 020055 044514 042516
      027710 021440 020040 000

```

```

5743 ;INFORMATION FOR MESSAGE 34
5744
5745
5746 027715 115 046125 044524 EM34: .ASCIZ 'MULTI-LINE PARITY DATA TEST ERROR - LINE # - SUBTEST # '
      027722 046055 047111 020105
      027730 040520 044522 054524
      027736 042040 052101 020101
      027744 042524 052123 042440
      027752 051122 051117 026440
      027760 046040 047111 020105
      027766 020043 020040 020055
      027774 052523 052102 051505
      030002 020124 020043 000040

```

```

5747 ;INFORMATION FOR MESSAGE 35
5748
5749
5750 030010 052515 052114 026511 EM35: .ASCIZ 'MULTI-LINE PARITY DATA TEST TIMEOUT'
      030016 044514 042516 050040
      030024 051101 052111 020131
      030032 040504 040524 052040
      030040 051505 020124 044524
      030046 042515 052517 000124
5751 030054 024040 041520 020051 DM14: .ASCIZ ' (PC) (LPRG) ACTLIN'
      030062 020040 046050 051120
      030070 024507 020040 041501
      030076 046124 047111 000

```

```

5752 .EVEN
5753 030104 001116 001200 001206 DT6: .WORD SERRPC,STMP0,STMP3,0
      030112 000000

```

```

5754 ;INFORMATION FOR MESSAGE 36
5755
5756
5757 030114 044103 051101 040440 EM36: .ASCIZ 'CHAR AVAILABLE TIMEOUT'
      030122 040526 046111 041101
      030130 042514 052040 046511
      030136 047505 052125 000

```

```

5758 ;INFORMATION FOR MESSAGE 37
5759
5760
5761 030143 104 052101 020101 EM37: .ASCIZ 'DATA COMPARE ERROR - LINE # '
      030150 047503 050115 051101
      030156 020105 051105 047522
      030164 020122 020055 044514
      030172 042516 021440 020040
      030200 000

```

5762

```

5763 ;INFORMATION FOR MESSAGE 40
5764
5765 030201 102 043125 042506 EM40: .ASCIZ 'BUFFER ACTIVE REG ERROR - LINE # '
      030206 020122 041501 044524
      030214 042526 051040 043505
      030222 042440 051122 051117
      030230 026440 046040 047111
      030236 020105 020043 000040

5766 ;INFORMATION FOR MESSAGE 41
5767
5768
5769 030244 041522 051126 043040 EM41: .ASCIZ 'RCVR FALSE INTERRUPT'
      030252 046101 042523 044440
      030260 052116 051105 052522
      030266 052120 000

5770 ;INFORMATION FOR MESSAGE 42
5771
5772
5773 030271 123 046111 020117 EM42: .ASCIZ 'SILO OVERFLOW ERROR'
      030276 053117 051105 046106
      030304 053517 042440 051122
      030312 051117 000

5774 ;INFORMATION FOR MESSAGE 43
5775
5776
5777 030315 123 046111 020117 EM43: .ASCIZ 'SILO OVERFLOW FAILED TO GENERATE RCVR INTERRUPT'
      030322 053117 051105 046106
      030330 053517 043040 044501
      030336 042514 020104 047524
      030344 043440 047105 051105
      030352 052101 020105 041522
      030360 051126 044440 052116
      030366 051105 052522 052120
      030374 000

5778 ;INFORMATION FOR MESSAGE 44
5779
5780
5781 030375 116 047117 042440 EM44: .ASCIZ 'NON EX MEMORY FAILED TO GENERATE XMITTR INTERRUPT'
      030402 020130 042515 047515
      030410 054522 043040 044501
      030416 042514 020104 047524
      030424 043440 047105 051105
      030432 052101 020105 046530
      030440 052111 051124 044440
      030446 052116 051105 052522
      030454 052120 000

5782 ;INFORMATION FOR MESSAGE 45
5783
5784
5785 030457 130 044515 020124 EM45: .ASCIZ 'XMIT DONE FAILED TO GENERATE XMITTR INTERRUPT'
      030464 047504 042516 043040
      030472 044501 042514 020104
      030500 047524 043440 047105
      030506 051105 052101 020105

```

030514 046530 052111 051124
030522 044440 052116 051105
030530 052522 052120 000

5786
5787
5788
5789

; INFORMATION FOR MESSAGE 46

030535 103 051125 042522
030542 052116 040440 042104
030550 042522 051523 046440
030556 046505 051117 020131
030564 040520 052124 051105
030572 051516 052040 051505
030600 020124 051105 047522
030606 020122 020055 044514
030614 042516 021440 020040
030622 000

EM46: .ASCIZ 'CURRENT ADDRESS MEMORY PATTERNS TEST ERROR - LINE # '

5790

030623 040 050050 024503
030630 020040 046540 047111
030636 053505 020122 050040
030644 052101 051124 020116
030652 020040 042524 052123
030660 020040 042040 053105
030666 042101 020122 051040
030674 043505 042101 020122
030702 020040 040527 020123
030710 020040 020040 027523
030716 000102

DM10: .ASCIZ '(PC) LINEWR PATTRN TEST DEVAOR REGADR WAS S/B'

5791
5792

030720 001116 001200 001202
030726 001160 001162 001164
030734 001166 001170 000000

.EVEN

DT5: .WORD SERRPC, STMP0, STMP1, SREG0, SREG1, SREG2, SREG3, SREG4, 0

5793
5794
5795
5796

030742 054502 042524 041440
030750 052517 052116 046440
030756 046505 051117 020131
030764 040520 052124 051105
030772 051516 052040 051505
031000 020124 051105 047522
031006 020122 020055 044514
031014 042516 021440 020040
031022 000

; INFORMATION FOR MESSAGE 47

EM47: .ASCIZ 'BYTE COUNT MEMORY PATTERNS TEST ERROR - LINE # '

5797
5798
5799

031023 124 051505 020124
031030 044524 042515 052517
031036 020124 040527 052111
031044 047111 020107 047506
031052 020122 046530 052111
031060 042040 047117 020105
031066 020055 044514 042516
031074 021440 020040 000

; INFORMATION FOR MESSAGE 50

EM50: .ASCIZ 'TEST TIMEOUT WAITING FOR XMIT DONE - LINE # '

5800

```

5801 ;INFORMATION FOR MESSAGE 51
5802
5803 031101 116 051120 046040 EMS1: .ASCIZ 'NPR LOGIC TEST 2 ERROR'
      031106 043517 041511 052040
      031114 051505 020124 020062
5804 031122 051105 047522 000122
      031130 024040 041520 020051
      031136 020040 044514 040516
      031144 052103 020040 044514
      031152 041516 045510 020040
      031160 052040 051505 020124
      031166 020040 042504 040526
      031174 051104 020040 042522
      031202 040507 051104 020040
      031210 053440 051501 020040
      031216 020040 051440 041057
      031224 000

```

```

5805 ;INFORMATION FOR MESSAGE 52
5806
5807
5808 031225 102 051501 041511 EMS2: .ASCIZ 'BASIC DATA COMPARE ERROR'
      031232 042040 052101 020101
      031240 047503 050115 051101
      031246 020105 051105 047522
      031254 000122

```

```

5809 ;INFORMATION FOR MESSAGE 53
5810
5811
5812 031256 024040 041520 020051 DH12: .ASCIZ ' (PC) SPEED (SP) TEST DEVADR REGADR WAS S/B'
      031264 020040 050123 042505
      031272 020104 020040 024040
      031300 050123 020051 020040
      031306 052040 051505 020124
      031314 020040 042504 040526
      031322 051104 020040 042522
      031330 040507 051104 020040
      031336 053440 051501 020040
      031344 020040 051440 041057
      031352 000

```

```

5813 ;INFORMATION FOR MESSAGE 55
5814
5815
5816 031353 040 050050 024503 DH13: .ASCIZ ' (PC) (PS) (SP) TEST DEVADR CHRLNG SCRNAS SCRS/B'
      031360 020040 020040 050050
      031366 024523 020040 020040
      031374 051450 024520 020040
      031402 020040 042524 052123
      031410 020040 042040 053105
      031416 042101 020122 041440
      031424 051110 047114 020107
      031432 051440 051103 040527
      031440 020123 051440 051103
      031446 027523 000102
5817

```

5818
5819
5820 031452 053117 051105 052522
031460 020116 044502 020124
031466 040506 046111 042105
031474 052040 020117 042523
031502 020124 020055 044514
031510 042516 021440 020040
031516 000

;INFORMATION FOR MESSAGE 56

EMS6: .ASCIZ 'OVERRUN BIT FAILED TO SET - LINE # '

5821
5822
5823
5824 031517 123 047524 040522
031524 042507 047440 042526
031532 043122 047514 020127
031540 044502 020124 040506
031546 046111 042105 026440
031554 046040 047111 020105
031562 020043 000040

;INFORMATION FOR MESSAGE 57

EMS7: .ASCIZ 'STORAGE OVERFLOW BIT FAILED - LINE # '

5825
5826
5827
5828 031566 005015 040515 047111
031574 042504 026503 030461
031602 042055 042132 046510
031610 040455 020040 041104
031616 030461 042040 040511
031624 047107 051517 044524
031632 006503 000012

.EVEN
;MISCELLANEOUS MESSAGES

TITLE: .ASCIZ <15><12>'MAINDEC-11-DZDMM-A DH11 DIAGNOSTIC'<15><12>

5829 031636 005015 042524 052123
031644 047111 020107 044104
031652 030461 021440 020040
031660 005015 000
5830 031663 015 052012 050131
031670 020105 041523 020122
031676 042101 051104 051505
031704 020123 047506 020122
031712 044506 051522 020124
031720 044104 030461 005015
031726 000

TITLE2: .ASCIZ <15><12>'TESTING DH11 # '<15><12>

INMSG1: .ASCIZ <15><12>'TYPE SCR ADDRESS FOR FIRST DH11'<15><12>

5831 031727 015 052012 050131
031734 020105 042526 052103
031742 051117 040440 042104
031750 042522 051523 043040
031756 051117 043040 051111
031764 052123 042040 030510
031772 006461 000012

INMSG2: .ASCIZ <15><12>'TYPE VECTOR ADDRESS FOR FIRST DH11'<15><12>

5832 031776 005015 054524 042520
032004 042040 030510 020061
032012 042504 044526 042503
032020 051440 046105 041505
032026 044524 047117 050040
032034 051101 046501 052105
032042 051105 005015 000

INMSG3: .ASCIZ <15><12>'TYPE DH11 DEVICE SELECTION PARAMETER'<15><12>

5833	032047	015	044412	053116	INMSG4: .ASCIZ <15><12>'INVALID DM11 SCR ADDRESS - TRY AGAIN'<15><12>
	032054	046101	042111	042040	
	032062	030510	020061	041523	
	032070	020123	042101	051104	
	032076	051505	020123	020055	
	032104	051124	020131	043501	
	032112	044501	006516	000012	
5834	032120	005015	047111	040526	INMSG5: .ASCIZ <15><12>'INVALID DM11 VECTOR ADDRESS - TRY AGAIN'<15><12>
	032126	044514	020104	044104	
	032134	030461	053040	041505	
	032142	047524	020123	042101	
	032150	051104	051505	020123	
	032156	020055	051124	020131	
	032164	043501	044501	006516	
	032172	000012			
5835	032174	005015	054524	042520	INMSG6: .ASCIZ <15><12>'TYPE LINE SELECTION PARAMETER'<15><12>
	032202	046040	047111	020105	
	032210	042523	042514	052103	
	032216	047511	020116	040520	
	032224	040522	042515	042524	
	032232	006522	000012		
5836	032236	005015	042504	051120	INMSG7: .ASCIZ <15><12>'DEPRESS "CONTINUE" TO START TESTING'<15><12>
	032244	051505	020123	041442	
	032252	047117	044524	052516	
	032260	021105	052040	020117	
	032266	052123	051101	020124	
	032274	042524	052123	047111	
	032302	006507	000012		
5837					
5838	032306	005015	054524	042520	VCMC: .ASCIZ <15><12>'TYPE NO. OF WORDS (OCTAL) BETWEEN VECTORS (4 OR 10)'<15><12>
	032314	047040	027117	047440	
	032322	020106	047527	042122	
	032330	020123	047450	052103	
	032336	046101	020051	042502	
	032344	053524	042505	020116	
	032352	042526	052103	051117	
	032360	020123	032050	047440	
	032366	020122	030061	006451	
	032374	000012			
5839					.EVEN
5840					;SIXTEEN CHAR COUNTERS USED BY THE AUTO ECHO TEST #3
5841					
5842	032376	000020			RCNT: .BLKW 16.
5843					
5844					;256. WORD RECEIVER INPUT BUFFER
5845					
5846	032436	000400			RBUF: .BLKW 256.
5847					
5848					
5849					;256(10) BYTE TRANSMITTER OUTPUT DATA BUFFER
5850					
5851					.EVEN
5852	033436	000400			TBUF: .BLKB 256.
5853					

H06

MAINDEC-11-DZDMM-A MACY11 27(663) 12-DEC-75 08:41 PAGE 94-17
DZDHMA.P11 POWER DOWN AND UP ROUTINES

SEQ 0278

5854

000001

.END

	2843#	2852#	2854#	2855	2864#	2865	2876#	2877#	2886#	2888#	2889	2899#	2900	
	2913#	2914#	2923#	2926#	2927#	2928	2938#	2939	2951#	2952#	2962#	2963#	2965#	
	2966	2976#	2977	2989#	2990#	3000#	3001#	3003#	3004	3015#	3016	3028#	3029#	
	3040#	3052	3074#	3075#	3086#	3098	3120#	3121#	3131#	3132	3142#	3143	3154#	
	3155#	3165#	3166	3176#	3177	3188#	3189#	3199#	3206#	3212	3234#	3235#	3245#	
	3252#	3258	3280#	3281#	3290#	3297#	3303	3325#	3326#	3335#	3342#	3348	3370#	
	3371#	3399	3423#	3467#	3507#	3547#	3587#	3635#	3679#	3692#	3703#	3749#	3770#	
	3771#	3786#	3787#	3827#	3839#	3855#	3856#	3861#	3865#	3870	3888#	3889#	3896#	
	3899#	3911	3932#	3933#	3937#	3939#	3943#	3967	3991#	4042#	4082#	4099#	4100#	
	4101#	4102#	4103#	4146#	4163#	4164#	4165#	4166#	4167#	4188#	4195	4232#	4233	
	4264#	4265	4280#	4282	4299#	4305#	4306#	4307	4311	4319#	4320#	4321	4325	
	4333#	4334#	4335	4339	4382#	4395#	4396#	4424#	4425#	4426#	4457#	4458#	4459#	
	4460	4461#	4462	4467	4469#	4480#	4481#	4483	4552#	4571#	4572#	4642#	4643#	
	4647	4652	4653#	4667#	4677#	4679	4694#	4696	4700#	4722#	4723#	4724#	4725	
	4726#	4727#	4728	4732	4733#	4742	4746	4786#	4787#	4788	4818#	4819#	4820	
	4835#	4836#	4837	4856#	4857#	4891#	4892#	4905#	4941#	4961#	4962#	4989#	4993#	
	4995#	5006#	5007	5008#	5009	5020	5032	5044	5055	5097	5101#	5102#	5103#	
	5104#	5105#	5106#	5107	5111#	5143#	5149#	5153#	5238#	5239#	5241	5257#	5258#	
	5260	5272#	5285#	5324#	5331#	5335#								
R3	=%000003	2258#	2689#	2714#	2715	2724#	2746#	2772#	2773	2784#	2785	2796#	2797	2821#
		2822	2831#	2855#	2856	2865#	2889#	2890#	2891	2900#	2901#	2902	2928#	2929
		2939#	2940	2966#	2967	2977#	2978	3004#	3005#	3006	3016#	3017#	3018	3052#
		3053	3098#	3099	3132#	3133	3143#	3166#	3167	3177#	3212#	3216	3258#	3262
		3303#	3307	3348#	3352	3398#	3399#	3400#	3404	3424#	3425#	3426#	3427	3428#
		3429#	3447#	3454#	3468#	3469#	3470#	3471	3472#	3473#	3484#	3493#	3508#	3509#
		3510#	3511	3512#	3513#	3524#	3533#	3548#	3549#	3550#	3551	3552#	3553#	3564#
		3573#	3588#	3589#	3590#	3591	3592#	3593#	3604#	3613#	3627#	3628#	3629#	3630#
		3656#	3657#	3669#	3675#	3686#	3687	3698#	3710#	3711#	3712#	3713#	3714#	3746#
		3760#	3764	3776#	3780	3800#	3801#	3802#	3809#	3810#	3814#	3821#	3823	3832#
		3833#	3835	3846#	3847#	3848#	3849#	3862#	3863#	3870#	3871#	3873	3894#	3900#
		3903#	3908#	3911#	3912#	3915	3940#	3941#	3967#	4008#	4009#	4028#	4029#	4043#
		4080#	4081#	4092#	4094	4145#	4156#	4158	4207#	4208#	4218#	4223	4265#	4266#
		4288#	4289	4311#	4325#	4339#	4379#	4390#	4391	4456#	4457	4462	4553#	4554#
		4565#	4566	4626#	4639	4647	4663	4695#	4699#	4721#	4722	4728	4788#	4789#
		4820#	4837#	4838#	4850#	4851	4890#	4906#	4907#	4937#	4938#	4954#	4957	4988#
		4989#	4992#	4995#	5033	5056	5144#	5150	5152#	5298#	5299#	5300#	5301#	5302#
		5303#	5304#	5325#	5332#	5334#								
R4	=%000004	2258#	2682#	2710#	2715	2721#	2742#	2766#	2767#	2773	2781#	2785	2792#	2793
		2797	2817#	2822	2828#	2851#	2856	2862#	2885#	2891	2897#	2902	2922#	2925#
		2929	2936#	2940	2960#	2961#	2967	2974#	2978	2998#	2999#	3006	3013#	3018
		3036#	3040	3041#	3048#	3053	3063#	3082#	3086	3087#	3094#	3099	3109#	3124#
		3131	3133	3140#	3142	3158#	3165	3167	3174#	3176	3209#	3215#	3216	3255#
		3261#	3262	3300#	3306#	3307	3345#	3351#	3352	3394#	3403#	3404	3440#	3487#
		3494#	3527#	3534#	3567#	3574#	3607#	3614#	3636#	3637#	3680#	3691#	3702#	3747#
		3748#	3759#	3763#	3764	3775#	3779#	3780	3815#	3822#	3823	3834#	3835	3860#
		3866#	3872#	3873	3895#	3897#	3904#	3906#	3913#	3914#	3915	3968#	3969#	3970#
		3971#	3972#	4006#	4007#	4026#	4027#	4041#	4083#	4084#	4093#	4094	4147#	4148#
		4157#	4158	4209#	4210#	4219#	4220#	4221#	4222#	4223	4267#	4268#	4282#	4283#
		4284#	4285#	4286#	427#	4289	4312#	4326#	4340#	4359#	4360#	4361#	4380#	4381#
		4391	4467#	4527#	4555#	4556#	4566	4639#	4640#	4641#	4642	4644	4651	4652#
		4668#	4732#	4760#	4785	4790#	4791#	4796#	4817	4821#	4826#	4834	4839#	4840#
		4845#	4851	4893#	4908#	4939#	4940#	4950#	4951#	4952#	4953#	4957	4988#	4995#
		5034	5057	5145#	5150#	5154#	5157#	5158#	5159	5326#	5332#	5336#	5336#	5336#
R5	=%000005	2258#	2637#	2650#	2651#	2658	2673#	2674#	2675#	2676#	2677#	2678#	2679#	2680#

SBD0AT	001126	2259#	2596							
SCDM1	001310	2259#								
SCDM2	001312	2259#								
SCHARC	021616	4990#								
SCHTAG	001100	2259#	2596							
SCM1	000010	2259#								
SCM2	000020	2259#								
SCM3	000010	2259#								
SCM4	000010	2259#								
SCPUOP	001256	2259#								
SCRLF	001225	2259#	4986	4987	4990	4992	4993			
SDBLK	021334	4989#								
SDDMD	001314	2259#								
SDDM1	001316	2259#								
SDDM10	001340	2259#								
SDDM11	001342	2259#								
SDDM12	001344	2259#								
SDDM13	001346	2259#								
SDDM14	001350	2259#								
SDDM15	001350	2259#								
SDDM2	001320	2259#								
SDDM3	001322	2259#								
SDDM4	001324	2259#								
SDDM5	001326	2259#								
SDDM6	001330	2259#								
SDDM7	001332	2259#								
SDDM8	001334	2259#								
SDDM9	001336	2259#								
SDEVCT	001240	2259#								
SDEVN	001306	2259#								
SDOAGN	020056	4984#								
SDTBL	021324	4989#								
SENDAD	020046	2251#	4984#							
SENDCT	020014	4984#								
SENDHC	020062	4984#								
SENULL	020077	4984#								
SENV	001250	2259#	4986	4990	4991					
SENVH	001251	2259#	2596	4990	4991					
SEOP	017760	4973#	4979	4984#						
SEOPCT	020006	4984#								
SERFLG	001103	2259#	4985#	4986#						
SERMAX	001115	2259#	2596#	4985#						
SEROR	020370	2259#	4986#							
SERAPC	001116	2259#	4986#	4987	5626	5633	5722	5738	5753	5792
SERRTB	001354	2259#	4987							
SERTY	020536	4986#	4987#							
SERTTL	001112	2259#	4986#							
SESCAP	001222	2259#	2595#	4985#	4986					
SETABL	001250	2259#								
SETEND	001354	2259#	2259#							
SFATAL	001232	2259#	4991#							
SFFLG	022066	4991#								
SFILLC	001154	2259#	4990							
SFILLS	001153	2259#	4990							

SOUES	001224	2259#	4986	4990	4992	4993								
SROCHR	022070	4992#	4994											
SRODEC=	##### U	4994												
SROLIN	022124	4992#	4994											
SROOCT	022242	4993#	4994											
SROSZ =	000010	4992#												
SREGAD	001156	2259#												
SREGO	001160	2259#	5018#	5030#	5042#	5053#	5626	5633	5722	5792				
SREG1	001162	2259#	5019#	5031#	5043#	5054#	5626	5633	5722	5792				
SREG2	001164	2259#	5020#	5032#	5044#	5055#	5626	5633	5722	5792				
SREG3	001166	2259#	5033#	5056#	5633	5792								
SREG4	001170	2259#	5034#	5057#	5633	5792								
SREG5	001172	2259#												
SREG6	001174	2259#	5021#	5022#	5035#	5036#	5270#	5283#	5626	5633	5722			
SREG7	001176	2259#												
SR2A =	##### U	4994												
SSAVRE =	##### U	4994												
SSAVR6	022576	4995#												
SSCOPE	020102	2596	4985#											
SSETUP =	000017	2595#	2596	4984	4986									
SSTUP =	177777	2595#												
SSVLAD	020316	4985#												
SSVPC =	000214	2251#												
SSMR =	165400	2237#	2244	2245	2259	2596	2647	2671	2638	2732	2755	2805	2839	2873
		2910	2948	2986	3026	3072	3118	3152	3186	3232	3278	3323	3368	3420
		3464	3504	3544	3584	3624	3725	3797	3853	3886	3929	3985	4053	4118
		4182	4238	4353	4405	4515	4594	4689	4750	4868	4917	4984	4985	4986
		4995												
SSWREG	001252	2259#	2596											
SSWRNK =	000000	2245	4985											
STESTN	001234	2259#	2647#	2671#	2648#	2732#	2755#	2805#	2839#	2873#	2910#	2948#	2986#	3026#
		3072#	3118#	3152#	3185#	3232#	3278#	3323#	3368#	3420#	3464#	3504#	3544#	3584#
		3624#	3725#	3797#	3853#	3886#	3929#	3985#	4053#	4118#	4182#	4238#	4353#	4405#
		4515#	4594#	4689#	4750#	4868#	4917#	4985#						
STIMES	001220	2259#	2596#	4984#	4985#									
STKB	001144	2259#	4992											
STKS	001142	2259#	4992											
STMP0	001200	2259#	3767#	3768#	3783#	3784#	4075#	4076#	4077#	4078#	4079#	4140#	4141#	4142#
		4143#	4144#	4421#	4632#	4717#	5050#	5051#	5269#	5282#	5371#	5626	5633	5722
		5738	5753	5792										
STMP1	001202	2259#	3191#	3206	3209	3237#	3252	3255	3283#	3328#	3373#	3389	3769#	3785#
		4785#	4796	4817#	4826	4834#	4845	4949#	4955#	5792				
STMP2	001204	2259#	3208#	3210	3213	3221	3226#	3227	3254#	3256	3259	3267	3272#	3273
		3299#	3301	3304	3312	3317#	3318	3344#	3346	3349	3357	3362#	3363	3374#
		3394	4430#	4431#	4619#	4620#	4704#	4705#	5738					
STMP3	001206	2259#	3197#	3198	3200#	3201	3243#	3244	3246#	3247	3288#	3289	3291#	3292
		3333#	3334	3336#	3337	3377#	3379	3382#	3383	4422#	4496#	4510#	4518#	4586#
		5753												
STMP4	001210	2259#	3393#	3396	3401	3409	3414#	3415	4460#	4472	5141#	5155#		
STMP5	001212	2259#	4417#	4651#	4657	5148								
STMP6	001214	2259#	4416#	4538#	4564#	4725#	4737	4744	5147	5158	5323#	5337#		
STMP7	001216	2259#	3030#	3069	3076#	3115	3757#	3758	3761	3769	3777	3785	3791#	3792
		3857#	3859	3880#	3881	3890#	3893	3903	3922#	3923	3934#	3936	3968	3972
		3975	3980#	3981	4017#	4018#	4058#	4064	4075	4099	4111#	4112	4123#	4129

ADD	2626	2627	2628	2651	2656	2678	2679	2680	2809	2843	2877	2914	2952	2990	3029
	3041	3063	3075	3087	3109	3121	3155	3189	3235	3281	3326	3371	3628	3679	3692
	3703	3711	3713	3771	3787	3827	3839	3848	3856	3889	3933	4042	4111	4175	4300
	4306	4320	4334	4396	4468	4469	4572	4653	4700	4733	4734	4787	4819	4836	4857
	4892	4962	4975	4976	4977	4987	4988	4989	4990	4991	4993	5022	5036	5081	5106
	5109	5153	5154	5237	5240	5256	5259	5300	5303	5317	5335	5336			
ASL	2631	2706	2738	2762	2813	2847	2881	2918	2956	2994	3433	4141	4142	4164	4165
	4461	4643	4726	4978	4987	4993	4994	5077							
ASL B	4989														
ASR	4077	4078	4101	4102	4991	5102	5103	5104							
BCC	4989														
BEQ	2596	2609	2613	2632	2686	2690	2707	2716	2725	2739	2747	2763	2774	2786	2798
	2814	2823	2832	2848	2857	2866	2882	2892	2903	2919	2930	2941	2957	2968	2979
	2995	3007	3019	3054	3100	3134	3144	3168	3178	3192	3214	3217	3228	3238	3260
	3263	3274	3305	3308	3319	3350	3353	3364	3375	3402	3405	3416	3434	3676	3688
	3699	3765	3781	3824	3836	3874	3916	3957	3961	3982	4023	4038	4224	4290	4308
	4322	4336	4392	4413	4450	4463	4491	4505	4526	4529	4567	4587	4603	4606	4645
	4648	4664	4729	4759	4762	4779	4811	4831	4884	4958	4979	4981	4984	4985	4986
	4987	4988	4990	4991	4993	5078	5080	5171	5173	5175	5194	5202	5210	5217	
BGE	4985	4990	5230	5249											
BGT	4581	4984	4988	4989	4993										
BHI	4985														
BIC	2723	2780	2781	2783	2795	2830	2864	2890	2899	2901	2925	2927	2961	2965	2999
	3003	3005	3017	3400	3657	3833	3865	3871	3899	3912	3943	4009	4029	4079	4081
	4103	4144	4167	4208	4266	4283	4431	4459	4483	4554	4583	4620	4621	4641	4679
	4682	4705	4724	4745	4746	4789	4838	4907	4938	4984	4988	4992	4993	5105	5108
BICB	3378	3380	3395	5084	5124	5136									
BIS	2688	2767	2769	2771	2793	2854	2888	2936	2938	2974	2976	3013	3015	3441	3478
	3479	3518	3519	3558	3559	3598	3599	3644	3645	3730	3806	3861	3896	3939	4195
	4252	4940	4953	4988	4989										
BISB	3038	3050	3084	3096	3125	3159	3379	3388	3389	3396	3637	3641	3733	3748	3972
	4007	4027	4061	4084	4126	4148	4191	4196	4210	4220	4222	4247	4268	4285	4286
	4360	4362	4381	4556	4614	4875	4952	4987							
BIT	2629	2704	2736	2760	2811	2845	2879	2916	2954	2992	3431	3981	4000	4022	4449
	4528	4605	4761	4980	4985	4986	5079	5235	5254						
BITB	2596	4990	4991												
BLO	4095	4159													
BLOS	4992														
BLT	4988	4989	4990	4993	5233	5252									
BMI	3666	3740	3946	4070	4135	4201	4258	4373	4443	4545	4582	4627	4712	4932	4989
BNE	2596	2604	2615	2630	2659	2705	2737	2761	2812	2846	2880	2917	2955	2993	3035
	3047	3081	3093	3202	3248	3293	3338	3384	3432	3762	3778	3793	3811	3864	3868
	3882	3898	3901	3907	3909	3924	3942	3948	3976	4001	4021	4113	4177	4234	4278
	4302	4482	4678	4681	4684	4703	4743	4747	4852	4956	4985	4986	4987	4988	4989
	4990	4991	4992	4995	5010	5071	5123	5135	5156	5160	5236	5242	5255	5261	5314
	5316	5338													
BPL	4902	4986	4988	4989	4990	4992	5222								
BR	2596	2633	2660	2667	2695	2708	2729	2740	2752	2764	2778	2790	2802	2815	2836
	2849	2870	2883	2907	2920	2934	2945	2958	2972	2983	2996	3011	3023	3033	3042
	3045	3064	3067	3079	3088	3091	3110	3113	3123	3149	3157	3183	3194	3229	3240
	3275	3285	3320	3330	3365	3387	3417	3435	3444	3490	3530	3570	3610	3632	3652
	3663	3673	3684	3696	3707	3717	3729	3742	3755	3794	3818	3830	3954	3965	3978
	3989	4003	4015	4035	4050	4057	4072	4090	4114	4122	4137	4154	4178	4186	4203
	4216	4235	4242	4243	4260	4274	4357	4375	4388	4402	4438	4447	4454	4478	4493

	4507	4523	4531	4547	4562	4578	4584	4600	4608	4629	4637	4714	4719	4740	4756
	4764	4765	4781	4798	4813	4828	4872	4886	4899	4914	4921	4934	4947	4968	4985
	4986	4987	4988	4989	4990	4991	4992	4993	4995	5075	5082	5176	5178	5196	5204
	5219	5231	5234	5243	5250	5253	5262								
CLC	5279	4524	4601	4757											
CLR	2593	2594	2596	2611	2682	2711	2721	2742	2743	2768	2818	2828	2852	2862	2886
	2897	2923	2962	3000	3036	3048	3066	3082	3094	3112	3140	3174	3199	3215	3245
	3261	3297	3300	3342	3345	3381	3403	3485	3486	3495	3496	3525	3526	3535	3536
	3565	3566	3575	3576	3605	3606	3615	3616	3649	3680	3702	3714	3738	3757	3763
	3767	3775	3783	3845	3849	3860	3934	3993	3999	4026	4063	4068	4128	4133	4188
	4199	4256	4312	4326	4371	4409	4489	4503	4518	4521	4538	4543	4551	4570	4588
	4598	4625	4699	4710	4754	4777	4809	4821	4882	4893	4930	4984	4985	4987	4988
	4989	4993	4995	5006	5050	5083	5117	5119	5120	5130	5143	5301	5304	5324	5327
	5340	5363													
CLRB	3197	3208	3243	3254	3288	3299	3333	3344	3377	3393	3971	4982	4985	4989	4990
CNP	4991	4992	4993	5074	5142	5149	5161	5331							
	2596	2614	2658	2663	2685	2715	2773	2785	2797	2822	2856	2891	2902	2929	2940
	2967	2978	3006	3018	3053	3099	3133	3167	3216	3262	3307	3352	3404	3687	3764
	3780	3792	3823	3835	3873	3881	3915	3923	3960	4094	4112	4158	4176	4223	4233
	4277	4289	4301	4335	4391	4412	4462	4566	4647	4663	4728	4742	4851	4957	4985
CNPB	4989	4992	5009	5122	5134	5159	5172	5174	5229	5232	5241	5248	5251	5260	
	3201	3213	3227	3247	3259	3273	3292	3304	3318	3337	3349	3363	3383	3401	3415
	3761	3777	4037	4644	4778	4830	4985	4986	4990	4991	4992	4993			
COM	2607	4586													
COMB	5072														
DEC	3810	3863	3867	3897	3900	3906	3908	3941	3944	4020	4411	4580	4702	4955	4984
	4987	5155	5315	5337											
DECB	4988	4990													
ENT	2258														
HALT	2247	4986	4990	4995	5225										
INC	2625	3779	3791	3866	3880	3922	3938	3969	4232	4299	4418	4481	4564	4701	4727
	4974	4984	4985	4986	4988	4989	4991	4995	5008	5121	5133	5151	5152	5313	5333
	5334														
INCB	3200	3226	3246	3272	3291	3317	3336	3362	3382	3414	4480	4677	4985	4986	4990
	5076														
IOT	2258														
JMP	2247	2248	2249	2616	2617	4244	4347	4433	4499	4513	4590	4660	4675	4685	4686
	4766	4847	4853	4863	4983	4984	5185	5188	5226	5280	5293				
JSR	2610	2637	2642	2662	2692	2718	2727	2749	2776	2788	2800	2825	2834	2859	2868
	2894	2905	2932	2943	2970	2981	3009	3021	3032	3044	3056	3057	3078	3090	3102
	3103	3122	3126	3136	3146	3156	3160	3170	3180	3193	3219	3220	3239	3265	3266
	3284	3310	3311	3329	3355	3356	3386	3407	3408	3438	3442	3446	3448	3453	3455
	3461	3476	3480	3483	3488	3492	3497	3501	3516	3520	3523	3528	3532	3537	3541
	3556	3560	3563	3568	3572	3577	3581	3596	3600	3603	3608	3612	3617	3621	3631
	3638	3639	3646	3651	3655	3658	3659	3668	3670	3671	3678	3681	3682	3690	3693
	3694	3701	3704	3705	3715	3719	3728	3731	3732	3741	3745	3750	3751	3772	3788
	3803	3808	3813	3816	3826	3828	3838	3840	3844	3876	3918	3950	3951	3952	3959
	3962	3963	3988	4002	4005	4010	4011	4025	4030	4031	4040	4044	4045	4056	4071
	4085	4086	4098	4104	4105	4121	4136	4149	4150	4162	4168	4169	4185	4202	4206
	4211	4212	4226	4227	4241	4259	4263	4269	4270	4292	4293	4310	4313	4314	4324
	4327	4328	4338	4341	4342	4356	4374	4378	4383	4384	4394	4397	4398	4420	4423
	4432	4436	4437	4465	4470	4471	4474	4492	4506	4530	4532	4546	4550	4557	4558
	4569	4573	4574	4607	4612	4628	4633	4650	4655	4656	4666	4670	4671	4713	4731
	4735	4736	4763	4768	4780	4784	4792	4793	4812	4816	4822	4823	4833	4841	4842

MOV	4855	4858	4859	4871	4874	4885	4889	4894	4895	4904	4909	4910	4920	4933	4936
	4943	4943	4960	4963	4964	4984	4986	4990	4991	5016	5028	5195	5203	5275	5279
	2600	2601	2602	2602	2619	2620	2621	2622	2623	2634	2635	2636	2644	2647	2647
	2650	2650	2650	2653	2664	2669	2671	2672	2673	2676	2677	2684	2687	2689	2689
	2700	2701	2702	2701	2710	2711	2712	2713	2714	2724	2725	2731	2745	2750	2750
	2757	2759	2766	2759	2770	2771	2772	2773	2774	2780	2781	2787	2808	2810	2817
	2820	2821	2822	2832	2834	2843	2846	2847	2853	2855	2857	2869	2876	2878	2885
	2887	2888	2889	2890	2891	2892	2894	2895	2903	2904	2911	2913	2919	2929	2931
	2932	2935	2936	2937	2940	2967	2968	2969	2987	2990	2991	2994	2995	2996	2998
	3006	3008	3017	3018	3019	3028	3041	3043	3053	3055	3058	3059	3060	3062	3064
	3065	3067	3075	3080	3082	3083	3092	3093	3099	3110	3118	3120	3123	3124	3125
	3127	3129	3130	3132	3140	3145	3146	3147	3156	3157	3163	3174	3182	3184	3187
	3189	3190	3192	3193	3194	3198	3207	3209	3218	3219	3238	3240	3245	3246	3248
	3249	3250	3251	3253	3255	3254	3265	3267	3276	3280	3281	3288	3304	3305	3311
	3319	3325	3333	3339	3340	3353	3355	3358	3359	3363	3364	3365	3366	3367	3368
	3370	3379	3380	3382	3390	3395	3405	3407	3408	3414	3415	3416	3417	3419	3421
	3422	3424	3427	3427	3428	3435	3435	3445	3452	3456	3457	3460	3466	3467	3488
	3496	3497	3503	3510	3511	3515	3517	3519	3520	3527	3534	3535	3537	3539	3540
	3542	3552	3553	3555	3565	3571	3589	3594	3596	3604	3604	3610	3611	3615	3617
	3618	3622	3624	3626	3632	3639	3642	3651	3652	3654	3667	3668	3669	3689	3691
	3692	3693	3694	3696	3696	3697	3698	3706	3707	3709	3717	3721	3722	3725	3732
	3744	3750	3752	3753	3760	3767	3771	3772	3773	3774	3776	3785	3786	3788	3790
	3796	3800	3802	3803	3804	3805	3806	3808	3817	3818	3820	3826	3834	3835	3837
	3839	3845	3850	3856	3868	3870	3873	3876	3877	3878	3879	3881	3890	3891	3905
	3906	3917	3919	3922	3924	3925	3926	3927	3929	3937	3941	3949	3954	3961	3973
	3984	3985	3986	3987	3988	3989	3990	3991	3992	3993	3994	3995	4005	4006	4009
	5020	5021	5030	5031	5032	5033	5034	5035	5042	5043	5044	5053	5054	5055	5056
	5057	5073	5097	5098	5099	5101	5111	5132	5141	5144	5145	5146	5147	5148	5157
	5156	5170	5177	5179	5184	5187	5193	5201	5209	5211	5216	5218	5220	5238	5239
	5257	5258	5269	5270	5271	5272	5274	5276	5282	5283	5284	5285	5287	5289	5298
	5299	5302	5323	5325	5326	5328	5329	5330	5339	5347	5348	5355	5356	5364	5365
MOV8	2596	2712	2722	2744	2770	2782	2819	2829	2853	2863	2924	2937	2964	2975	3039
	3051	3085	3097	3130	3141	3164	3175	3195	3198	3204	3205	3210	3211	3241	3244
	3250	3251	3256	3257	3286	3289	3295	3296	3301	3302	3331	3334	3340	3341	3346
	3347	3426	3429	3470	3473	3510	3513	3550	3553	3590	3593	3630	3758	3768	3784
	3802	3992	4282	4426	4430	4533	4619	4704	4769	4801	4908	4923	4939	4950	4985

	4986	4988	4989	4990	4991	4992	4993	4994	5007	5017	5029	5051	5052	5100	5107
	5110	5118	5131	5150	5273	5286	5332								
NEG	4988	4989													
NOP	3443	3481	3521	3561	3601	3650	4349	4864	4984						
RESET	4984	5278	5291												
ROL	3980	4525	4602	4758	4988	4993									
RTI	3451	3458	4484	4985	4986	4988	4989	4990	4992	4993	4995	5349	5357	5369	
RTS	3722	3973	4987	4990	4991	4994	5011	5023	5037	5045	5058	5085	5112	5125	5137
	5162	5180	5246	5265	5305	5318	5341	5350	5358	5372					
SEC	3977	4522	4599	4755											
SUB	2654	3893	4410	4536	4616	4986	4989	4991							
SWAB	3872	3914	3970	4076	4100	4143	4166	4221	4284	4297	4361	4458	4640	4723	4791
	4840	4951													
TRAP	4994	5366													
TST	2596	2603	2608	2612	2657	2681	3665	3739	3905	3947	3956	3975	4069	4307	4321
	4490	4504	4680	4810	4883	4931	4985	4986	4987	4988	4989	4990	4991	4993	4994
	5221														
TSTB	3034	3046	3080	3092	3427	3471	3511	3551	3591	3945	4134	4200	4257	4372	4442
	4544	4683	4711	4901	4985	4989	4990	4991	4992	5070					
.ASCII	2259														
.ASCIZ	2259	4984	4987	4995	5623	5624	5630	5631	5637	5641	5645	5649	5653	5657	5661
	5665	5669	5673	5677	5681	5685	5686	5690	5694	5698	5702	5703	5707	5711	5715
	5719	5720	5726	5730	5731	5735	5736	5742	5746	5750	5751	5757	5761	5765	5769
	5773	5777	5781	5785	5789	5790	5796	5799	5803	5804	5808	5812	5816	5820	5824
	5828	5829	5830	5831	5832	5833	5834	5835	5836	5838					
.BLKB	4992	5852													
.BLKH	4989	5395	5499	5842	5846										
.BYTE	2259	4984	4986	4988	4991	4992	5592	5593							
.ENABL	2243														
.END	5854														
.ENDC	2244	2245	2247	2251	2252	2258	2259	2595	2596	2647	2671	2686	2690	2698	2707
	2732	2739	2755	2763	2805	2814	2839	2848	2873	2882	2910	2919	2948	2957	2986
	2995	3026	3033	3045	3072	3079	3091	3118	3123	3134	3152	3157	3168	3186	3192
	3232	3238	3278	3285	3323	3330	3368	3375	3420	3464	3504	3544	3584	3624	3717
	3725	3729	3797	3853	3886	3929	3985	3989	4053	4057	4118	4122	4182	4186	4238
	4242	4322	4353	4357	4405	4413	4438	4515	4531	4582	4587	4594	4608	4689	4719
	4740	4750	4764	4868	4872	4917	4921	4984	4985	4986	4987	4988	4989	4990	4991
	4992	4993	4994	4995	4998	5000	5375	5377	5617	5619					
.EQUIV	2258														
.EVEN	2259	4987	4991	4995	5623	5632	5721	5737	5752	5791	5825	5839	5851		
.IF	2244	2245	2247	2251	2252	2258	2259	2595	2596	2647	2671	2686	2690	2698	2707
	2732	2739	2755	2763	2805	2814	2839	2848	2873	2882	2910	2919	2948	2957	2986
	2995	3026	3033	3045	3072	3079	3091	3118	3123	3134	3152	3157	3168	3186	3192
	3232	3238	3278	3285	3323	3330	3368	3375	3420	3464	3504	3544	3584	3624	3717
	3725	3729	3797	3853	3886	3929	3985	3989	4053	4057	4118	4122	4182	4186	4238
	4242	4322	4353	4357	4405	4413	4438	4515	4531	4582	4587	4594	4608	4689	4719
	4740	4750	4764	4868	4872	4917	4921	4984	4985	4986	4987	4988	4989	4990	4991
	4992	4993	4994	4995	4998	5000	5375	5377	5617	5619					
.IFF	2245	2251	2252	2258	2259	2596	2647	2671	2686	2690	2698	2707	2732	2739	2755
	2763	2805	2814	2839	2848	2873	2882	2910	2919	2948	2957	2986	2995	3026	3033
	3045	3072	3079	3091	3118	3123	3134	3152	3157	3168	3186	3192	3232	3238	3278
	3285	3323	3330	3368	3375	3420	3464	3504	3544	3584	3624	3717	3725	3729	3797
	3853	3886	3929	3985	3989	4053	4057	4118	4122	4182	4186	4238	4242	4322	4353
	4357	4405	4413	4438	4515	4531	4582	4587	4594	4608	4689	4719	4740	4750	4764

	4868	4872	4917	4921	4984	4985	4986	4987	4988	4989	4990	4991	4992	4993	4994
.IFT	4993	4998	5000	5375	5377	5617	5619								
.IFTF	4985	4986	4992	4993											
.IIF	2244	2245	2246	2247											
.IRP	2259	2247	2246	2247	2259	2596	4984	4985	4986	4987	4990	4992	4993	4994	3118
	3152	3186	3232	3278	3323	3368	3420	3464	3504	3544	3584	3624	3725	3797	3853
	3886	4053	4118	4053	4118	4182	4238	4353	4405	4515	4594	4689	4750	4868	4917
.LIST	4985	4989	4991	4993	4995	5097	5111								
	2233	2236	2245	2247	2258	2259	2596	2647	2671	2698	2732	2755	2805	2839	2839
	2873	2910	2948	2986	3026	3072	3118	3152	3186	3232	3278	3323	3368	3420	3464
	3504	3544	3584	3624	3725	3797	3853	3886	3929	3985	4053	4118	4182	4238	4353
.MACRO	4405	4515	4594	4689	4750	4868	4917	4984	4985	4986	4992	4994	4994	4994	4994
	45	45	93	149	185	245	297	350	405	448	491	534	633	732	773
	814	859	904	946	988	1052	1100	1154	1199	1244	1287	1381	1432	1493	1537
	1572	1622	1675	1732	1763	1817	1883	1917	1950	2015	2047	2078	2136	2184	2245
.MCALL	2253	2259	4994												
.MLIST	2238	2239	2240	2241	2242	2258	2259	2259	2596	2647	2671	2698	2732	2755	2805
	2873	2910	2948	2986	3026	3072	3118	3152	3186	3232	3278	3323	3368	3420	2839
	3504	3544	3584	3624	3725	3797	3853	3886	3929	3985	4053	4118	4182	4238	3464
.PAGE	4405	4515	4594	4689	4750	4868	4917	4984	4985	4986	4992	4994	4994	4994	4994
	2259	2592	2598	2646	2670	2697	2731	2754	2804	2838	2872	2909	2947	2985	3025
	3071	3117	3151	3185	3231	3277	3322	3367	3419	3463	3503	3543	3583	3623	3724
	3796	3852	3885	3928	3984	4052	4117	4181	4237	4352	4404	4514	4593	4688	4749
.REM	4867	4916	4971	4997	5616										
	2648	2672	2699	2733	2756	2806	2840	2874	2911	2949	2987	3027	3073	3119	3153
	3187	3233	3279	3324	3369	3421	3465	3505	3545	3585	3625	3726	3798	3854	3887
	3930	3986	4054	4119	4183	4239	4354	4406	4516	4595	4690	4751	4869	4918	
.REPT	2247	2259													
.SBTTL	2245	2247	2251	2252	2258	2259	2647	2671	2698	2732	2755	2805	2839	2873	2910
	2948	2986	3026	3072	3118	3152	3186	3232	3278	3323	3368	3420	3464	3504	3544
	3584	3624	3725	3797	3853	3886	3929	3985	4053	4118	4182	4238	4353	4405	4515
	4594	4689	4750	4868	4917	4984	4985	4986	4987	4988	4989	4990	4991	4992	4993
.TITLE	2244	2251	2252	2259	4984	4987	4988	4990	4991	4993	4995	5626	5633	5722	5738
.WORD	2247	2251	2252	2259	4984	4987	4988	4990	4991	4993	4995	5626	5633	5722	5738
	5753	5792													

ERRORS DETECTED: 0
 #DZDHMA,DZDHMA/CRF=DHMMAC,DZDHMA
 RUN-TIME: 40 30 4 SECONDS
 CORE USED: 37K

