

DM11A

LOGIC TEST
CZDMA D0

AH-8537D-MC

COPYRIGHT © 72-78

FICHE 1 OF 1

AUG 1978

digital

MADE IN USA

The microfiche card contains 48 frames of data. The first 47 frames are organized into a grid of 8 rows and 6 columns. Each frame contains technical information, including logic test results, component values, and other data relevant to the DM11A system. The 48th frame is blank. The data is printed in white on a dark background.

REM %

IDENTIFICATION

PRODUCT CODE: AC-8536D-MC
PRODUCT NAME: CZDMADO DM11A LGC TST
DATE RELEASED: APRIL 1978
MAINTAINER: DIAGNOSTIC GROUP

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

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1972, 1978 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

1. ABSTRACT

TWO SEPARATE DIAGNOSTIC PROGRAMS ARE PROVIDED FOR TESTING THE DM11A (ASYNCHRONOUS DATA MULTIPLEXER), CZDMA (DM11A LOGIC TESTS), AND CZDMB (DM11A MULTIPLE LINE DATA TESTS). THE LOGIC TESTS INDIVIDUALLY TEST EACH OF THE 16 DM11 LINES AND ALL COMMON LOGIC. THE MULTIPLE LINE DATA TESTS RUN SEVERAL LINES CONCURRENTLY AND ARE USED TO TEST LINE INTERACTION AND DATA TRANSMISSION/RECEPTION RELIABILITY. THIS DOCUMENT DESCRIBES THE LOGIC TESTS.

THE AVAILABLE TESTS ARE:

- PRG0 - LOGIC TEST
- PRG1 - TRANSMITTER SCOPE LOOP
- PRG2 - TRANSMIT/RECEIVE SCOPE LOOP

2. REQUIREMENTS

2.1 EQUIPMENT

- A. PDP 11 FAMILY PROCESSOR
- B. DM11
- C. JUMPERS CONNECTING 16 TRANSMITTERS TO THEIR RESPECTIVE RECEIVERS.

2.2 STORAGE

THIS PROGRAM USES ALL OF CORE (8K) EXCEPT THAT AREA RESERVED FOR THE LOADERS.

3. LOADING PROCEEDURE

THE ABSOLUTE LOADER IS USED TO LOAD THE PROGRAM.

4. USE PROCEDURE

4 1 STARTING PROCEDURE

BEFORE STARTING MAKE SURE THAT THE TTY IS IN REMOTE MODE, AND THE JUMPERS ARE INSTALLED. THREE STARTING ADDRESSES ARE PROVIDED

0200 - THIS STARTING ADDRESS REQUESTS DM11 PARAMETERS, AND MUST BE USED TO INITIALLY START THE PROGRAM, AND WHENEVER ANY OF THE PARAMETERS LISTED BELOW IS CHANGED.

A. VECTOR ADDRESS ?

RESPONSE: TYPE IN THE VECTOR ADDRESS OF THE DM11 RECEIVER UNDER TEST. CARRIAGE RETURN SELECTS 0300

B. UNIT #(8)?

RESPONSE: THE DM11 UNIT NUMBER CORRESPONDS TO THE ADDRESS TO WHICH THE CONTROL STATUS REGISTER (CSR) RESPONDS.

CSR ADDRESS	DM11 UNIT #	CSR ADDRESS	DM11 UNIT #
175000	0	175100	10
175010	1	175110	11
175020	2	175120	12
175030	3	175130	13
175040	4	175140	14
175050	5	175150	15
175060	6	175160	16
175070	7	175170	17

CARRIAGE RETURN SELECTS UNIT # 0

C. WHAT IS THE CHARACTER LENGTH?

RESPONSE: CHARACTER LENGTH REFERS TO THE NUMBER OF DATA BITS PER CHARACTER (5-8). CARRIAGE RETURN DEFAULTS A CHARACTER LENGTH OF "8". IF A CHARACTER LENGTH 5-7 IS DESIRED, TYPE THE VALUE(5-7) OF THE DESIRED LENGTH AT THE KEYBOARD WHEN PROMPTED.

D. PRG #

RESPONSE: TYPE PROGRAM NUMBER OF PROGRAM YOU WISH TO RUN. CARRIAGE RETURN SELECTS PROGRAM # 0.

NOTES:

CARRIAGE RETURN TERMINATES ALL RESPONSES
 ANY UNACCEPTABLE RESPONSE WILL RESULT IN A ? TYPEOUT AND THE PARAMETER WILL AGAIN BE REQUESTED.

0204 - THIS STARTING ADDRESS USES PREVIOUSLY DEFINED DM11 PARAMETERS AND REQUESTS THE PROGRAM NUMBER OF THE PROGRAM YOU WISH TO RUN.

0210 - THIS STARTING ADDRESS STARTS THE PREVIOUSLY SELECTED PROGRAM USING PREVIOUSLY SELECTED PARAMETERS.

4.2 SWITCH SETTINGS

THE FOLLOWING SWITCH SETTINGS APPLY TO PROGRAM #0.

SR 0-6	ROUTINE TO BE RUN (IF ENABLED BY SR-9)
SR 8	RING BELL ON ERROR
SR 9	LOOP SELECTED ROUTINE
SR 11	INHIBIT ITERATION (DO EACH ROUTINE ONCE)
SR 13	INHIBIT PRINTOUT
SR 14	SCOPE (LOOP ROUTINE)
SR 15	HALT ON ERROR

THIS PROGRAM HAS BEEN MODIFIED TO RUN ON A PROCESSOR WITH OR WITHOUT A HARDWARE SWITCH REGISTER. WHEN FIRST EXECUTED THE PROGRAM TESTS THE EXISTENCE OF A HARDWARE SWITCH REGISTER. IF NOT FOUND A SOFTWARE SWITCH REGISTER LOCATION (SWREG=LOC. 176) IS DEFAULTED TO. IF THIS IS THE CASE, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

(IE) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

1. <CR> IF NO CHANGES ARE TO BE MADE
2. 6 DIGITS 0-7 TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER VALUE ;LAST DIGIT FOLLOWED BY <CR>.
3. U TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED KEYING IN SWREG VALUE.

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING G (CNTRL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM CODE (IE) ERROR ROUTINES, AFTER HALTS END OF PASS, AND OTHER APPLICABLE AREAS.

5.0 PROGRAM DESCRIPTION

5.1 PRGO - LOGIC TESTS

PRGO CONSISTS OF 152(8) INDEPENDENT ROUTINES WHICH TEST VARIOUS FUNCTIONS OF THE DM11 HARDWARE. ANY OF THESE ROUTINES MAY BE INDIVIDUALLY SELECTED AND RUN (SEE SEC 4.2 FOR SWITCH SETTING)

5.1.1 ROUTINE DESCRIPTION

ROUTINE TESTS

RTO TESTS THE ABILITY TO REFERENCE THE FOUR DM11 REGISTERS CONTROL STATUS REGISTER (CSR), BUFFER ACTIVE REGISTER (BAR), BREAK STATUS REGISTER (BKCSR), AND THE BASE REGISTER (BASREG). IF AN ILLEGAL REFERENCE OCCURS WHEN THE CSR IS REFERENCED THE PROGRAM WILL INDICATE AN ERROR, AND AUTOMATICALLY LOOP THE ERROR AS LONG AS THE ERROR CONDITION EXISTS.
RTO PC=XXXXXX

RT1-RT10 BIT 'BANGS' THE CSR (BITS 0,1,2,4,5,6,12,13), TESTING THAT EACH BIT IN THE CSR CAN BE INDIVIDUALLY SET AND CLEARED. TWO ERROR TYPES ARE DETECTED IN THESE TESTS, A BIT FAILED TO SET, AND/OR A BIT FAILED TO CLEAR. THE ERROR PRINTOUT SHOWS THE ROUTINE THAT FAILED AND THE PC WHERE THE ERROR WAS DETECTED.

RT11- TESTS THAT RESET AND CLEAR CLEAR ALL R/W BITS IN THE CSR. TWO ERROR TYPES ARE DETECTED IN THIS ROUTINE SHOWING THE CONTENTS OF THE CSR AFTER THE RESET & CLEAR INSTRUCTION. THE PROGRAM AUTOMATICALLY LOOPS IF AN ERROR OCCURS. SHOWN BELOW IS THE ERROR TYPEOUT
RT11 PC=XXXXXX ERR S/B: 000000 HAS: XXXXXX

RT12 LOADS A BINARY COUNT PATTERN INTO THE BKCSR AND READS BACK THE RESULTS. IF THE DATA READ BACK IS INCORRECT AN ERROR IS INDICATED THE SCOPE SWITCH WILL CAUSE THE PROGRAM TO RELOAD THE BINARY NUMBER AND REPEAT THE TEST. THE ERROR TYPEOUT SHOWS CORRECT AND ACTUAL RESULTS.
THE SECOND PORTION OF THE TEST CLEARS THE PREVIOUSLY LOADED NUMBER IF THE SCOPE SWITCH IS SET THE PROGRAM LOOPS BACK AND REPEATS THE CLEAR INSTRUCTION.

RT13 THIS ROUTINE LOADS RANDOM NUMBERS INTO THE BKCSR. IF A RANDOM NUMBER IS LOADED INCORRECTLY AN ERROR IS INDICATED SHOWING THE CORRECT AND ACTUAL RESULTS.

RT14 THIS ROUTINE TESTS THAT RESET WILL CLEAR ALL BREAK STATUS REGISTER (BKCSR) BITS. IF ALL BITS DO NOT CLEAR WHEN THE RESET IS GIVEN AN ERROR IS INDICATED. THE ERROR TYPEOUT SHOWS THE CORRECT RESULT (ALL 0'S) AND THE ACTUAL RESULT.

RT15-RT16 THESE ROUTINES ARE THE SAME AS RT12 & RT13 EXCEPT THAT THE BASE REGISTER IS TESTED.

RT17 THIS ROUTINE TESTS THAT ALL BAR BITS CAN BE INDIVIDUALLY SET AND CLEARED. THE ROUTINE SHIFTS A '1' THROUGH THE BAR THEREBY SETTING EACH BAR BIT AND THEN THE BAR BIT IS CLEARED. THE ERROR TYPEOUTS SHOW CORRECT AND ACTUAL RESULTS.

RT20 THIS ROUTINE TESTS THAT RESET AND CLEAR CLEAR ALL BAR BITS THE ERROR TYPEOUT SHOWS CORRECT AND ACTUAL RESULTS.

RT21-RT23 THESE ROUTINES TEST THAT THE CSR, BAR, AND BKCSR RESPOND PROPERLY TO BYTE COMMANDS. BOTH BYTES ARE REFERENCED IN THESE ROUTINES USING CLAB INSTRUCTIONS. THE ERROR TYPEOUT SHOWS CORRECT AND ACTUAL RESULTS.

RT24 THIS ROUTINE TESTS THAT THE DM11 CAN INTERRUPT THE PROCESSOR VIA THE OVER RUN BIT (CSR BIT 13). THE ERROR TYPEOUT SHOWS THE ROUTINE NUMBER AND THE PC WHERE THE ERROR WAS DETECTED.

RT25 THIS ROUTINE TESTS THAT THE DM11 INTERRUPTS THE PROCESSOR AT THE PROPER LEVEL.

RT26-RT45 THESE ROUTINES TEST THE BASIC TRANSMITTER FUNCTIONS ON EACH LINE

RT46-RT65 THESE ROUTINES TEST THE BASIC RECEIVER FUNCTIONS ON EACH LINE

RT66 THIS ROUTINE TESTS THAT THE DM11 WILL SET THE NEX BIT (CSR BIT 14). WHEN THE DM11 TRIES TO TRANSMIT FROM NON-EXISTANT MEMORY. ALL LINES ARE INDIVIDUALLY TRANSMITTED ON. THE ERROR TYPEOUT SHOWS THE FAILING LINE. ALSO TESTED IS THAT THE NEX BIT WHEN SET CAUSES AN INTERRUPT.

RT67 THIS ROUTINE TESTS THAT THE NEX BIT (CSR BIT 14) SETS WHEN THE DM11 TRIES TO REFERENCE THE TUMBLE TABLE THAT IS IN NON-EXISTANT MEMORY.

RT70 THIS ROUTINE TESTS THAT WHEN THE GO BIT (CSR BIT 0) IS CLEAR THAT NO DATA IS RECEIVED ON ANY LINE. ALL LINES ARE TRANSMITTED ON AND AFTER THE TRANSMISSION IS COMPLETE THE RECEIVER DONE FLAG IS TESTED. THE ERROR TYPEOUT SHOWS THE LINE ON WHICH DATA WAS RECEIVED.

THE TYPEOUT SHOWN BELOW SHOWS THAT DATA WAS RECEIVED ON LINE 0
 RT70 PC=XXXXXX ERRS/B: 000001 WAS: 000001

RT71 THIS ROUTINE TESTS THAT THE CURRENT ADDRESS IS INCREMENTED PROPERLY BY THE DM11. THE TABLE BELOW SHOWS THE ADDRESS LOADED INTO IN THE CURRENT ADDRESS TABLE BEFORE 2 CHARACTERS ARE TRANSMITTED AND THE RESULTANT ADDRESS AFTER THE CHARACTER IS TRANSMITTED

BEFORE	AFTER	BEFORE	AFTER
000000	000001	000777	001000
000001	000002	001777	002000
000003	000004	003777	004000
000007	000010	007777	010000
000017	000020	017777	020000
000037	000040	037777	040000
000077	000100	077777	100000
000177	000200	177777	000000

000377 000400

THE ERROR TYPEOUT SHOWS CORRECT AND ACTUAL CURRENT ADDRESS.

RT72 THIS ROUTINE TESTS THAT DATA CAN BE TRANSMITTED FROM ALL AVAILABLE CORE AND RECEIVED CORRECTLY. THIS IS DONE BY TRANSMITTING 1 CHARACTER FROM SEVERAL ADDRESSES IN EACH 4K BLOCK OF CORE ON LINE 0. THE ERROR TYPEOUT WILL SHOW TRANSMITTED AND ACTUAL RECEIVED DATA. IF A DATA ERROR RESULTED WHEN TRANSMITTING FROM THE FIRST 4K OF CORE EXAMINE THE CURRENT ADDRESS OF LINE 0 TO DETERMINE WHERE IN THE FIRST 4K OF CORE THE DM11 WAS TRANSMITTING FROM WHEN ERROR OCCURRED, FOR ERRORS IN OTHER 4K BLOCKS THE CORRECT RESULT CORRELATES TO THE ADDRESS WHERE THE ERROR OCCURRED. FOR EXAMPLE

RT72 PC=XXXXXX ERR S/B: 000001 WAS XXXXXX.
 INDICATES THAT THE DM11 FAILED TO TRANSMIT AND RECEIVE CORRECT DATA WHEN TRANSMITTING FROM LOCATION 20000.
 THE TEST IS ABORTED BEFORE TRANSMITTING IF THE CORE LOCATION IS NON-EXISTANT.

RT73 THIS ROUTINE TESTS THAT THE TRANSMITTER CAN TRANSMIT 100 CHARACTERS ON EACH LINE. THE ROUTINE TESTS THAT EXACTLY 100 CHARACTERS HAVE BEEN TRANSMITTED BEFORE READY (CSR BIT15) SETS AND THE BAR BIT CLEARS. THE ERROR TIMEOUT GIVES THE NUMBER OF CHARACTERS RECEIVED AT THE TIME OF AN ERROR, AND THE FAILING LINE NUMBER (X2).

RT74 THIS ROUTINE TESTS THAT THE DM11 WILL STORE DATA SEQUENTIALLY IN THE TUMBLE TABLE AND ALSO THAT THE POINTER RETURNS TO THE TOP OF THE TABLE WHEN 64 CHARACTERS HAVE BEEN RECEIVED.

RT75-114 THESE ROUTINES CHECK THAT A BREAK CAN BE TRANSMITTED AND RECEIVED ON ALL ALINES

R115-R134 THESE ROUTINES INDIVIDUALLY TRANSMIT, RECEIVE AND CHECK DATA PLUS PARITY ON EACH OF THE 16 DM11 LINES ONLY DATA AND PARITY ERRORS ARE REPORTED.

RT131 THIS ROUTINE SIMULTANEOUSLY TRANSMITS AND RECEIVES A CHARACTER (ALL 1'S) ON THE 16 DM11 LINES THE FOLLOWING TESTS ARE PERFORMED:

- A: THERE ARE 16 DATA ENTRIES (1 PER LINE)
- B: THERE ISN'T A 17TH ENTRY
- C: DATA IS CORRECT
- D: ONE ENTRY FOR EACH LINE

RT136 THIS ROUTINE TRANSMITS A BREAK ON EACH LINE TESTS PERFORMED ARE THE SAME AS IN RT135.

RT137-RT144 THESE ROUTINES TRANSMIT 64 CHARACTERS ON EACH LINE WITH A DELAY BEFORE BEGINNING TRANSMISSION ON THE NEXT SUCCESSIVE LINE. THE DELAY BEFORE TRANSMITTING ON THE NEXT LINE IS HALVED BY SUCCESSIVE TESTS. NO DATA CHECKING IS PERFORMED BY THESE TESTS. TESTED ARE THAT OVER RUN (CSR BIT13) AND NEX (CSR BIT14) ARE NOT SET DURING TRANSMISSION/RECEPTION.

RT145 THIS ROUTINE TESTS PROPER OPERATION OF THE HALF DUPLEX BIT (CSR BIT1)

RT146 THIS ROUTINE TESTS THAT THE DM11 COMES TO AN 'ORDERLY HALT' WHEN THE RESET INSTRUCTION IS GIVEN. 'ORDERLY HALT' IS DEFINED AS CSR, BAR, AND BKCS CLEAR IMMEDIATELY AFTER THE RESET INSTRUCTION AND STAY CLEARED.

- 5. 2 PRG1- TRANSMITTER SCOPE LOOP
PROGRAM 1 ALLOWS THE USER TO SCOPE THE DM11 TRANSMITTER FUNCTIONS WITH THE DM11 CONTINUOUSLY RUNNING UNDER USER SUPPLIED PARAMETERS
- 5. 3 PRG2- TRANSMITTER/RECEIVER SCOPE LOOP
PROGRAM 2 ALLOWS THE USER TO SCOPE THE DM11 RECEIVER FUNCTIONS WITH THE DM11 CONTINUOUSLY RUNNING UNDER USER SUPPLIED PARAMETERS

6.0 PROGRAM 1 AND PROGRAM 2 PARAMETERS
 WHEN PROGRAM 1 OR PROGRAM 2 ARE SELECTED ADDITIONAL PARAMETERS WILL
 BE REQUESTED BY EACH PROGRAM AS SHOWN BELOW

A TYPE LINES TO BE TESTED
 EXAMPLES:
 TYPE TO SELECT LINE(S)
 1 0
 3 1,0
 10 3
 17 3,2,1,0
 50 5,3
 3101 10,7,6,0
 17770 14,13,12,11,10,7,6,5,4,3
 177777 ALL

NOTE, LINE NUMBERS ARE GIVEN IN OCTAL.

B. HOW MANY CHARACTERS
 TYPE THE NUMBER OF CHARACTERS YOU WISH TO TRANSMIT NOTE,
 THE NUMBER OF CHARACTERS MUST BE LESS THAN 200, AND IS TAKEN
 IN OCTAL.

C. PUT CHARACTER IN SR (0-7); DELAY IN SR (8-15)
 SELF-EXPLANATORY. NOTE, THE DELAY REFERS TO A DELAY AFTER
 ALL THE CHARACTERS HAVE BEEN TRANSMITTED AND BEFORE A NEW
 TRANSMISSION PERIOD BEGINS.

7.0 PROGRAM LIMITATIONS
 BECAUSE THE DM11 DIAGNOSTICS ARE INSENSITIVE TO 'REAL' ELAPSED TIME
 THE DIAGNOSTIC DOES NOT 'KNOW' IF THE DM11 IS OPERATING AT THE COR-
 RECT FREQUENCY OR THAT THE STOP CODE SELECTION LOGIC IS CORRECT,
 THESE SHOULD BE CHECKED WITH A SCOPE.

8.0 PROGRAM NOTES
 IF THE POWER FAILS THE PROGRAM TYPES AN ERROR MESSAGE INDICATING THE
 ROUTINE THAT WAS RUNNING (PROG NO ONLY) AND RESTARTS THE PROGRAM

***** IMPORTANT NOTE *****

POWER FAIL TEST

A TEST OF THE POWER FAIL LOGIC SHOULD BE PERFORMED ON EACH UNIT
 SELECT & RUN ROUTINE 144 (L.A. = 210 SR = 5144 *PRESS START). TURN
 THE POWER OFF THEN ON. THE PROGRAM WILL TYPE OUT THE POWER FAIL
 ERROR

R144 PC=003622

AND CONTINUE RUNNING ROUTINE 144. LOWER SR 9 AND WAIT FOR END OF
 TEST MESSAGE. 'TEST OZDMA COMPLETE'.

NOTE: IF THE POWER IS TURNED OFF DURING A RESET INSTRUCTION THE
 PROGRAM WILL HALT. PRESS CONTINUE AND REPEAT THE TEST

IF THE PROGRAM HANGS THE BUS EXAMINE THE CONTENTS OF RTNNO. THE
 CONTENTS OF RTNNO IS THE ROUTINE NUMBER THAT WAS RUNNING AT THE TIME
 OF THE FAILURE.

%

. TITLE CZDMAD0 DM11A LGC TST
LIST ME,BIN,SEQ
. ENABLE ABS,AMA

413		;CZDMAD0 DM11A LGC TST
414		;PRG0- INPUT-OUTPUT LOGIC TESTS
415		;PRG1- TRANSMITTER SCOPE LOOP
416		;PRG2- TRANSMIT/RECEIVE SCOPE LOOP
417		;STANDARD SR SWITCH OPTIONS (SWITCH SET TO A 1)
418		;SR15- HALT ON ERROR
419		;SR14- SCOPE.
420		;SR13- INHIBIT PRINTOUT
421		;SR12- INHIBIT TRACE (NOT USED)
422		;SR11- INHIBIT ITERATION
423		;SR10- LOOP PROGRAM (NOT USED)
424		;SR9- LOOP ROUTINE.
425		;SR8- RING BELL ON AN ERROR
426		;SR6 THROUGH SR0 - NUMBER OF ROUTINE TO BE LOOPED
427		
428		;EQUATE STATEMENTS
429		CC=177776
430		PSW=177776
431		ERRVEC=4
432	177776	
433	177776	
434	000004	. ADDRESS OF ERROR TRAP VECTOR
435	000240	
436	000000	
437	100000	
438	100000	
439	040000	
440	020000	
441	010000	
442	004000	
443	002000	
444	001000	
445	000400	
446	000200	
447	000100	
448	000040	
449	000020	
450	000010	
451	000004	
452	000002	
453	000001	
454	100000	
455	040000	
456	020000	
457	010000	
458	004000	
459	002000	
460	001000	
461	000400	
462	000200	
463	000100	
464	000040	
465	000020	

466	000010	BIT3=10
467	000004	BIT2=4
468	000002	BIT1=2
469	000001	BIT0=1
470	005726	POPSP=5726
471	022626	POPSP2=022626
472	000340	PRTY7=340
473	000300	PRTY6=300
474	000240	PRTY5=240
475	000200	PRTY4=200
476	000140	PRTY3=140
477	000100	PRTY2=100
478	000040	PRTY1=40
479	000000	PRTY0=0
480		; LINE NUMBERS
481	000000	LINE0=0
482	000002	LINE1=2
483	000004	LINE2=4
484	000006	LINE3=6
485	000010	LINE4=10
486	000012	LINE5=12
487	000014	LINE6=14
488	000016	LINE7=16
489	000020	LINE10=20
490	000022	LINE11=22
491	000024	LINE12=24
492	000026	LINE13=26
493	000030	LINE14=30
494	000032	LINE15=32
495	000034	LINE16=34
496	000036	LINE17=36
497	000000	RO=X0
498	000001	R1=X1
499	000002	R2=X2
500	000003	R3=X3
501	000004	R4=X4
502	000005	R5=X5
503	000006	SP=X6
504	000007	PC=X7
505		; ENT CALLS
506	104000	TYPE=ENT+0
507	104001	ERROR=ENT+1
508	104002	DATCHK=ENT+2
509	104003	CHALT=ENT+3
510	104004	EHALT=ENT+4
511	104005	SRESET=ENT+5
512	104006	SCOPE=ENT+6
513	104007	SAVREG=ENT+7
514	104010	RSTREG=ENT+10
515	104011	ERROR1=ENT+11
516	104012	INITIALIZE=ENT+12
517	104013	SUSMR=ENT+13
518	104014	KBDIN=ENT+14
519	104015	CNTLU=ENT+15
520	104400	DELAY=TRAP+0
521	000007	BELL=007

; POP THE STACK SAME AS TST (6)+
; POP STACK TWICE SAME AS CMP (6)+, (6)+
; PRIORITY LEVEL DEFINITIONS

522	177777	RTLAST=-1	
523			
524	000000	Y=0	
525	177777	X=-1	
526	000000	A=0	
527	000000	=0	
528	000000	.+2	; UNASSIGNED TRAP
529	000002	HALT	
530	000004	.+2	; SP OVERFLOW, BUS ERROR TRAP
531	000006	HALT	
532	000010	.+2	; RESERVED INSTRUCTION TRAP
533	000012	HALT	
534	000014	.+2	; TRACE TRAP
535	000016	HALT	
536	000020	.+2	; TRAP TO CALL IOX
537	000022	HALT	
538	000024	.+2	; POWER FAIL TRAP
539	000026	HALT	
540	000030	ENTINT	; ENT TRAP
541	000032	PRTY7	
542	000034	DLY	
543	000036	PRTY7	; TRAP TRAP. SIMILAR TO ENT
544	000040	.+2	
545	000042	HALT	; TRAPPED TO PREVIOUS ADDRESS
546	000044	.+2	
547	000046	HALT	; TRAPPED TO PREVIOUS ADDRESS
548	000050	.+2	
549	000052	HALT	; TRAPPED TO PREVIOUS ADDRESS
550	000054	.+2	
551	000056	HALT	; TRAPPED TO PREVIOUS ADDRESS
552	000060	.+2	
553	000062	HALT	; TRAPPED TO PREVIOUS ADDRESS
554	000064	.+2	
555	000066	HALT	; TRAPPED TO PREVIOUS ADDRESS.
556	000070	.+2	
557	000072	HALT	; TRAPPED TO PREVIOUS ADDRESS.
558	000074	.+2	
559	000076	HALT	; TRAPPED TO PREVIOUS ADDRESS.
560	000100	.+2	
561	000102	HALT	; TRAPPED TO PREVIOUS ADDRESS.
562	000104	.+2	
563	000106	HALT	; TRAPPED TO PREVIOUS ADDRESS.
564	000110	.+2	
565	000112	HALT	; TRAPPED TO PREVIOUS ADDRESS.
566	000114	.+2	
567	000116	HALT	; TRAPPED TO PREVIOUS ADDRESS.
568	000120	.+2	
569	000122	HALT	; TRAPPED TO PREVIOUS ADDRESS.
570	000124	.+2	
571	000126	HALT	; TRAPPED TO PREVIOUS ADDRESS.
572	000130	.+2	
573	000132	HALT	; TRAPPED TO PREVIOUS ADDRESS.
574	000134	.+2	
575	000136	HALT	; TRAPPED TO PREVIOUS ADDRESS.
576	000140	.+2	
577	000142	HALT	; TRAPPED TO PREVIOUS ADDRESS.

MACHER:

578	000144	000146	.+2	
579	000146	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS
580	000150	000152	.+2	
581	000152	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS
582	000154	000156	.+2	
583	000156	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS
584	000160	000162	.+2	
585	000162	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS
586	000164	000166	.+2	
587	000166	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS
588	000170	000172	.+2	
589	000172	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS
590	000174	000176	.+2	
591	000176	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
592	000200	000202	.+2	
593	000202	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS
594	000204	000206	.+2	
595	000206	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS
596	000210	000212	.+2	
597	000212	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS
598	000214	000216	.+2	
599	000216	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS
600	000220	000222	.+2	
601	000222	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS
602	000224	000226	.+2	
603	000226	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS
604	000230	000232	.+2	
605	000232	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS
606	000234	000236	.+2	
607	000236	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
608	000240	000242	.+2	
609	000242	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
610	000244	000246	.+2	
611	000246	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
612	000250	000252	.+2	
613	000252	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
614	000254	000256	.+2	
615	000256	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
616	000260	000262	.+2	
617	000262	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
618	000264	000266	.+2	
619	000266	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
620	000270	000272	.+2	
621	000272	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
622	000274	000276	.+2	
623	000276	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
624	000300	000302	.+2	
625	000302	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
626	000304	000306	.+2	
627	000306	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
628	000310	000312	.+2	
629	000312	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
630	000314	000316	.+2	
631	000316	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
632	000320	000322	.+2	
633	000322	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.

634	000324	000326		
635	000326	000000	+2	
636	000330	000332	HALT	, TRAPPED TO PREVIOUS ADDRESS
637	000332	000000	+2	
638	000334	000336	HALT	, TRAPPED TO PREVIOUS ADDRESS
639	000336	000000	+2	
640	000340	000342	HALT	, TRAPPED TO PREVIOUS ADDRESS
641	000342	000000	+2	
642	000344	000346	HALT	, TRAPPED TO PREVIOUS ADDRESS
643	000346	000000	+2	
644	000350	000352	HALT	, TRAPPED TO PREVIOUS ADDRESS
645	000352	000000	+2	
646	000354	000356	HALT	, TRAPPED TO PREVIOUS ADDRESS
647	000356	000000	+2	
648	000360	000362	HALT	, TRAPPED TO PREVIOUS ADDRESS
649	000362	000300	+2	
650	000364	000366	HALT	, TRAPPED TO PREVIOUS ADDRESS
651	000366	000000	+2	
652	000370	000372	HALT	, TRAPPED TO PREVIOUS ADDRESS
653	000372	000000	+2	
654	000374	000376	HALT	, TRAPPED TO PREVIOUS ADDRESS
655	000376	000000	+2	
656			HALT	, TRAPPED TO PREVIOUS ADDRESS

657					
658		000046		=46	
659	000046	003036		SENDAD	
660		000052		=52	
661	000052	060000		60000	
662					
663					
664		000174		=174	
665	000174	000000	DISPREG: 0		
666	000176	000000	SWREG: 0		
667					
668		000200		=200	
669	000200	000137	002360	JMP	@#START
670	000204	000137	002426	JMP	@#RSTAT1
671					
672	000210	000137	002514	JMP	@#RSTAT2
673					
674					
675		001200		=1200	
676	001200	000000	SPBOT: 0		..++D
677	001202	177570	SWR 177570		
678	001204	175570	DISPLAY: 175570		
679		001400		=1400	
680	001400	000000	CAT: OPEN		..++D
681		001440		=CAT+32	
682	001440	000000	WCT: OPEN		
683		001500		=WCT+32	
684	001500	000000	BAT: OPEN		
685		001540		=BAT+32	
686	001540	000000	VAC: OPEN		
687	001542	175000	CSR: 175000		
688	001544	175002	BAR: 175002		
689	001546	175004	BKCSR: 175004		
690	001550	175006	BASREG: 175006		
691	001552	000000	CLKINT: OPEN		
692	001554	000240	CLKLVL: PRYS		
693	001556	000000	XMTINT: OPEN		
694	001560	000240	XMTLVL: PRYS		
695	001562	000000	TTDAT: OPEN		
696	001564	000000	LINBIT: OPEN		
697	001566	000000	RCVDAT: OPEN		
698	001570	000000	XMTDAT: OPEN		
699	001572	000000	CARMSK: OPEN		
700		001600		=VAC+32	
701	001600	000000	TUMTAB: OPEN		
702		002000		=TUMTAB+128	
703	002000	000000	KSTART: OPEN		
704	002002	000000	CURTST: OPEN		
705	002004	000000	RTNNO: OPEN		
706	002006	000000	NXTST: OPEN		
707	002010	000000	ICTR: OPEN		
708	002012	000000	SCOPTR: OPEN		
709	002014	000000	PRGLIM: OPEN		
710	002016	007076	PRGTAB: PRGO		
711	002020	016514		PRG1	
712	002022	016574		PRG2	

GO TO START OF DIAGNOSTIC
 GO GET PROGRAM # & RESTART PROGRAM
 USING PREVIOUS DM11 PARAMETERS
 RESTART PREVIOUS PROGRAM USING
 PREVIOUS DM11 PARAMETERS

STARTING ADDRESS OF
 CURRENT ADDRESS TABLE
 STARTING ADDRESS OF
 WORD COUNT TABLE
 STARTING ADDRESS OF
 BIT ASSEMBLY TABLE
 32 SPARE WORDS
 ADDRESS OF CLOCK STATUS REGISTER
 ADDRESS OF BUFFER ACTIVE REGISTER
 ADDRESS OF BREAK STATUS REGISTER
 ADDRESS OF BASE REGISTER
 DM11 VECTOR ADDRESS (RECEIVER)
 PRIORITY LEVEL
 DM11 VECTOR ADDRESS (TRANSMITTER)
 TRANSMITTER PRIORITY LEVEL
 TUMBLE TABLE DATA
 LINE BIT (FOR BAR)

STARTING ADDRESS OF
 TUMBLE TABLE
 CURRENT PROGRAM START ADDRESS
 CONTAINS ADDR OF CURRENT TEST
 CONTAINS CURRENT TEST #
 CONTAINS ADDR OF NEXT TEST
 CONTAINS CURRENT ITERATION COUNT
 CONTAINS CURRENT SCOPE POINTER

PRGO START ADDRESS
 PRG1 START ADDRESS
 PRG2 START ADDRESS

713 002024 007120
 714 002026 016524
 715 002030 016604
 716 002032 003322
 717 002034 002146
 718 002036 002126
 719 002040 000000
 720 002042 000000
 721 002044 003232
 722 002046 002700
 723 002050 003132
 724 002052 003172
 725 002054 002164
 726 002056 003434
 727 002060 017412
 728 002062 017472
 729 002064 017546
 730 002066 177560
 731 002070 177562
 732 002072 177564
 733 002074 177566
 734 002076 000000
 735 002100 000000
 736 002102 000000
 737 002104 000000
 738 002106 175001
 739 002110 175003
 740 002112 175005
 741 002114 175007
 742 002116 000000

RSTART: PRGOR
 PRG1R
 PRG2R
 EMTTAR: TYP
 ERR
 DTCHK
 0
 0
 SRSETT
 ESCOPE
 SAVRG
 RSTRG
 ERR1
 INIT
 SUSWR
 KBDINTT
 CNTLUU
 TKCSR: 177560
 TKDBR: 177562
 TPCSR: 177564
 TPDBR: 177566
 COUNT: OPEN
 PCADD: OPEN
 APCADD: OPEN
 PRVCNT: OPEN
 CSR: 175001
 BAR: 175003
 BKCSR: 175005
 BASREG: 175007
 PASS: OPEN

; PRGO RESTART ADDRESS
 ; PRG1 " "
 ; PRG2 " "
 ; POINTER TO TYPEOUT ROUTINE
 ; POINTER TO ERROR ROUTINE
 ; POINTER TO DATA COMPARISON ROUTINE
 ; POINTER TO RESET ROUTINE
 ; POINTER TO SCOPE ROUTINE
 ; POINTER TO SAVE REGISTERS ROUTINE
 ; POINTER TO RESTORE REGISTERS ROUTINE
 ; POINTER TO ERROR1 ROUTINE
 ; POINTER TO INITIALIZE ROUTINE

```
743
744
745
746 002120 104000
747 002122 020017
748 002124 000207
749
750
751 002126 123737 001566 001570
752 002134 001403
753 002136 004737 002332
754 002142 104011
755 002144 000002
756
757
758
759
760 002146 012737 000402 002246
761 002154 013737 002100 002102
762 002162 000410
763 002164 012737 000240 002246
764 002172 013737 002100 002102
765 002200 004737 002332
766 002204 104014
767 002206 032777 020000 176766
768 002214 001017
769 002216 004537 006154
770 002222 002102
771 002224 020224
772 002226 000006
773 002230 004537 006154
774 002234 002004
775 002236 020216
776 002240 000003
777 002242 104000
778 002244 020213
779 002246 000000
780 002250 104000
781 002252 017754
782 002254 032777 000400 176720
783 002256 001411
784 002258 105777 177602
785 002270 100375
786 002272 012777 000007 177574
787 002300 105777 177566
788 002304 100375
789 002306 023737 000042 000046
790 002314 001403
791 002316 005777 176660
792 002322 100001
793 002324 000000
794 002326 104014
795 002330 000002
796
797
798
```

ROUTINE TO TYPE OUT INCORRECT ROUTINE SELECTED
INCRTN. TYPE
M1
RTS %7
TYPE INCORRECT ROUTINE SELECTED
EXIT.

DATA COMPARISON ROUTINE
DTCHK: CMPB RCVDAT,XMTDAT ;COMPARE RECEIVED & TRANSMITTED DATA
BEQ 15 ;CHARS. BRANCH IF SAME
JSR 7,CNV DAT ;CONVERT RCVDAT & XMTDAT TO ASCII
ERROR1
15: RTI ;EXIT.

ERROR ROUTINE WHENEVER THE PROGRAM DETECTS AN ERROR THE ERROR
AND ERROR1 ENT INSTRUCTIONS ENTER HERE. ERROR AT ERR: AND
ERROR1 AT ERR1:

ERR: MOV #402,ERRB ;MOV BR +6 TO ERRB
MOV PCADD,APCADD ;GET PC WHERE ERROR OCCURRED
BR ERRA
ERR1: MOV #240,ERRB ;MOVE NOP TO ERRB
MOV PCADD,APCADD ;GET PC WHERE ERROR OCCURRED
JSR 7,CNV DAT ;CONVERT RCVDAT & XMT DAT TO ASCII
ERRA: KBDIN ;GO CHECK FOR G
BIT #BIT13,JSR ;ERROR PRINTOUT DESIRED
BNE ERRC ;BRANCH IF NO PRINTOUT
JSR 5,ORCNV ;CONVERT
APCADD ;DATA
APC ;TO
6 ;ASCII
JSR 5,ORCNV ;FOR
RTNNO ;PRINTOUT
ATNMB
3
TYPE
EMO
ERRB: OPEN ;TYPE ERROR
TYPE ;MESSAGE
ERDAT ;NOP IF ERROR1, BR +6 IF ERROR
BIT #BIT8,JSR ;TYPE ANOTHER MESSAGE
BEQ ERRD ;IF ERROR 1
TSTB #TPCSR ;RING BELL ON ERROR?
BPL -4 ;BRANCH IF NO BELL ON ERROR
MOV #BELL,ATPOBR ;TELEPRINTER
TSTB #TPCSR ;READY?
BPL -4 ;RING THE BELL
CMP #42,#46 ;WAIT FOR THE BELL TO RING
BEQ ERRLT ;ACT11?
TST JSR ;HALT ON ERROR
BPL ERREX ;GO TO EXIT IF NO HALT ON ERROR
ERRHLT: HALT
ERREX: KBDIN ;CHECK FOR G
RTI ;RETURN

SUBROUTINE TO CONVERT RCVDAT AND XMTDAT TO ASCII AND PLACE

799								
800	002332	004537	006154					
801	002336	001570						
802	002340	017766						
803	002342	000006						
804	002344	004537	006154					
805	002350	001566						
806	002352	020002						
807	002354	000006						
808	002356	000207						
809								
810								

IN MESSAGE
CNVDAT. JSR 5.0ACNV
XMTDAT
AASB
6
JSR 5.0ACNV
RCVDAT
AWAS
6
RTS 7 .EXIT

```

811
812
813
814
815
816
817
818 002360 012706 001200
819 002364 104013
820 002366 012737 002410 000004
821 002374 005737 172300
822 002400 012737 012716 012330
823
824 002406 000402
825 002410 012706 001200
826
827 002414 012737 000006 000004
828
829
830
831 002422 004737 003464
832 002426 012706 001200
833 002432 104012
834 002434 023737 000042 000046
835 002442 001405
836 002444 104000
837 002446 020011
838 002450 004537 004416
839 002454 000000
840 002456 043737 002014 002454
841 002464 006337 002454
842 002470 012737 004576 000024
843 002476 012737 000340 000026
844 002504 013700 002454
845 002510 000170 002016
846 002514 012737 004576 000024
847 002522 012737 000340 000026
848 002530 012706 001200
849 002534 104012
850 002536 013700 002454
851 002542 000170 002024
852 002546 022737 000176 001202
853 002554 001410
854 002556 023737 000042 000046
855 002564 001405
856 002566 104000
857 002570 020030
858 002572 000000
859 002574 000401
860 002576 104015
861 002600 013737 002000 002006
862 002606 012737 000006 000004
863 002614 104012
864 002616 004737 003046
865 002622 032777 001000 176352
866 002630 001003

; THE FIRST PART OF THE START ROUTINE CONTAINS A SHORT
; ROUTINE TO CHECK FOR MEMORY MANAGEMENT. ALTHOUGH THIS
; DIAGNOSTIC DOES NOT USE MEMORY MANAGEMENT, ITS PRESENCE
; INDICATES THAT OVER 28K OF MEMORY MAY BE PRESENT IN WHICH
; CASE TESTS RT66 AND RT67 MAY FAIL. IF MEM. MAN. IS
; PRESENT THESE TESTS ARE SKIPPED BY THE PROGRAM.

START: MOV #SPBOT,%6
      SUSWR
      MOV #15,%ERRVEC ; SET IF SWITCH-LESS PROCESSOR
      YST #172300 ; SET UP FOR ERROR TRAP
      MOV #RT70,%RT65+2 ; TEST FOR KT11
      ; KT11 PRESENT, SET UP TO
      ; SKIP RT66 AND RT67
      BR +6
15: MOV #SPBOT,%6 ; TRAP OCCURRED, NO KT11 PRESENT,
      MOV #ERRVEC+2,%ERRVEC ; RESET STACK
      ; RESET ERROR TRAP

; OUT INTERRUPTS (SET PRIORITY LEVEL 7)
; GET DM11 PARAMETERS
RSTAT1: JSR 7,%DMPPAR
        MOV #SPBOT,%6
        INITIALIZE
        CMP #42,%46 ; ACT11?
        BEQ PRGNUM+2
        TYPE
        MO
        JSR 5,RECD ; GET PRGNUM AND PUT IT
        PRGNUM: 0 ; HERE
        BIC PRGLIN,PRGNUM ; MASK OFF UNUSED BITS
        ASL PRGNUM ; SHIFT PROGRAM #
        MOV #PFAIL,%24
        MOV #PRTY7,%26
        MOV PRGNUM,%X0 ; GET PROGRAM #
        JMP #PRGTAB(0) ; GO START PROGRAM
RSTAT2: MOV #PFAIL,%24
        MOV #PRTY7,%26
        MOV #SPBOT,%6
        INITIALIZE
        MOV PRGNUM,%X0 ; GET PROGRAM #
        JMP #RSTART(0) ; GO RESTART PROGRAM
SRSET: CMP #SWREG,%SWR
        BEQ 15
        CMP #42,%46 ; ACT11?
        BEQ GETRDY
        TYPE ; TYPE MESSAGE TO REQUEST SWITCH
        NJ ; REGISTER SETTINGS
        HALT ; WAIT FOR OPERATOR TO SET SWITCHES
        BR GETRDY
15: CNTLU
GETRDY: MOV KSTART,NXTST ; GO GET SWREG SETTINGS
GTRDYX: MOV #6,%ERRVEC ; ADDR OF 1ST ROUTINE TO NXTST
        INITIALIZE ; RESET ERROR TRAP VECTOR
GTRDYA: JSR X7,FORWD ; ROLL FORWARD TO "NEXT" ROUTINE.
        BIT #BIT9,%SWR ; CHECK SELECT ROUTINE SWITCH
        BNE GTRDYC ; BRANCH IF SELECT ROUTINE SWITCH IS SET

```

867	002632	000177	177144		JMP	SCURTST			
868	002636	000457			BR	SCOPE			
869	002640	017700	176336		MOV	ASUR, X0			
870	002644	042700	177600		BIC	R177600, X0			
871	002650	123700	002004		CMPB	RTNNO, X0			
872	002654	001002			BNE	GTRDYD			
873	002656	000177	177120		JMP	SCURTST			
874	002662	022737	177777	002006	GTRDYD	CMP	R-1, NXTST		
875	002670	001352			BNE	GTRDYA			
876	002672	004737	002120		JSR	X7, INCRTN			
877	002676	000740			BR	GETRDY			
878									
879									
880	002700	000240							
881	002702	104014			ESCOPE	NOP			
882	002704	005077	176634		KBDIN				
883	002710	005077	176626		CLR	ASAR			
884	002714	005077	176626		CLR	ACSR			
885	002720	104012			CLR	AKCSR			
886	002722	013716	002012		INITIALIZE				
887	002726	032777	040000	176246	MOV	SCOptr, (SP)			
888	002734	001402			BIT	BIT14, ASUR			
889	002736	000176	000000		BEQ	SCOPEB			
890	002742	032777	004000	176232	SCOPEA:	JMP	AS(P)		
891	002750	001012			SCOPEB:	BIT	BIT11, ASUR		
892	002752	023737	000042	000046	BNE	SCOPE			
893	002760	001003			CMP	AS42, AS46			
894	002762	005737	002116		BNE	IS			
895	002766	001403			TST	ASPASS			
896	002770	005337	002010		BEQ	SCOPE			
897	002774	001360			15:	DEC	ICTR		
898	002776	032777	001000	176176	BNE	SCOPEA			
899	003004	001275			SCOPE:	BIT	BIT9, ASUR		
900	003006	022737	177777	002006	BNE	GETRDY			
901	003014	001274			CMP	R-1, NXTST			
902	003016	005237	002116		BNE	GTRDYX			
903	003022	104000			INC	ASPASS			
904	003024	020022			TYPE				
905	003026	013700	000042		M2				
906	003032	001662			MOV	AS42, X0			
907	003034	000005			BEQ	GETRDY			
908	003036	004710			RESET				
909	003040	000240			SENDAD:	JSR	7, (0)		
910	003042	000240			NOP				
911	003044	000240			NOP				
912	003046	013706	002006		NOP				
913	003052	012537	002004		FORMD:	MOV	NXTST, X5		
914	003056	012537	002006			MOV	(5)+, RTNNO		
915	003062	012537	002010			MOV	(5)+, NXTST		
916	003066	012537	002012			MOV	(5)+, ICTR		
917	003072	010537	002002			MOV	(5)+, SCOptr		
918	003076	000207			FORMDA:	MOV	X5, CURTST		
919						RTS	X7		
920									
921									
922	003100	011646							

; GO RUN CURRENT ROUTINE
 ; NO GO. MANUAL RTN BYPASSED
 ; (SR) TO R0
 ; MASK UNDESIED BITS
 ; COMPARE RTNNO TO (R0)
 ; BRANCH IF ROUTINE NOT FOUND YET
 ; GO RUN ROUTINE.
 ; NO. CHECK FOR LAST ROUTINE
 ; BRANCH IF NOT LAST ROUTINE
 ; YES. INCORRECT ROUTINE SELECTED
 ; START OVER

; SCOPE SERVICE ROUTINE
 ESCAPE.

; CHECK FOR G
 ; CLEAR ALL DM11 REGISTERS
 ; AND SET BASE REGISTER
 ; AT THE STARTING ADDRESS

; CHECK FOR SCOPE OPTION
 ; BRANCH IF SCOPE SW NOT SET
 ; RETURN TO ROUTINE
 ; TEST INHIBIT ITERATION SWITCH
 ; BRANCH IF INHIBIT ITERATION SW SET
 ; ACT11?
 ; BR IF NO
 ; 1ST PASS?
 ; BR IF YES
 ; DECREMENT ITERATION COUNT
 ; BRANCH IF COUNT NOT 0
 ; CHECK SELECT ROUTINE SWITCH
 ; BRANCH IF SELECT RTN SW SET
 ; LAST TEST?
 ; BRANCH IF NOT LAST TEST

; TYPE
 ; 'PRGEND'
 ; CHECK XXDP/ACT11 MONITOR HOOK

; RETURN TO XXDP/ACT11 MONITOR

; ADDR OF NEXT ROUTINE TO R5
 ; GET NEXT ROUTINE NUMBER.
 ; GET ADDR OF NEXT "NEXT" ROUTINE
 ; GET ITERATION COUNT.
 ; GET SCOPE LOOP ENTRY POINTER.
 ; ADDR OF NON CURRENT TEST TO CURTST
 ; EXIT FORMD SUBROUTINE.

; EMT TRAP INTERPRETER
 EMTINT. MOV (6), -(6)

; GET PC OF NEXT INSTRUCTION

923	003102	162716	000002			
924	003106	011637	002100	SUB	#2, (6)	, FORM PC OF EMT INSTRUCTION
925	003112	017616	000000	MOV	(6), PCADD	, GET PC OF EMT INSTRUCTION
926	003116	105066	000001	MOV	2(6), (6)	, GET EMT INSTRUCTION
927	003122	006316		CLRB	1(6)	, CLEAR MSH OF EMT INSTRUCTION
928	003124	062716	002032	ASL	(6)	, SHIFT EMT IDENTIFIER
929	003130	013607		ADD	#EMTTAB, (6)	
930				MOV	2(6)+, X7	, GO TO PROPER EMT
931						
932	003132	012637	003166			
933	003136	012637	003170	, SAVE REGS 0 TO 4 SUBROUTINE		
934	003142	010446		SAVRG: MOV	(6)+, SVRPC	, SAVE PC AND PSW
935	003144	010346		MOV	(6)+, SVRPSW	
936	003146	010246		MOV	X4, -(6)	, SAVE REGS 0 - 4
937	003150	010146		MOV	X3, -(6)	, IN STACK
938	003152	010046		MOV	X2, -(6)	
939	003154	013746	003170	MOV	X1, -(6)	
940	003160	013746	003166	MOV	X0, -(6)	
941	003164	000002		MOV	SVRPSW, -(6)	, RESTORE PC AND PSW
942	003166	000000		MOV	SVRPC, -(6)	
943	003170	000000		RTI		, EXIT
944				SVRPC: OPEN		
945				SVRPSW: OPEN		
946	003172	012637	003226			
947	003176	012637	003230	, RESTORE REGS 0 TO 4 SUBROUTINE.		
948	003202	012600		RSTRG: MOV	(6)+, RSTPC	, SAVE PC AND PSW
949	003204	012601		MOV	(6)+, RSTPSW	
950	003206	012602		MOV	(6)+, X0	, RESTORE REGS 0 - 4
951	003210	012603		MOV	(6)+, X1	, FROM STACK
952	003212	012604		MOV	(6)+, X2	
953				MOV	(6)+, X3	
954	003214	013746	003230	MOV	(6)+, X4	
955	003220	013746	003226	MOV	RSTPSW, -(6)	, RESTORE PC AND PSW
956	003224	000002		MOV	RSTPC, -(6)	
957	003226	000000		RTI		, EXIT
958	003230	000000		RSTPC: OPEN		
959				RSTPSW: OPEN		
960						
961	003232	012700	052525	, ROUTINE TO ISSUE RESET.		
962	003236	005100		SRSETT: MOV	#52525, X0	, DATA TO R0.
963	003240	010037	003234	COM	X0	, COMPLEMENT (R0).
964	003244	000005		MOV	X0, SRSETT+2	, (R0) TO SRSETT+2.
965	003246	000002		RESET		, ISSUE RESET. (R0) IS
966				RTI		, DISPLAYED. EXIT.
967						
968	003250	013700	003316	, RANDOM NUMBER GENERATOR. ROUTINE EXITS WITH NUMBER IN REGISTER 0.		
969	003254	006100		RNGEN: MOV	RP1, X0	
970	003256	006100		RQ'	X0	
971	003260	063700	003320	ROL	X0	
972	003264	010037	003316	ADD	RP2, X0	
973	003270	006100		MOV	X0, RP1	
974	003272	006100		ROL	X0	
975	003274	063700	003320	ROL	X0	
976	003300	006100		ADD	RP2, X0	
977	003302	006100		ROL	X0	
978	003304	010037	003320	ROL	X0	
				MOV	X0, RP2	

979	003310	013700	003316		MOV	RP1,%0	
980	003314	000207			RTS	%7	
981	003316	001233					.EXIT NUMBER IN R0
982	003320	007622		RP1	1233		
983				RP2	7622		
984	003322	011600			; SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER		
985	003324	062716	000002	TYP	MOV	%X6,%0	; GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
986	003330	011000			ADD	%2,%X6	; SET UP EXIT.
987	003332	112037	003432		MOV	%X0,%0	; ADDRESS OF MESSAGE TO R0.
988	003336	122737	000100	003432	TYP A:	MOVH	(0)+, TYPDAT
989	003344	001001			CMPB	%100, TYPDAT	; CHECK FOR "a" CHARACTER
990	003346	000002			BNE	TYP C	; BRANCH IF NOT "a"
991	003350	122737	000045	003432	RTI		; TERMINATOR CHAR. DONE EXIT
992	003356	001412			CMPB	%45, TYPDAT	; CHECK FOR "x"
993	003360	004737	003366		BEQ	TYP F	; BRANCH IF "x"
994	003364	000762			JSR	%7, TYPD	; TYPE CHAR IN TYPDAT
995	003366	113777	003432	176500	BR	TYP A	
996	003374	105777	176472		MOVB	TYPDAT, %TPD8R	; OUTPUT CHARACTER TO PRINTER
997	003400	100375			TSTB	%TPCSR	; WAIT FOR DONE FLAG.
998	003402	000207			BPL	-4	
999	003404	112737	000015	003432	RTS	%7	; EXIT
1000	003412	004737	003366		MOVB	%15, TYPDAT	; MOVE CARRIAGE RETURN CODE TO TYPDAT
1001	003416	112737	000012	003432	JSR	%7, TYPD	; GO TYPE CHAR.
1002	003424	004737	003366		MOVB	%12, TYPDAT	; MOVE LF CODE TO TYPDAT
1003	003430	000740			JSR	%7, TYPD	; GO TYPE CHAR.
1004	003432	000000			BR	TYP A	
1005					TYPDAT: OPEN		
1006							
1007					; SUBROUTINE TO INITIALIZE STACK POINTER AND SET PROCESSOR PRIORITY		
1008					; LEVEL 7		
1009	003434	012777	001400	176106	INIT.	MOV	%CAT, %BASREG
1010	003442	012737	000340	177776		MOV	%PRTY7, %PSH
1011	003450	011637	001200			MOV	(%SP), %SPBOT
1012	003454	012706	001200			MOV	%SPBOT, %SP
1013	003460	000176	000000			JMP	%SP
1014					; RETURN		
1015							
1016					; SUBROUTINE TO GET DM11 PARAMETERS		
1017					; VECTOR ADDRESS		
1018	003464	023737	000042	000046	DMPAR:	CMP	%042, %046
1019	003472	001060				BNE	%5
1020						; ACT11?	
1021	003474	012700	000302			; BR IF NO	
1022	003500	010060	177776			; SIZE FOR INTERRUPT VECTOR IN AUTO MODE	
1023	003504	012720	000003		45:	MOV	%302, %R0
1024	003510	006720				MOV	%R0, -2(%R0)
1025	003512	022700	000776			MOV	%3, (%R0)+
1026	003516	100370				TST	(%R0)+
1027	003520	012737	003610	000014		CMP	%776, %R0
1028	003526	012737	000340	000016		BPL	%5
1029	003534	012737	177777	001440		MOV	%55, %014
1030	003542	012737	017102	001400	35:	MOV	%340, %016
1031	003560	012777	000105	175764		MOV	%-1, %CT
1032	003566	005037	177776			MOV	%OUTBUF, %CAT
1033	003562	012777	000001	175754		MOV	%BIT6+BIT2+BIT0, %CSR
1034	003570	012737	177777	002076		CLR	%PSH
						MOV	%LBITO, %BAR
						MOV	%-1, %COUNT

1035	003576	005337	002076		25	DEC	COUNT	
1036	003602	001375				BNE	25	
1037	003604	104001				ERROR		; NO INT OCCURRED
1038	003606	000752				BR	35	; REPEAT IT
1039	003610	162716	000004		55	SUB	#4, (SP)	; CALC INT VECT
1040	003614	011637	003650			MOV	(SP), @VECTOR	; STORE IT
1041	003620	012737	000016	000014		MOV	#16, @#14	; RESTORE BPT VECT
1042	003626	004737	004372			JSR	7, OVRLAY	; +2, HALT IN VECT AREA
1043	003632	000415				BR	VECOK	
1044	003634	004737	004372		65	JSR	7, OVRLAY	; PUT HALT, +2 IN VECTOR AREA
1045	003640	104000				TYPE		; ASK USER FOR RECEIVER INT VECTOR
1046	003642	017626				WHERE		; OF UNIT UNDER TEST
1047	003644	004537	004416			JSR	5, RECD	; GET VECTOR AND PUT IT
1048	003650	000000			VECTOR:	O		; HERE
1049	003652	005737	003650			TST	VECTOR	
1050	003656	001003				BNE	VECOK	
1051	003660	012737	000300	003650		MOV	#300, VECTOR	; SET VECTOR = TO 0300
1052	003666	023727	003650	000300	VECOK:	CMP	VECTOR, #300	; IS VECTOR HIGHER OR
1053	003674	103003				BHS	VECOKB	; EQUAL TO 0300
1054	003676	10400C			VECOKA:	TYPE		; TYPE '?'
1055	003700	020017				M1		
1056	003702	000670				BR	DMPAR	; ASK FOR ANOTHER VECTOR
1057	003704	023727	003650	000770	VECOKB:	CMP	VECTOR, #770	; IS VECTOR = TO OR
1058	003712	101371				BHI	VECOKA	; LESS THAN 770
1059	003714	032737	000007	003650		BIT	#7, VECTOR	; LSB OF VECTOR MUST BE ALL 0'S
1060	003722	001365				BNE	VECOKA	
1061	003724	013737	003650	001552		MOV	VECTOR, CLKINT	
1062	003732	062737	000004	003650		ADD	#4, VECTOR	
1063	003740	013737	003650	001556		MOV	VECTOR, XMTINT	
1064								
1065								
1066	003746	023737	000042	000046	; UNIT NUMBER	DMPARB:	CMP	@#42, @#46
1067	003754	001405				BEQ	UNIT+2	; ACT11?
1068	003756	104000				TYPE		; BR IF YES
1069	003760	017725				WHICH		
1070	003762	004537	004416			JSR	5, RECD	; GET UNIT AND PUT IT
1071	003766	000000			UNIT	O		; HERE
1072	003770	023727	003766	000017		CMP	UNIT, #17	; UNIT SELECTED MUST BE
1073	003776	101403				BLOS	UNTOKA	; BETWEEN 0 & 17
1074	004000	104000				TYPE		
1075	004002	020017				M1		
1076	004004	030760				BR	DMPARB	
1077	004006	006337	003766		UNTOKA:	ASL	UNIT	
1078	004012	006337	003766			ASL	UNIT	
1079	004016	006337	003766			ASL	UNIT	
1080	004022	012702	000004			MOV	#4, X2	
1081	004026	012701	001542			MOV	#CSR, X1	
1082	004032	042711	000370		UNTOKB:	BIC	#370, (1)	; FORM ADDRESSES OF
1083	004036	063721	003766			ADD	UNIT, (1)+	; REGISTERS OF UNIT SELECTED
1084	004042	005302				DEC	X2	
1085	004044	001372				BNE	UNTOKB	
1086								
1087	004046	012702	000004			MOV	#4, X2	
1088	004052	012703	001542			MOV	#CSR, X3	
1089	004056	012701	002106			MOV	# CSR, X1	
1090	004062	012311			UNTOKC:	MOV	(3)+, (	

1091	004064	005221			INC	(1)+	
1092	004066	005302			DEC	X2	
1093	004070	001374			BNE	UNOKC	
1094							
1095	004072	023737	000042	000045	CHARACTER LENGTH		
1096	004100	001405			DMPARC: CMP	2042, 2046	, ACT117
1097	004102	104000			BEQ	LENGTH+2	, BR IF YES
1098	004104	017740			TYPE		
1099	004106	004537	004416		LEVEL		
1100	004112	000000			JSR	5, RECD	, GET LENGTH AND PUT IT
1101	004114	005737	004112		LENGTH: 0		, HERE
1102	004120	001003			TST	LENGTH	
1103	004122	012737	000010	004112	BNE	LENOKA	
1104	004130	023727	004112	000005	MOV	#8, LENGTH	
1105	004136	103003			LENOKA: CMP	LENGTH, #5	, CHARACTER LENGTH SELECTED MUST
1106	004140	104000			BHIS	LENOKC	, BE BETWEEN 5-8
1107	004142	020017			LENOKB: TYPE		, CARRIAGE RETURN SELECTS 8
1108	004144	000752			M1		
1109	004146	023727	004112	000010	BR	DMPARC	
1110	004154	101371			LENOKC: CMP	LENGTH, #8	
1111	004156	162737	000005	004112	BHI	LENOKB	
1112	004164	006337	004112		SUB	#5, LENGTH	
1113	004170	013701	004112		ASL	LENGTH	
1114	004174	016137	004206	001572	MOV	LENGTH, X1	
1115	004202	000240			MOV	LENOKD(1), 20CARMK	, SET CHARACTER LENGTH MASK
1116	004204	000207			NOP		
1117					RTS	7	, EXIT PARAMETERS ROUTINE
1118							
1119							
1120	004206	177740					
1121	004210	177700					
1122	004212	177600					
1123	004214	177400					
1124							
1125							
1126	004216	005037	004366				
1127	004222	012737	177777	001440			
1128	004230	012737	017102	001400			
1129	004236	012777	004336	175306			
1130	004244	012777	000340	175302			
1131	004252	012777	000001	175264			
1132	004260	012777	000105	175254			
1133	004266	005037	177776				
1134	004272	012737	000044	002076			
1135	004300	062737	000001	004366			
1136	004306	001007					
1137	004310	006077	175226				
1138	004314	012737	000340	177776			
1139	004322	104001					
1140	004324	000734					
1141	004326	006337	002076				
1142	004332	001375					
1143	004334	000756					
1144	004336	006077	175200				
1145	004342	013737	004366	004370			
1146	004350	006037	004370				

, CHARACTER LENGTH
 DMPARC: CMP 2042, 2046
 BEQ LENGTH+2
 TYPE
 LEVEL
 JSR 5, RECD
 LENGTH: 0
 TST LENGTH
 BNE LENOKA
 MOV #8, LENGTH
 LENOKA: CMP LENGTH, #5
 BHIS LENOKC
 LENOKB: TYPE
 M1
 BR DMPARC
 LENOKC: CMP LENGTH, #8
 BHI LENOKB
 SUB #5, LENGTH
 ASL LENGTH
 MOV LENGTH, X1
 MOV LENOKD(1), 20CARMK
 NOP
 RTS 7

, ACT117
 , BR IF YES
 , GET LENGTH AND PUT IT
 , HERE
 , CHARACTER LENGTH SELECTED MUST
 , BE BETWEEN 5-8
 , CARRIAGE RETURN SELECTS 8
 , SET CHARACTER LENGTH MASK
 , EXIT PARAMETERS ROUTINE

, THE BELOW TABLE REPRESENTS THE CHARACTER LENGTH MASK FOR 5, 6, 7, AND 8
 ; BITS PER CHARACTER RESPECTIVELY.
 LENOKD: 177740
 177700
 177600
 177400

, CALCULATE MACHINE TIME TO TRANSMIT ONE CHARACTER
 TIMER: CLR TIME1
 MOV 0-1, MCT
 MOV 8OUTBUF, CAT
 MOV 8TIMEC, 2CLKINT
 MOV 8PTY7, 2CLKLVL
 MOV 8LBIT0, 20AR
 MOV 8BIT6+BIT2+BIT0, 2CSR
 CLR 20PSM
 MOV 844, COUNT
 ADD 81, TIME1
 BNE TIMEB
 CLR 2CSR
 MOV 8PTY7, PSM
 ERROR
 BR TIMER
 DEC COUNT
 BNE -4
 BR TIMEA
 TIMEC: CLR 2CSR
 MOV TIME1, TIME14
 ROR TIME14

; SET UP TO TRANSMIT
 ; 1 CHARACTER ON LINE 1
 ; LOAD RECEIVER INTERRUPT
 ; AND PRIORITY
 ; START TRANSMITTING
 ; SET IE BIT
 ; SET PROCESSER PRIORITY LEVEL = 0
 ; INCREMENT MACH TIME TO TRANSMIT
 ; SET PROCESSER PRIORITY LEVEL = 7
 ; TRANSMITTER FAILED TO INTERRUPT

Address	Hex	Dec	Label	Instruction	Comment
1147	004354	000241		CLC	
1148	004356	006037	004370	ROR	TIME14
1149	004362	022626		POPSP2	
1150	004364	000207		RTS	7
1151					; RESTORE STACK POINTER
1152	004366	000000			; EXIT TIME CALCULATION ROUTINE
1153	004370	000000		TIME1: OPEN	; CONTAINS MACHINE TIME TO XMIT 1 CHAR
1154				TIME14: OPEN	; CONTAIN TIME TO XMIT 1/4 CHAR
1155					; SUBROUTINE TO PUT HALT.. +2 IN VECTOR AREA (0300-1000)
1156	004372	012702	000302	OVRLAY: MOV	#302, R2
1157	004376	010262	177776	15: MOV	R2, -2(R2)
1158	004402	005022		CLR	(R2)+
1159	004404	005722		TST	(R2)+
1160	004406	022702	000776	CMP	#776, R2
1161	004412	100371		BPL	15
1162	004414	000207		RTS	7
1163					
1164					
1165					; SUBROUTINE TO RECEIVE DATA
1166					; THIS SUBROUTINE RECEIVES DATA FROM THE KEYBOARD (UP TO SIX OCTAL
1167					; DIGITS AND PLACES THEM INTO THE ADDRESS FOLLOWING THE SUBROUTINE
1168					; CALL (JSR 5, RECD). NO REGISTER CONTENTS ARE DISTURBED.
1169					
1170					
1171					; SUBROUTINE TO INPUT DATA FROM TTY
1172					
1173	004416	010046		RECD: MOV	RO, -(SP)
1174	004420	005015		15: CLR	(5)
1175	004422	012737	000007	MOV	#7, CNT
1176	004430	105777	175432	25: TSTB	@TKCSR
1177	004434	100375		BPL	25
1178	004436	117700	175426	MOVB	@TKDBR, RO
1179	004442	142700	000200	BICB	#200, RO
1180	004446	110077	175422	MOVB	RO, @TPDBR
1181	004452	122700	000025	CPMB	#25, RO
1182	004456	001443		BEQ	55
1183	004460	122700	000015	CPMB	#15, RO
1184	004464	001415		BEQ	65
1185	004466	142700	000060	BICB	#60, RO
1186	004472	132700	000110	BITB	#110, RO
1187	004476	001031		BNE	75
1188	004500	006315		ASL	(5)
1189	004502	006315		ASL	(5)
1190	004504	006315		ASL	(5)
1191	004506	150015		BISB	RO, (5)
1192	004510	006337	004574	DEC	CNT
1193	004514	001422		BEQ	75
1194	004516	000744		BR	25
1195	004520	105777	175346	65: TSTB	@TPCSR
1196	004524	100375		BPL	65
1197	004526	012777	000012	MOV	#12, @TPDBR
1198	004534	105777	175332	85: TSTB	@TPCSR
1199	004540	100375		BPL	85
1200	004542	005077	175326	CLR	@TPDBR
1201	004546	105777	175320	95: TSTB	@TPCSR
1202	004552	100375		BPL	95

1203	004554	005725			TST	(R5)+			
1204	004556	012600			MOV	(SP)+, R0		ADJUST R5	
1205	004560	000205			RTS	R5		RESTORE R0	
1206	004562	104000		75	TYPE				
1207	004564	020017			M1				
1208	004566	104000		55:	TYPE				
1209	004570	017666			SCTLU				
1210	004572	000712			BR	15			
1211	004574	000000		CNT:	0			START OVER	
1212									
1213									
1214	004576	012737	004606	000024	PFail:	MOV	BPWRUP, 24		
1215	004604	000000				HALT			
1216									
1217									
1218	004606	000005			PHRUP:	RESET			
1219	004610	012706	001200			MOV	BSPBOT, X6		GIVE TELEPRINTER TIME TO START
1220	004614	104001				ERROR			TYPE POWER FAIL ERROR
1221	004616	000137	002514			JMP	BARSTAT2		GO RESTART PROGRAM
1222									
1223									
1224									
1225									
1226									
1227									
1228									
1229									
1230									
1231									
1232									
1233									
1234	004622	012537	016730		XMTTST:	MOV	(5)+, @LINE		GET LINE NUMBER
1235	004626	004737	007044			JSR	7, GTLINE		GO FORM LINE BIT (FOR BAR)
1236	004632	005037	001570			CLR	XMTDAT		
1237	004636	004537	006246		15:	JSR	5, @XMITD		GO TO TRANSMIT SUBROUTINE
1238	004642	177777				-1			TRANSMIT ONE CHARACTER
1239	004644	012703	000010			MOV	@10, X3		WAIT IN
1240	004650	005002				CLR	X2		THIS
1241	004652	005302			25:	DEC	X2		LOOP
1242	004654	001376				BNE	-2		UNTIL THE
1243	004656	005303				DEC	X3		TRANSMITTER
1244	004660	001374				BNE	25		IS FINISHED
1245	004662	017737	174656	001566		MOV	@BAR, RCVDAT		BAR SHOULD NOW BE CLEAR
1246	004670	001401				BEQ	35		BRANCH IF IT IS
1247	004672	104011				ERROR1			ERROR! BAR BIT FAILED TO CLEAR
1248	004674	005777	174642		35:	TST	@CSR		TEST READY BIT SHOULD BE SET
1249	004700	100401				BMI	45		BRANCH IF SET
1250	004702	104001				ERROR			ERROR! READY NOT SET
1251	004704	013701	016730		45:	MOV	@LINE, X1		GET LINE NUMBER
1252	004710	016137	001440	001566		MOV	WCT(1), RCVDAT		WORD COUNT SHOULD BE 0
1253	004716	001401				BEQ	55		
1254	004720	104011				ERROR1			ERROR! WORD COUNT NOT EQUAL TO 0
1255	004722	012737	017102	001570	55:	MOV	@OUTBUF, XMTDAT		
1256	004730	016137	001400	001566		MOV	CAT(1), RCVDAT		CURRENT ADDRESS SHOULD NOT HAVE INCREMENTED
1257	004736	023737	001566	001570		CMP	RCVDAT, XMTDAT		
1258	004744	001401				BEQ	65		

1259	00746	104011				ERROR1		
1260	004750	012777	005002	174600	65:	MOV	#75, @XMTINT	; ERROR! CURRENT ADDR. DID NOT INCREMENT
1261	004756	052777	010000	174556		BIS	#BIT12, @CSR	; LOAD TRANSMITTER INTERRUPT VECTOR
1262	004764	005037	177776			CLR	@PSW	; ENABLE TRANSMITTER INTERRUPT
1263	004770	000240				NOP		; SET PROCESSOR PRIORITY = 0
1264	004772	012737	000340	177776		MOV	#PRTY7, @PSW	; LOCK OUT INTERRUPTS
1265	005000	104001				ERROR		; TRANSMITTER FAILED TO INTERRUPT OR
1266								; INTERRUPTED TO WRONG LOCATION AND HALTED WITH ADDRESS +2 DISPLAYED
1267	005002	022626			75:	CMP	(6)+, (6)+	; RESET STACK PTR
1268	005004	012737	000340	177776		MOV	#PRTY7, @PSW	; LOCK OUT INTERRUPTS
1269	005012	042777	110000	174522		BIC	#BIT12+BIT15, @CSR	; CLEAR XMIT IE & READY BITS
1270	005020	005777	174516			TST	@CSR	; TEST THAT READY CLEARED
1271	005024	100001				BPL	85	; GO TO EXIT
1272	005026	104001				ERROR		; ERROR! READY FAILED TO CLEAR
1273	005030	005726			85:	TST	(6)+	; RESET STACK PTR
1274	005032	104006				SCOPE		; SCOPE

1275
 1276
 1277
 1278
 1279
 1280
 1281
 1282
 1283
 1284
 1285
 1286
 1287
 1288
 1289
 1290
 1291
 1292
 1293
 1294
 1295
 1296
 1297
 1298
 1299
 1300
 1301
 1302
 1303
 1304
 1305
 1306
 1307
 1308
 1309
 1310
 1311
 1312
 1313
 1314
 1315
 1316
 1317
 1318
 1319
 1320
 1321
 1322
 1323
 1324
 1325
 1326
 1327
 1328
 1329
 1330

005034 012737 177777 017102
 005042 005037 001600
 005046 005037 001602
 005052 012737 000340 177776
 005060 012537 016730
 005064 004537 006246
 005070 177777
 005072 052777 000001 174442
 005100 005777 174436
 005104 100375
 005106 042777 100000 174426
 005114 005046
 005116 105777 174420
 005122 100404
 005124 005216
 005126 001373
 005130 104001
 005132 000550
 005134 005726
 005136 012777 005172 174406
 005144 052777 000100 174370
 005152 005037 177776
 005156 000240
 005160 012737 000340 177776
 005166 104001
 005170 000531
 005172 012737 000340 177776
 005200 022626
 005202 042777 000300 174332
 005210 105777 174326
 005214 100002
 005216 104001
 005220 000515

, RECEIVER LINE TESTS
 , THE RECEIVER LINE TEST SUBROUTINE IS ENTERED WITH
 , A JSR 5, RCVTST INSTRUCTION FOLLOWED BY THE
 , LINE BIT AND LINE NUMBER OF THE LINE TO BE
 , TESTED. THE SUBROUTINE PERFORMS THE FOLLOWING
 , TEST AS SHOWN BELOW IN THE EVENT OF AN ERROR
 , THE REMAINING TESTS ARE ABORTED
 , TEST SEQUENCE AND ADDRESS TAG

, CHARACTER DONE SETS	RTSTA
, CHARACTER DONE CAUSES INTERRUPT	RTSTB
, CHARACTER DONE CAN BE CLEARED	RTSTC
, TUMBLE TABLE ENTRY IS CORRECT	RTSTD
, NO ENTRY IN NEXT TABLE ADDRESS	RTSTE
, HARDWARE TABLE POINTER INCREMENTED	RTSTF
, NEXT ENTRY WAS CORRECT	RTSTG

, NOTES: IF THE HARDWARE PROVIDES AN INCORRECT VECTOR
 , ADDRESS THE PROGRAM WILL HALT AND DISPLAY
 , THE INCORRECT VECTOR+2 IN THE ADDRESS LIGHTS.

RCVTST: MOV	8-1, OUTBUF	, LOAD ALL 1'S INTO OUTPUT BUFFER
CLR	TUMTAB	, CLEAR THE FIRST
CLR	TUMTAB+2	, TWO TUMBLE TABLE ADDRESSES
MOV	8PTY7, 2APSW	, LOCK OUT INTERRUPTS
MOV	(5)+, LINE	, GET LINE NUMBER
JSR	5, 2AXMITD	, TRANSMIT 1 CHARACTER (0'S)
-1		, ON LINE SPECIFIED BY JSR
BIS	8BIT0, 2CSR	, SET GO BIT
TST	2CSR	, WAIT FOR TRANSMITTER
BPL	-4	, TO TRANSMIT 1 CHAR.
BIC	8BIT15, 2CSR	, CLEAR TRANSMITTER READY FLAG
CLR	-(SP)	, SET WATCH DOG TIMER
TSTB	2CSR	, TEST CHAR. DONE FLAG
BMI	25	, BRANCH IF SET
INC	(SP)	, WAIT FOR THE FLAG
BNE	15	
ERROR		, ERROR! CHAR. DONE FLAG FAILED TO SET
BR	85	, GO TO EXIT
25:		
TST	(SP)+	, RESTORE STACK PTR
MOV	835, 2CLKINT	, LOAD RECEIVER INTERRUPT VEC. ADRS.
BIS	8BIT6, 2CSR	, SET RECEIVER IE BIT
CLR	2APSW	, ENABLE INTERRUPTS
NOP		
MOV	8PTY7, PSW	, LOCK OUT INTERRUPTS
ERROR		, RECEIVER FAILED TO INTERRUPT
BR	85	, GO TO EXIT
35:		
MOV	8PTY7, 2APSW	, LOCK OUT INTERRUPTS
CMP	(6)+, (6)+	
BIC	8BIT7+8BIT6, 2CSR	, CLEAR CHAR. DONE FLAG
TSTB	2CSR	, TEST THAT CHAR DONE FLAG CLEARED
BPL	45	, BRANCH IF CHAR. DONE FLAG CLEARED
ERROR		, ERROR! CHAR. DONE FAILED TO CLEAR
BR	85	, GO TO EXIT

```

1331 005222 013737 001600 001566 45: MOV TUMTAB,RCV DAT ;GET TUMBLE TABLE ENTRY
1332 005230 042737 020000 001566 BIC #BIT13,RCV DAT ;CLEAR PARITY INDICATOR
1333 005236 012737 000377 001570 MOV #377,XMT DAT ;LOAD XMT DAT WITH TRANSMITTED DATA
1334 005244 043737 001572 001570 BIC CARMSK,XMT DAT ;CLEAR NON TRANSMITTED BITS
1335 005252 153737 016730 001571 BISB LINE,XMT DAT+1 ;LOAD XMT DAT WITH PROPER LINE #
1336 005260 052737 100000 001570 BIS #BIT15,XMT DAT ;SET VALID DATA ENTRY BIT IN XMT DAT
1337 005266 023737 001566 001570 CMP RCV DAT,XMT DAT ;COMPARE TUMBLE TABLE ENTRY (RCV DAT) &
1338 ;CORRECT RESULT (XMT DAT)
1339
1340 005274 001402 BEQ 55
1341 005276 104011 ERROR1 ;ERROR! INCORRECT TUMBLE TABLE
1342 005300 000465 BR 85 ;ENTRY; GO TO EXIT
1343 005302 005037 001600 55: CLR TUMTAB
1344 005306 013737 001602 001566 MOV TUMTAB+2,RCV DAT ;GET NEXT ENTRY
1345 005314 001404 BEQ 65 ;BRANCH IF ALL 0'S
1346 005316 005037 001570 CLR XMT DAT
1347 005322 104011 ERROR1 ;ERROR! FALSE ENTRY IN NEXT
1348 005324 000465 BR 85 ;TUMBLE TABLE ADDRESS
1349 005326 004537 006246 65: JSR 5,2AXMITD ;TRANSMIT 1 CHARACTER (ALL 1'S)
1350 005332 177777 -1 ;ON LINE SPECIFIED BY JSR
1351 005334 005777 174202 TST @CSR ;WAIT FOR TRANSMITTER
1352 005340 100375 BPL -4 ;READY FLAG
1353 005342 105777 174174 TSTB @CSR ;TEST FOR THE DONE FLAG
1354 005346 100375 BPL -4
1355 005350 042777 000200 174164 BIC #BIT7,@CSR ;CLEAR CHAR. DONE FLAG
1356 005356 013737 001600 001566 MOV TUMTAB,RCV DAT ;TEST THAT HARDWARE TUMBLE
1357 005364 001404 BEQ 75 ;TABLE POINTER INCREMENTED (+2)
1358 005366 005037 001570 CLR XMT DAT
1359 005372 104011 ERROR1 ;ERROR! TUMBLE TABLE POINTER DID
1360 005374 000427 BR 85 ;NOT INCREMENT; GO TO EXIT
1361 005376 013737 001602 001566 75: MOV TUMTAB+2,RCV DAT ;GET TUMBLE TABLE ENTRY
1362 005404 042737 020000 001566 BIC #BIT13,RCV DAT ;CLEAR PARITY INDICATOR
1363 005412 012737 000377 001570 MOV #377,XMT DAT ;LOAD XMT DAT WITH TRANSMITTED DATA
1364 005420 043737 001572 001570 BIC CARMSK,XMT DAT ;CLEAR NON-TRANSMITTED BITS
1365 005426 153737 016730 001571 BISB LINE,XMT DAT+1 ;LOAD LINE # INTO XMT DAT
1366 005434 052737 100000 001570 BIS #BIT15,XMT DAT ;SET VALID DATA ENTRY BIT INTO XMT DAT
1367 005442 023737 001566 001570 CMP RCV DAT,XMT DAT ;COMPARE TUMBLE TABLE ENTRY (RCV DAT) &
1368 ;CORRECT RESULT (XMT DAT)
1369 005450 001401 BEQ 85
1370 005452 104011 ERROR1 ;ERROR! 2ND TUMBLE TABLE ENTRY
1371 005454 104006 85: SCOPE ;WAS INCORRECT; SCOPE
1372
1373
1374 ;SUBROUTINE TO TEST BREAK OPERATION
1375 ;THE TRANSMITTER WILL TRANSMIT THE BREAK FOR TWO CHARACTER
1376 ;TIMES AND THEN THE FOLLOWING TESTS WILL BE PERFORMED
1377 ;
1378 ; A VALID DATA ENTRY WAS MADE BKTSTB
1379 ; BREAK BIT SET BKTSTC
1380 ; DATA WAS ALL 0'S BKTSTD
1381 005456 012777 000001 174056 BRKTST: MOV #1,@CSR ;SET THE GO BIT
1382 005464 011577 174056 MOV (5),@BKRCSR ;SET THE BREAK BIT
1383 005470 105777 174046 TSTB @CSR ;WAIT FOR THE RECEIVER TO
1384 005474 100375 BPL -4 ;RECEIVE BREAK
1385 005476 042777 000200 174036 BIC #BIT7,@CSR ;CLEAR FLAG
1386 005504 105777 174032 TSTB @CSR ;WAIT FOR THE RECEIVER TO
1387 005510 100375 BPL -4 ;TO RECEIVE BREAK

```

1387	005512	042777	000200	174022		BIC	#BIT7, @CSR	; CLEAR FLAG
1388	005520	005077	174022			CLR	@KCSR	; CLEAR BREAK BIT
1389	005524	005737	001600		15:	TSI	TUMTAB	; TEST FOR VALID DATA ENTRY
1390	005530	100402				BMI	25	
1391	005532	104001				ERROR		; ERROR! NO VALID DATA ENTRY
1392	005534	000421				BR	45	; GO TO EXIT
1393	005536	032737	040000	001600	25:	BIT	#BIT14, TUMTAB	; TEST THAT BREAK BIT IS SET
1394	005544	001002				BNE	35	; IN TUMBLE TABLE
1395	005546	104001				ERROR		; ERROR! BREAK BIT FAILED TO SET
1396	005550	000413				BR	45	; GO TO EXIT
1397	005552	105737	001600		35	TSTB	TUMTAB	; TEST THAT DATA IS ALL 0'S
1398	005556	001410				BEQ	45	
1399	005560	005037	001566			CLR	RCVDAT	
1400	005564	113737	001600	001566		MOVB	TUMTAB, RCVDAT	; GET RECEIVED DATA
1401	005572	005037	001570			CLR	XMTDAT	
1402	005576	104011				ERROR1		; ERROR! DATA WAS NOT ALL 0'S
1403	005600	104006			45	SCOPE		; SCOPE
1404								
1405								
1406								
1407								
1408								
1409	005602	012537	002076					
1410	005606	005037	001570					
1411	005612	004537	006224					
1412	005616	020233						
1413	005620	017102						
1414	005622	000100						
1415	005624	012737	000001	001564				
1416	005632	005037	016730					
1417	005636	012777	000001	173676	15:			
1418	005644	004537	006246		25:			
1419	005650	177700						
1420	005652	013737	004366	005666				
1421	005660	013704	002076					
1422	005664	104400			35:			
1423	005666	000000			45:			
1424	005670	005304						
1425	005672	001374						
1426	005674	052737	000002	016730				
1427	005702	006337	001564					
1428	005706	103356						
1429	005710	012704	000100					
1430	005714	013737	004366	05724				
1431	005722	104400			55:			
1432	005724	000000			65:			
1433	005726	005304						
1434	005730	001374						
1435	005732	017737	173606	001566				
1436	005740	001402						
1437	005742	104011						
1438	005744	000413						
1439	005746	022777	100201	173566	75:			
1440	005754	001407						
1441	005756	012737	100201	001570				
1442	005764	017737	173552	001566				

; SUBROUTINE TO TRANSMIT & RECEIVE ON ALL LINES THE DELAY BETWEEN
 ; TRANSMITTING ON A LINE IS SUPPLIED BY THE CALLING JSR INSTRUCTION
 ; NOTE NO DATA CHECKING IS PERFORMED BY THIS TEST

DLXMT: MOV (5)+, @COUNT ; GET CHARACTER DELAY COUNT
 CLR XMTDAT
 JSR 5, @MOVE ; LOAD OUTPUT BUFFER WITH DATA
 MSG1 ; TO BE TRANSMITTED
 OUTBUF
 64.
 MOV @LBIT0, @LNBIT
 CLR @LINE
 MOV @BIT0, @CSR ; SET THE GO BIT
 JSR 5, @XMITD ; TRANSMIT 64. CHAR.
 -64. ; ON A LINE
 MOV @TIME1, 45
 MOV @COUNT, X4 ; GET CHARACTER DELAY COUNT
 DELAY
 0
 DEC X4
 BNE 35
 ADD @2, LINE ; FORM NEXT LINE NUMBER
 ASL LNBIT ; SHIFT LINE BIT
 BCC 25 ; BRANCH IF ALL LINES NOT DONE
 MOV @64, X4
 MOV TIME1, 65
 DELAY
 0
 DEC X4
 BNE 55
 MOV @BAR, RCVDAT ; GET & TEST BAR DATA
 BEQ 75 ; EXIT IF DONE
 ERROR1 ; ERROR! BAR SHOULD'VE BEEN CLEAR
 BR 85
 CMP @100201, @CSR ; TEST THAT ONLY DONE, GO, & READY BITS ARE SET
 BEQ 85
 MOV @100201, XMTDAT
 MOV @CSR, RCVDAT ; GET CSR CONTENTS

1443	005772	104011				ERROR1		
1444	005774	005726				POPSP		, INCORRECT CSR CONTENTS
1445	005776	104006			85:	SCOPE		, RESET THE STACK
1446								, SCOPE
1447								
1448	006000	011637	006050			; SUBROUTINE TO DELAY A SPECIFIED NUMBER OF MILLISECONDS		
1449	006004	062716	000002			DLY MOV (6), 35		, GET DELAY COUNT ADDRESS
1450	006010	017737	000034	006050		ADD #2, (6)		, SET UP EXIT ADDRESS
1451	006016	001413				MOV #35, 35		, GET DELAY COUNT
1452	006020	012737	000050	006052	15	BEQ 25		, EXIT IF NO DELAY
1453	006026	162737	000001	006050		MOV #50, 45		
1454	006034	001404				SUB #1, 35		
1455	006036	005337	006052			BEQ 25		
1456	006042	001375				DEC 45		
1457	006044	000765				BNE -4		
1458	006046	000002			25	BR 15		
1459	006050	000000			35	RTI		, EXIT
1460	006052	000000			45	OPEN		, CONTAINS DELAY COUNT
1461								, CONTAINS DELAY ROUTINE CONSTANT
1462								
1463	006054	012737	177777	006076		; SUBROUTINE TO INITIALIZE BINARY COUNT PATTERNS		
1464	006062	004537	006224			INBIN MOV #1, RIND		, SET ALL VARIABLES
1465	006066	006076				JSR #5, BMOVE		, TO MINUS 1.
1466	006070	006077				RIND		
1467	006072	000005				RIND+1		
1468	006074	000207				5		
1469	006076	000000				RTS #7		, EXIT
1470	006100	000000			RIND:	OPEN		
1471	006102	000000			PTO:	OPEN		
1472					PT1:	OPEN		
1473								
1474	006104	013737	006100	006102		; SPECIAL BINARY COUNT PATTERN SUBROUTINE. EXITS WITH BIN CHAR IN R1		
1475	006112	005137	006102			GTBIN MOV PTO, PT1		, PREVIOUS BIN CHAR TO PT1
1476	006116	005137	006076			COM PT1		
1477	006122	001002				COM RIND		
1478	006124	005237	006102			BNE +6		
1479	006130	042737	177400	006102		INC PT1		
1480	006136	013737	006102	006100		BIC #177400, PT1		, MASK TO 8 BITS
1481	006144	013701	006102			MOV PT1, PTO		, SAVE BIN CHAR IN PTO
1482	006150	000240				MOV PT1, X1		, BIN CHAR TO R1
1483	006152	000207				NOP		
1484						RTS #7		, EXIT
1485								
1486	006154	104007				; OCTAL TO ASCII CONVERT ROUTINE		
1487	006156	013504				OACNV: SAVREG		, SAVE REGISTERS ON THE STACK
1488	006160	012501				MOV #5, X4		, GET OCTAL VALUE
1489	006162	012502				MOV (5), X1		, GET DESTINATION ADDR
1490	006164	060201				MOV (5), X2		, GET CONVERT COUNT
1491	006166	010403				ADD X2, X1		, DEVELOP ADDR TO STORE 1ST CHAR
1492	006170	042703	177770			OACNVA: MOV X4, X3		
1493	006174	062703	000060			BIC #177770, X3		, ISOLATE LEAST SIGNIFICANT DIGIT
1494	006200	110341				ADD #60, X3		, CONVERT DIGIT TO ASCII
1495	006202	042704	000007			MOVB X3, -(1)		, STORE ASCII CHARACTER
1496	006206	006004				BIC #7, X4		
1497	006210	006004				ROR X4		
1498	006212	006004				ROR X4		

1499	006214	005302				DEC	%2		: DONE ALL DIGITS?
1500	006216	001363				BNE	OACNVA		: BRANCH IF NOT DONE.
1501	006220	104010				RSTREG			: RESTORE THE REGISTERS
1502	006222	000205				RTS	%5		: DONE. EXIT.
1503									
1504									
1505	006224	104007				: SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES.			
1506	006226	012501				BMOVE:	SAVREG		: SAVE REGS.
1507	006230	012502				MOV	(5)+,%1		: GET "FROM" ADDRESS
1508	006232	012503				MOV	(5)+,%2		: GET "TO" ADDRESS
1509	006234	112122				MOV	(5)+,%3		: GET COUNT
1510	006236	005303			15	MOVB	(1)+,(2)+		: MOVE BYTE
1511	006240	001375				DEC	%3		: DECREMENT COUNT
1512	006242	104010				BNE	15		: BRANCH IF NOT DONE
1513	006244	000205				RSTREG			: RESTORE REGS.
1514						RTS	%5		: DONE EXIT
1515									
1516						: SUBROUTINE TO TRANSMIT DATA SUBROUTINE CALLED BY			
1517	006246	010046				: JSR 5,XMITD			
1518	006250	013700	016730			XMITD:	MOV	%0,-(SP)	: SAVE RO ON THE STACK
1519	006254	004737	007044				MOV	%LINE,%0	: GET LINE
1520	006260	012777	001400	173262			JSR	7,%GETLINE	: FORM LINE BIT (FOR BAR)
1521	006266	012760	017102	001400			MOV	%CAT,%BASEREG	: INITIALIZE BASE REGISTER
1522	006274	012560	001440				MOV	%OUTBUF,CAT(0)	: LOAD FIRST CHAR ADDRESS IN CAT
1523	006300	053777	001564	173236			MOV	(5)+,%CT(0)	: GET WORD COUNT
1524	006306	012600					BIS	%LINEBIT,%BAR	: LOAD LINE POSITION INTO BAR
1525	006310	000205					MOV	(SP)+,%0	: RESTORE RO
1526							RTS	5	: EXIT
1527									
1528						: ROUTINE TO TEST A LINE			
1529						: THE LINE TO BE TESTED IS PROVIDED BY THE JSR CALL TO THE ROUTINE			
1530	006312	012537	016730			: 100 CHARACTERS ARE TRANSMITTED, RECEIVED AND CHECKED BY THIS ROUTINE			
1531	006316	012737	000144	002076		DATTST:	MOV	(5)+,%LINE	: GET LINE NUMBER
1532	006324	012702	017102			DATTAR:	MOV	%100,%COUNT	: GET CHARACTER COUNT
1533	006330	004737	006104				MOV	%OUTBUF,%2	: GET ADDRESS OF OUTPUT BUFFER
1534	006334	110122			15	JSR	7,%GETBIN		: GET DATA
1535	006336	005337	002076				MOVB	%1,(2)+	: LOAD OUTPUT BUFFER WITH DATA
1536	006342	001372					DEC	COUNT	: GOT ALL DATA?
1537	006344	012701	001600				BNE	15	
1538	006350	005037	001600				MOV	%TUMTAB,%1	: LOAD TUMBLE TABLE POINTER
1539	006354	004537	006224				CLR	TUMTAB	
1540	006360	001600					JSR	5,BMOVE	: CLEAR
1541	006362	001601					TUMTAB		: TUMBLE
1542	006364	000177					TUMTAB+1		: TABLE
1543	006366	012702	017246				177		: 64 WORDS
1544	006372	052777	000001	173142			MOV	%INBUF,%2	: SETUP INPUT BUFFER POINTER
1545	006400	004537	006246				BIS	%BIT0,%CSR	: SET THE GO BIT
1546	006404	177634					JSR	5,%XMITD	: TRANSMIT
1547	006406	032777	160000	173126	25		-100		: 100 CHARACTERS
1548	006414	001004					BIT	%BIT15+BIT14+BIT13,%CSR	: TEST IF READY OR ANY ERROR
1549	006416	105777	173120				BNE	35	: FLAGS ARE SET
1550	006422	100371					TSTB	%CSR	: WAIT FOR THE RECEIVER
1551	006424	000415					BPL	25	: TO RECEIVE A CHARACTER
1552	006426	032777	060000	173106	35		BR	55	
1553	006434	001403					BIT	%BIT14+BIT13,%CSR	: TEST FOR ERROR FLAGS
1554	006436	104001					BEQ	45	: BRANCH NO ERROR
							ERROR		

1555	006440	000137	006760		JMP	155	.GO EXIT
1556	006444	042777	100000	173070 45	BIC	#BIT15, @CSR	.CLEAR TRANSMITTER READY FLAG
1557	006452	105777	173064		TSTB	@CSR	.TEST FOR CHARACTER READY
1558	006456	100375			BPL	-4	
1559	006460	042777	030200	173054 55	BIC	#BIT7, @CSR	.CLEAR CHAR DONE BIT
1560	006466	005711			TST	(1)	
1561	006470	100401			BMI	+4	.TEST FOR VALID ENTRY
1562	006472	104001			ERROR		.REPORT INVALID ENTRY
1563	006474	111122			MOVB	(1), (2)+	.MOVE CHAR FROM TUM. TAB TO INPUT BUFFER
1564							
1565							
1566	006476	012705	000001				.ROUTINE TO STORE RECEIVED PARITY BIT IN PARITY BIT BUFFER
1567	006502	000261			MOV	#1, X5	.GET ROTATE COUNT
1568	006504	032711	020000		SEC		.SET THE CARRY BIT
1569	006510	001001			BIT	#BIT13, (1)	.TEST RECEIVED PARITY BIT
1570	006512	000241			BNE	65	.BRANCH IF RECEIVED PARITY WAS ODD
1571	006514	004537	006764	65	CLC		.CLEAR CARRY BIT
1572					JSR	5, RORPARBUF	.ROTATE RECEIVED PARITY INTO PARITY BUFFER
1573							
1574	006520	011137	001562				.ROUTINE TO TEST THAT ENTRY IS FOR THE CORRECT LINE
1575	006524	042737	160777	001562	MOV	(1), @TTDAT	.GET TABLE ENTRY
1576	006532	123737	016730	001563	BIC	#160777, TTDAT	.CLEAR ALL BUT LINE NUMBER
1577	006540	001410			CMPB	LINE, TTDAT+1	.COMPARE LINE NUMBERS
1578	006542	013737	016730	001570	BEQ	75	
1579	006550	013737	001562	001566	MOV	LINE, XMTDAT	.GET CORRECT LINE # (X2)
1580	006556	104011			MOV	TTDAT, RCVDAT	.GET LINE # (X2) THAT FALSE DATA CAME IN ON
1581					ERROR1		.ERROR! DATA CAME IN ON A LINE THAT
1582	006560	000477					.PROGRAM WAS NOT TRANSMITTING ON
1583	006562	020127	001776	75	BR	155	.EXIT TEST
1584	006566	001002			CMP	X1, @TUMTAB+176	.IS POINTER AT THE END
1585	006570	012701	001576		BNE	85	.OF THE TABLE
1586	006574	005721			MOV	@TUMTAB-2, X1	
1587	006576	010046			TST	(1)+	.INCREMENT POINTER
1588	006600	013700	016730		MOV	X0, -(6)	.SAVE REGISTER ZERO
1589	006604	005760	001440		MOV	LINE, X0	.FETCH LINE NUMBER
1590	006610	001402			TST	WCT(0)	.HAS THE LAST CHARACTER BEEN TRANSMITTED
1591	006612	012600			BEQ	+6	.LAST CHARACTER HAS BEEN TRANSMITTED
1592	006614	000674			MOV	(6)+, X0	.RESTORE REGISTER ZERO
1593	006616	012600			BR	25	.GO WAIT FOR NEXT CHARACTER
1594	006620	012701	017102		MOV	(6)+, X0	.RESTORE REGISTER ZERO
1595	006624	012702	017246		MOV	@OUTBUF, X1	
1596	006630	012706	000014		MOV	@INBUF, X2	
1597	006634	004537	006764		MOV	#12, X5	.ROTATE PARITY BUFFER
1598	006640	006037	001566	105	JSR	5, RORPARBUF	.12 PLACES RIGHT
1599	006644	006037	001570		CLR	RCVDAT	
1600	006650	020127	017245		CLR	XMTDAT	
1601	006654	001441			CMP	X1, @OUTBUF+99	.HAVE ALL CHARS BEEN COMPARED
1602	006656	112137	001570		BEQ	155	
1603	006662	043737	001572	001570	MOVB	(1)+, XMTDAT	.GET TRANSMITTED CHARACTER
1604	006670	111237	001566		BIC	CARMSK, XMTDAT	.CLEAR NON-TRANSMITTED BITS
1605	006674	104002			MOVB	(2), RCVDAT	.GET RECEIVED CHARACTER
1606					DATCHK		.COMPARE TRANS. & RCVD CHARS
1607							
1608	006676	012703	000010				.ROUTINE TO COMPUTE AND CHECK PARITY ON RECEIVED DATA
1609	006702	005000			115: MOV	#8, X3	.GET BIT COUNTER
1610	006704	106037	001566		CLR	X0	.CLEAR COMPUTED PARITY INDICATOR
					125: RORB	RCVDAT	.LOOK AT RECEIVED BIT

1611	006710	103001		BCC	135		; BRANCH IF A 0
1612	006712	005100		COM	%0		; COMPLEMENT R0 IF A 1
1613	006714	005303		DEC	%3		; DECREMENT BIT COUNTER
1614	006716	001372		BNE	125		; LOOK AT NEXT BIT IF NOT DONE
1615	006720	000240		NOP			; IF COMPUTED PARITY WAS ODD R0 WILL
1616							; CONTAIN ALL 1'S, IF EVEN R0 = 0
1617	006722	112237	001566	MOVB	(2)+, RCV0AT		; GET RECEIVED CHARACTER
1618	006726	012705	000001	MOV	#1, %5		; ROTATE PARITY BUFFER 1 PLACE
1619	006732	004537	006764	JSR	5, RORPARBUF		; RIGHT LEAVING RECEIVED PARITY BIT IN CARRY
1620	006736	103004		BCC	145		; BRANCH IF RECEIVED PARITY WAS EVEN
1621	006740	005700		TST	%0		; TEST FOR COMPUTED ODD PARITY
1622	006742	001336		BNE	105		; BRANCH IF COMPUTED & RECEIVED WAS ODD
1623	006744	104001		ERROR			; ERROR! COMPUTED = EVEN, RECEIVED = ODD
1624	006746	000734		BR	105		; CONTINUE TEST
1625	006750	005700		TST	%0		; TEST FOR EVEN COMPUTED PARITY
1626	006752	001732		BEQ	105		; BRANCH IF COMPUTED PARITY WAS EVEN
1627	006754	104001		ERROR			; ERROR! COMPUTED = ODD, RECEIVED = EVEN
1628	006756	000730		BR	105		; CONTINUE TEST
1629	006760	005726		POPS			; REPOSITION STACK POINTER
1630	006762	104006		SCOPE			; SCOPE
1631							
1632							
1633	006764	006037	007026	; ROUTINE TO ROTATE PARITY BUFFER.			
1634	006770	006037	007030	RORPARBUF: ROR	PAR0		
1635	006774	006037	007032	ROR	PAR1		
1636	007000	006037	007034	ROR	PAR2		
1637	007004	006037	007036	ROR	PAR3		
1638	007010	006037	007040	ROR	PAR4		
1639	007014	006037	007042	ROR	PAR5		
1640	007020	005316		ROR	PAR6		
1641	007022	001360		DEC	(SP)		; DECREMENT ROTATE COUNT
1642	007024	000205		BNE	RORPARBUF		
1643				RTS	5		
1644	007026	000000		; PARITY BUFFER			
1645	007030	000000		PAR0:	OPEN		
1646	007032	000000		PAR1:	OPEN		
1647	007034	000000		PAR2:	OPEN		
1648	007036	000000		PAR3:	OPEN		
1649	007040	000000		PAR4:	OPEN		
1650	007042	000000		PAR5:	OPEN		
1651				PAR6:	OPEN		
1652							
1653	007044	010046		; SUBROUTINE TO FORM LINE BIT POSITION WITH THE LINE # IN LINE			
1654	007046	005037	001564	GTLINE: MOV	%0, -(SP)		; SAVE R0 ON THE STACK
1655	007052	013700	016730	CLR	%LINEBIT		; CLEAR LINE BIT
1656	007056	000261		MOV	%LINE, %0		; GET LINE
1657	007060	006137	001564	SEC			; SET CARRY
1658	007064	162700	000002	15: ROL	LINEBIT		; SHIFT LINE BIT
1659	007070	100373		SUB	#2, %0		; SUBTRACT 2 FROM LINE NUMBER
1660	007072	012600		BPL	15		; BRANCH IF GREATER THAN 0
1661	007074	000207		MOV	(SP)+, %0		; RESTORE R0
1662				RTS	7		; EXIT SUBROUTINE

1663							
1664	007076	104000				PRGO-	TYPE
1665	007100	020103					PRGOM
1666	007102	012737	007136	002000		MOV	#RTO, KSTART ; GET ADDRESS OF FIRST TEST
1667	007110	005037	002004			CLR	RTNNO ; CLEAR ROUTINE #
1668	007114	000137	002546			JMP	SRSET
1669	007120	012737	007136	002000	PRGOR:	MOV	#RTO, KSTART ; GET ADDRESS OF FIRST TEST
1670	007126	005037	002004			CLR	RTNNO ; CLEAR ROUTINE NUMBER
1671	007132	000137	002600			JMP	GETROY ; GO AND START PROGRAM
1672						; *****	
1673	007136	000000				RTO.	0 ; ROUTINE # 0
1674	007140	007200				RT1	RT1 ; ADDR OF NEXT ROUTINE
1675	007142	000144				100.	100. ; ITERATION COUNT
1676	007144	007146				RT0A	RT0A ; SCOPE ENTRY POINT
1677		000000				X=X+1	X=X+1
1678						; *****	
1679						; TEST ABILITY TO REFERENCE CSR WITHOUT TRAPPING	
1680						RT0A:	MOV #15, @ERRVEC ; SET UP ERROR TRAP
1681	007146	012737	007170	000004		TST	@CSR ; REFERENCE CSR
1682	007154	005777	172362			MOV	#ERRVEC+2, @ERRVEC ; RESET TIME OUT TRAP
1683	007160	012737	000006	000004		SCOPE	
1684	007166	104006				15:	SUB #4, (6) ; RESTORE PC TO WHERE THE ILLEGAL
1685	007170	162716	000004				ERROR ; REFERENCE OCCURED
1686						RT1	RT1 ; ERROR! ILLEGAL REFERENCE OCCURED
1687	007174	104001					LOOP ILLEGAL REFERENCE INSTRUCTION
1688	007176	000002				; *****	
1689						RT1	1 ; ROUTINE # 1
1690	007200	000001				RT2	RT2 ; ADDR OF NEXT ROUTINE.
1691	007202	007252				100.	100. ; ITERATION COUNT
1692	007204	000144				RT1A	RT1A ; SCOPE ENTRY POINT.
1693	007206	007210				X=X+1	X=X+1
1694		000001				; *****	
1695						; TEST THAT CSR BIT0 CAN BE SET AND CLEARED	
1696						RT1A:	MOV #BIT0, @CSR ; SET BIT0.
1697	007210	012777	000001	172324		CMP	#BIT0, @CSR ; TEST THAT BIT0 IS SET
1698	007216	022777	000001	172316		BEQ	15 ; BRANCH IF SET
1699	007224	001402				ERROR	ERROR ; CSR BIT0 FAILED TO SET
1700	007226	104001				BR	25 ; OR AN ADDITIONAL BIT ALSO SET
1701	007230	000407				15:	BIC #BIT0, @CSR ; CLEAR BIT0
1702	007232	042777	000001	172302		TST	@CSR ; TEST THAT BIT0 IS CLEAR
1703	007232	042777	000001	172302		BEQ	25
1704	007240	005777	172276			ERROR	ERROR ; CSR BIT0 FAILED TO CLEAR
1705	007244	001401				25:	SCOPE
1706	007246	104001				; *****	
1707	007250	104006				RT2:	2 ; ROUTINE # 2
1708						RT3	RT3 ; ADDR OF NEXT ROUTINE.
1709	007252	000002				100.	100. ; ITERATION COUNT
1710	007254	007324				RT2A	RT2A ; SCOPE ENTRY POINT.
1711	007256	000144				X=X+1	X=X+1
1712	007260	007262				; *****	
1713		000002				; TEST THAT CSR BIT1 CAN BE SET AND CLEARED	
1714						RT2A:	MOV #BIT1, @CSR ; SET BIT1.
1715						CMP	#BIT1, @CSR ; TEST THAT BIT1 IS SET
1716	007262	012777	000002	172252			
1717	007270	022777	000002	172244			
1718							

```

1719 007276 001402      BEQ      15      ; BRANCH IF SET
1720 007300 104001      ERROR
1721 007302 000407      BR      25      ; CSR BIT1 FAILED TO SET
1722 007304 042777 000002 172230 15      BIC      #BIT1, @CSR ; OR AN ADDITIONAL BIT ALSO SET
1723 007312 005777 172224      TST      @CSR      ; CLEAR BIT1
1724 007316 001401      BEQ      25      ; TEST THAT BIT1 IS CLEAR
1725 007320 104001      ERROR
1726 007322 104006      ; CSR BIT1 FAILED TO CLEAR
1727      25:      SCOPE
1728 007324 000003      ; *****
1729 007326 007376      RT3:      3      ; ROUTINE # 3
1730 007330 000144      RT4      ; ADDR OF NEXT ROUTINE
1731 007332 007334      100.      ; ITERATION COUNT
1732      RT3A      ; SCOPE ENTRY POINT.
1733      X=X+1
1734      ; *****
1735      ; TEST THAT CSR BIT2 CAN BE SET AND CLEARED
1736 007334 012777 000004 172200 RT3A: MOV      @BIT2, @CSR ; SET BIT2
1737 007342 022777 000004 172172 CMP      @BIT2, @CSR ; TEST THAT BIT2 IS SET
1738 007350 001402      BEQ      15      ; BRANCH IF SET
1739 007352 104001      ERROR      ; CSR BIT2 FAILED TO SET
1740 007354 000407      BR      25      ; OR AN ADDITIONAL BIT ALSO SET
1741 007356 042777 000004 172156 15: BIC      @BIT2, @CSR ; CLEAR BIT2
1742 007364 005777 172152      TST      @CSR      ; TEST THAT BIT2 IS CLEAR
1743 007370 001401      BEQ      25      ; CSR BIT2 FAILED TO CLEAR
1744 007372 104001      ERROR
1745 007374 104006      25:      SCOPE
1746      ; *****
1747 007376 000004      RT4:      4      ; ROUTINE # 4
1748 007400 007450      RT5      ; ADDR OF NEXT ROUTINE
1749 007402 000144      100.      ; ITERATION COUNT
1750 007404 007406      RT4A      ; SCOPE ENTRY POINT.
1751      X=X+1
1752      ; *****
1753      ; TEST THAT CSR BIT4 CAN BE SET AND CLEARED
1754      RT4A: MOV      @BIT4, @CSR ; SET BIT4
1755 007406 012777 000020 172126 CMP      @BIT4, @CSR ; TEST THAT BIT4 IS SET
1756 007414 022777 000020 172120 BEQ      15      ; BRANCH IF SET
1757 007422 001402      ERROR      ; CSR BIT4 FAILED TO SET
1758 007424 104001      BR      25      ; OR AN ADDITIONAL BIT ALSO SET
1759 007426 000407      BIC      @BIT4, @CSR ; CLEAR BIT4
1760 007430 042777 000020 172104 15: TST      @CSR      ; TEST THAT BIT4 IS CLEAR
1761 007436 005777 172100      BEQ      25      ; CSR BIT4 FAILED TO CLEAR
1762 007442 001401      ERROR
1763 007444 104001      25:      SCOPE
1764 007446 104006      ; *****
1765      RT5:      5      ; ROUTINE # 5
1766 007450 000005      RT6      ; ADDR OF NEXT ROUTINE
1767 007452 007522      100.      ; ITERATION COUNT
1768 007454 000144      RT5A      ; SCOPE ENTRY POINT.
1769 007456 007460      X=X+1
1770      ; *****
1771      ; TEST THAT CSR BITS CAN BE SET AND CLEARED
1772      RT5A: MOV      @BIT5, @CSR ; SET BITS.
1773      1774 007460 012777 000040 172054

```

```
1775 007466 022777 000040 172046      CMP      #BIT5, @CSR      ; TEST THAT BIT5 IS SET
1776 007474 001402                      BEQ      15              ; BRANCH IF SET
1777 007476 104001                      ERROR                      ; CSR BIT5 FAILED TO SET
1778 007500 000407                      BR       25              ; OR AN ADDITIONAL BIT ALSO SET
1779 007502 042777 000040 172032 15:    BIC      #BIT5, @CSR      ; CLEAR BIT5
1780 007510 005777 172026                      TST      @CSR          ; TEST THAT BIT5 IS CLEAR
1781 007514 001401                      BEQ      25              ; CSR BIT5 FAILED TO CLEAR
1782 007516 104001                      ERROR
1783 007520 104006                      25:      SCOPE
1784                                     ; *****
1785 007522 000006                      RT6:      6              ; ROUTINE # 6
1786 007524 007574                      RT7      100.            ; ADDR OF NEXT ROUTINE
1787 007526 000144                      RT6A     100.            ; ITERATION COUNT
1788 007530 007532                      X=X+1      ; SCOPE ENTRY POINT.
1789 000006
1790                                     ; *****
1791
1792                                     ; TEST THAT CSR BIT6 CAN BE SET AND CLEARED
1793 007532 012777 000100 172002 RT6A:    MOV      #BIT6, @CSR      ; SET BIT6.
1794 007540 022777 000100 171774          CMP      #BIT6, @CSR      ; TEST THAT BIT6 IS SET
1795 007546 001402                      BEQ      15              ; BRANCH IF SET
1796 007550 104001                      ERROR                      ; CSR BIT6 FAILED TO SET
1797 007552 000407                      BR       25              ; OR AN ADDITIONAL BIT ALSO SET
1798 007554 042777 000100 171760 15:    BIC      #BIT6, @CSR      ; CLEAR BIT6
1799 007562 005777 171754          TST      @CSR          ; TEST THAT BIT6 IS CLEAR
1800 007566 001401                      BEQ      25              ; CSR BIT6 FAILED TO CLEAR
1801 007570 104001                      ERROR
1802 007572 104006                      25:      SCOPE
1803                                     ; *****
1804 007574 000007                      RT7:      7              ; ROUTINE # 7
1805 007576 007646                      RT10     100.            ; ADDR OF NEXT ROUTINE
1806 007600 000144                      RT7A     100.            ; ITERATION COUNT
1807 007602 007604                      X=X+1      ; SCOPE ENTRY POINT.
1808 000007
1809                                     ; *****
1810
1811                                     ; TEST THAT CSR BIT12 CAN BE SET AND CLEARED
1812 007604 012777 010000 171730 RT7A:    MOV      #BIT12, @CSR     ; SET BIT12.
1813 007612 022777 010000 171722          CMP      #BIT12, @CSR     ; TEST THAT BIT12 IS SET
1814 007620 001402                      BEQ      15              ; BRANCH IF SET
1815 007622 104001                      ERROR                      ; CSR BIT12 FAILED TO SET
1816 007624 000407                      BR       25              ; OR AN ADDITIONAL BIT ALSO SET
1817 007626 042777 010000 171706 15:    BIC      #BIT12, @CSR     ; CLEAR BIT12
1818 007634 005777 171702          TST      @CSR          ; TEST THAT BIT12 IS CLEAR
1819 007640 001401                      BEQ      25              ; CSR BIT12 FAILED TO CLEAR
1820 007642 104001                      ERROR
1821 007644 104006                      25:      SCOPE
1822                                     ; *****
1823 007646 000010                      RT10:     10              ; ROUTINE # 10
1824 007650 007720                      RT11     100.            ; ADDR OF NEXT ROUTINE
1825 007652 000144                      RT10A    100.            ; ITERATION COUNT
1826 007654 007656                      X=X+1      ; SCOPE ENTRY POINT.
1827 000010
1828                                     ; *****
1829
1830                                     ; TEST THAT CSR BIT13 CAN BE SET AND CLEARED
```

```

1831 007656 012777 020000 171656 RT10A: MOV #BIT13,ACSR ;SET BIT13.
1832 007664 022777 020000 171650 CMP #BIT13,ACSR ;TEST THAT BIT13 IS SET
1833 007672 001402 BEQ 15 ;BRANCH IF SET
1834 007674 104001 ERROR ;CSR BIT13 FAILED TO SET
1835 007676 000407 BR 25 ;OR AN ADDITIONAL BIT ALSO SET
1836 007700 042777 020000 171634 15: BIC #BIT13,ACSR ;CLEAR BIT13
1837 007706 005777 171630 TST ACSR ;TEST THAT BIT13 IS CLEAR
1838 007712 001401 BEQ 25
1839 007714 104001 ERROR ;CSR BIT13 FAILED TO CLEAR
1840 007716 104006 25: SCOPE
1841 *****
1842 007720 000011 RT11: 11 ;ROUTINE # 11
1843 007722 010010 RT12 ;ADDR OF NEXT ROUTINE
1844 007724 000144 100. ;ITERATION COUNT
1845 007726 007730 RT11A ;SCOPE ENTRY POINT
1846 000011 X=X+1
1847 *****
1848 ;TEST THAT RESET & CLEAR INSTRUCTION CLEAR ALL R/W BITS IN THE CONTROL
1849 ; STATUS REG (CSR)
1850 RT11A: MOV #30167,ACSR ;SET ALL R/W BITS IN THE CSR
1851 007730 012777 030167 171604 CLR XMTDAT
1852 007736 005037 001570 SRESET ;ISSUE RESET
1853 007742 104005 MOV ACSR,RCV DAT ;GET CSR CONTENTS
1854 007744 017737 171572 001566 BEQ 15 ;BRANCH IF RESET CLEARED ALL BITS
1855 007752 001402 ERROR1 ;ERROR! RESET DID NOT CLEAR ALL BITS
1856 007754 104011 BR RT11A ;LOOP ON ERROR
1857 007756 000764 15: MOV #30167,ACSR ;SET ALL R/W BITS IN CSR
1858 007760 012777 030167 171554 15: CLR ACSR ;CLEAR THE CSR
1859 007766 005077 171550 001566 MOV ACSR,RCV DAT ;GET & TEST CSR
1860 007772 017737 171544 001566 BEQ 25 ;GO TO EXIT IF RESULT = 0
1861 010000 001402 ERROR1 ;ERROR! CLEAR INST. DID NOT CLEAR ALL BITS
1862 010002 104011 BR 15 ;LOOP ERROR
1863 010004 000765 25: SCOPE ;SCOPE
1864 010006 104006 *****
1865 RT12: 12 ;ROUTINE # 12
1866 010010 000012 RT13 ;ADDR OF NEXT ROUTINE
1867 010012 010144 10. ;ITERATION COUNT
1868 010014 000012 RT12A ;SCOPE ENTRY POINT
1869 010016 010020 X=X+1
1870 000012 *****
1871 ;TEST THAT A BINARY COUNT CAN BE LOADED INTO A CLEAR BKCSR AND THAT
1872 ; A BINARY COUNT CAN BE CLEARED.
1873 RT12A: CLR XMTDAT
1874 15: MOV XMTDAT,BKCSR ;LOAD BINARY COUNT INTO BKCSR
1875 010020 005037 001570 MOV BKCSR,RCV DAT ;GET BKCSR DATA
1876 010024 013777 001570 171514 CMP XMTDAT,RCV DAT ;COMPARE DATA LOADED & DATA READ BACK
1877 010032 017737 171510 001566 BEQ 25 ;BRANCH IF DATA COMPARES
1878 010040 023737 001570 001566 ERROR1 ;ERROR! DATA DID NOT COMPARE
1879 010046 001405 BIT #BIT14,ASHR ;SCOPE LOOP?
1880 010050 104011 BNE 15 ;BRANCH IF SCOPE LOOP
1881 010052 032777 040000 171122 25: MOV XMTDAT,XI ;SAVE BINARY COUNT
1882 010060 001361 CLR XMTDAT
1883 010062 013701 001570 35: CLR BKCSR ;CLEAR BKCSR AND TEST
1884 010066 005037 001570 MOV BKCSR,RCV DAT ;BKCSR CAN BE CLEARED
1885 010072 005077 171450
1886 010076 017737 171444 001566

```

1887	010104	001405				BEQ	45		; BRANCH IF BKCSR CLEARED
1888	010106	104011				ERROR1			; ERROR! BKCSR DID NOT CLEAR
1889	010110	032777	040000	171064		BIT	8BIT14, 2SWR		; SCOPE LOOP?
1890	010116	001365				BNE	35		; BRANCH IF SCOPE LOOP
1891	010120	010137	001570		45:	MOV	X1, XMTDAT		; GET BINARY COUNT
1892	010124	023727	001570	177777		CMP	XMTDAT, #1		; ALL NUMBERS BEEN LOADED
1893	010132	001403				BEQ	55		; GO TO EXIT
1894	010134	005237	001570			INC	XMTDAT		; INCREMENT BINARY COUNT
1895	010140	000731				BR	15		; REPEAT TEST
1896	010142	104006				55:	SCOPE		; SCOPE
1897						; *****			
1898	010144	000013				RT13:	13		; ROUTINE # 13
1899	010146	010256					RT14		; ADDR OF NEXT ROUTINE.
1900	010150	000144					100.		; ITERATION COUNT
1901	010152	010154					RT13A		; SCOPE ENTRY POINT.
1902		000013					X=X+1		
1903						; *****			
1904						; TEST THAT RANDOM NUMBERS CAN BE LOADED INTO THE BKCSR			
1905						RT13A:	MOV	#10000, X2	; GET RANDOM COUNTER
1906	010154	012702	010000			15:	MOV	2BKCSR, PRVCNT	; GET PREVIOUS CONTENTS
1907	010160	017737	171362	002104			JSR	7, 2BRNGEN	; GO GET A RANDOM NUMBER
1908	010166	004737	003250				MOV	X0, XMTDAT	; GET RANDOM NUMBER
1909	010172	010037	001570			25:	MOV	XMTDAT, 2BKCSR	; LOAD RANDOM NUMBER INTO BKCSR
1910	010176	013777	001570	171342			MOV	2BKCSR, RCVDAT	; GET BKCSR DATA
1911	010204	017737	171336	001566			CMP	XMTDAT, RCVDAT	; COMPARE DATA
1912	010212	023737	001570	001566			BEQ	35	; BRANCH IF SAME
1913	010220	001401					ERROR1		; ERROR! DATA NOT THE SAME
1914	010222	104011				35:	BIT	8BIT14, 2SWR	; SCOPE LOOP?
1915	010224	032777	040000	170750			BEQ	45	; BRANCH IF NO LOOP ON ERROR
1916	010232	001406					CLR	2BKCSR	
1917	010234	005077	171306				MOV	PRVCNT, 2BKCSR	; LOAD PREVIOUS CONTENTS
1918	010240	013777	002104	171300			BR	25	; REPEAT TEST
1919	010246	000753				45:	DEC	X2	
1920	010250	005302					BNE	15	; BRANCH IF NOT
1921	010252	001342				55:	SCOPE		; SCOPE
1922	010254	104006				; *****			
1923						RT14:	14		; ROUTINE # 14
1924	010256	000014					RT15		; ADDR OF NEXT ROUTINE.
1925	010260	010316					10.		; ITERATION COUNT
1926	010262	000012					RT14A		; SCOPE ENTRY POINT.
1927	010264	010266					X=X+1		
1928		000014				; *****			
1929						; TEST THAT RESET CLEARS ALL BREAK STATUS REGISTER BITS			
1930						RT14A:	MOV	#-1, 2BKCSR	
1931							CLR	XMTDAT	
1932							SRESET		
1933	010266	012777	177777	171252			MOV	2BKCSR, RCVDAT	
1934	010274	005037	001570				BEQ	15	
1935	010300	104005				15:	SCOPE		
1936	010302	017737	171240	001566		; *****			
1937	010310	001401				RT15:	15		; ROUTINE # 15
1938	010312	104011					RT16		; ADDR OF NEXT ROUTINE
1939	010314	104006							
1940									
1941	010316	000015							
1942	010320	010452							

Line	Address	Hex	Label	Comment
1943	010322	000012		
1944	010324	010326		
1945		000015		
1946				
1947				
1948				
1949				
1950	010326	005037	001570	
1951	010332	013777	001570	171210
1952	010340	017737	171204	001566
1953	010346	023737	001570	001566
1954	010354	001405		
1955	010356	104011		
1956	010360	032777	040000	170614
1957	010366	001361		
1958	010370	013701	001570	
1959	010374	005037	001570	
1960	010400	005077	171144	
1961	010404	017737	171140	001566
1962	010412	001405		
1963	010414	104011		
1964	010416	032777	040000	170556
1965	010424	001365		
1966	010426	010137	001570	
1967	010432	023727	001570	177000
1968	010440	001403		
1969	010442	105237	001571	
1970	010446	000731		
1971	010450	104006		
1972				
1973	010452	000016		
1974	010454	010570		
1975	010456	000144		
1976	010460	010462		
1977		000016		
1978				
1979				
1980				
1981	010462	012702	010000	
1982	010466	017737	171056	002104
1983	010474	004737	003250	
1984	010500	042700	000377	
1985	010504	010037	001570	
1986	010510	013777	001570	171032
1987	010516	017737	171026	001566
1988	010524	023737	001570	001566
1989	010532	001401		
1990	010534	104011		
1991	010536	032777	040000	170436
1992	010544	001406		
1993	010546	005077	170776	
1994	010552	013777	002104	170770
1995	010560	000753		
1996	010562	005302		
1997	010564	001340		
1998	010566	104006		

```

1999
2000 010570 000017 ;*****
2001 010572 010662 RT17: 17 ;ROUTINE # 17
2002 010574 000144 RT20 ;ADDR OF NEXT ROUTINE
2003 010576 010600 100. ;ITERATION COUNT
2004 000017 RT17A ;SCOPE ENTRY POINT.
2005 X=X+1
2006 ;*****
2007 ;TEST THAT ALL BAR BITS CAN BE INDIVIDUALLY SET AND CLEARED
2008 010600 013701 001544 RT17A: MOV BAR,X1 ;GET BAR ADDRESS
2009 010604 012777 001400 170736 MOV BCAT,2BASREG ;INITIALIZE BASE REGISTER
2010 010612 012700 000001 MOV #1,X0 ;GET BIT TESTER
2011 010616 050011 15 BIS X0,(1) ;SET BAR BIT
2012 010620 020011 CMP X0,(1) ;TEST THAT ONLY THE PROPER BAR BIT SET
2013 010622 001006 BNE 35 ;BRANCH IF ERROR
2014 010624 040011 BIC X0,(1) ;CLEAR BAR BIT
2015 010626 005711 TST (1) ;TEST THAT BAR BIT CLEARED
2016 010630 001011 BNE 55 ;BRANCH IF BAR BIT FAILED TO CLEAR
2017 010632 006300 ASL X0 ;SHIFT BIT TESTER
2018 010634 103370 BCC 15
2019 010636 104006 25: SCOPE ;SCOPE
2020 010640 010037 001570 35: MOV X0,XMTDAT ;GET WHAT DATA WAS SUPPOSED TO BE
2021 010644 011137 001566 45: MOV (1),RCVDAT ;GET WHAT DATA WAS
2022 010650 104011 ERROR1 ;ERROR! IMPROPER BIT OPERATION
2023 010652 000771 BR 25 ;GO TO SCOPE
2024 010654 005037 001570 55: CLR XMTDAT ;GET WHAT DATA WAS SUPPOSED TO BE
2025 010660 000771 BR 45
2026 ;*****
2027 010662 000020 RT20: 20 ;ROUTINE # 20
2028 010664 010750 RT21 ;ADDR OF NEXT ROUTINE.
2029 010666 000012 10. ;ITERATION COUNT
2030 010670 010672 RT20A ;SCOPE ENTRY POINT.
2031 000020 X=X+1
2032 ;*****
2033 ;TEST THAT RESET CLEARS ALL BAR BITS
2034 RT20A: CLR XMTDAT
2035 010672 005037 001570 BIS #1,2BAR ;SET ALL BAR BITS
2036 010676 052777 177777 170640 SRESET ;RESET
2037 010704 104005 MOV 2BAR,RCVDAT ;GET BAR DATA
2038 010706 017737 170632 001566 BEQ 15 ;BRANCH IF ALL 0'S
2039 010714 001402 ERROR1 ;ERROR! RESET DID NOT CLEAR ALL BAR BITS
2040 010716 104011 BR 25 ;GO TO EXIT
2041 010720 000412 15: BIS #1,2BAR ;SET ALL BIT IN THE BAR
2042 010722 052777 177777 170614 CLR 2BAR ;CLEAR ALL BITS IN THE BAR
2043 010730 005077 170610 001566 MOV 2BAR,RCVDAT ;GET & TEST RESULT OF CLEAR OPERATION
2044 010734 017737 BEQ 25 ;EXIT IF ALL BITS CLEARED
2045 010742 001401 ERROR1 ;ERROR! ALL BITS DID NOT CLEAR
2046 010744 104011 25: SCOPE ;SCOPE
2047 010746 104006 ;*****
2048 RT21: 21 ;ROUTINE # 21
2049 010750 000021 RT22 ;ADDR OF NEXT ROUTINE.
2050 010752 011056 100. ;ITERATION COUNT
2051 010754 000144 RT21A ;SCOPE ENTRY POINT.
2052 010756 010760 X=X+1
2053 000021 ;*****
2054

```

```

2055
2056
2057 010760 012777 010100 170554 ;TEST THAT CSR RESPONDS PROPERLY TO BYTE COMMANDS
2058 010766 105077 170550 RT21A: MOV #10100,CSR ;LOAD TEST NUMBER IN CSR
2059 010772 022777 010000 170542 CLRB CSR ;CLEAR EVEN BYTE
2060 011000 001410 BEQ 15 ;TEST THAT ONLY EVEN BYTE CLEARED
2061 011002 012737 010100 001570 MOV #10100,XMTDAT ;LOAD CORRECT RESULT
2062 011010 017737 170526 001566 MOV CSR,RCVDAT ;GET ACTUAL RESULT
2063 011016 104011 ERROR1 ;ERROR! EVEN BYTE INSTRUCTION FAILED
2064 011020 000415 BR 25 ;GO TO SCOPE
2065 011022 012777 010100 170512 15: MOV #10100,CSR ;LOAD TEST NUMBER IN CSR
2066 011030 105077 171052 CLRB CSR ;TEST THAT ONLY ODD BYTE CLEARED
2067 011034 001407 BEQ 25
2068 011036 012737 000100 001570 MOV #00100,XMTDAT ;LOAD CORRECT RESULT
2069 011044 017737 170472 001566 MOV CSR,RCVDAT ;LOAD ACTUAL RESULT
2070 011052 104011 ERROR1 ;ERROR! ODD BYTE INSTRUCTION FAILED
2071 011054 104006 25: SCOPE ;SCOPE
2072 ;*****
2073 011056 000022 RT22: 22 ;ROUTINE # 22
2074 011060 011164 RT23 ;ADDR OF NEXT ROUTINE
2075 011062 000144 100. ;ITERATION COUNT
2076 011064 011066 RT22A ;SCOPE ENTRY POINT.
2077 000022 X=X+1
2078 ;*****
2079
2080 ;TEST THAT BAR RESPONDS PROPERLY TO BYTE COMMANDS
2081 011066 012777 010100 170450 RT22A: MOV #10100,BAR ;LOAD TEST NUMBER IN BAR
2082 011074 105077 170444 CLRB BAR ;CLEAR EVEN BYTE
2083 011100 022777 010000 170436 CMP #10000,BAR ;TEST THAT ONLY EVEN BYTE CLEARED
2084 011106 001410 BEQ 15
2085 011110 012737 010100 001570 MOV #10100,XMTDAT ;LOAD CORRECT RESULT
2086 011116 017737 170422 001566 MOV BAR,RCVDAT ;GET ACTUAL RESULT
2087 011124 104011 ERROR1 ;ERROR! EVEN BYTE INSTRUCTION FAILED
2088 011126 000415 BR 25 ;GO TO SCOPE
2089 011130 012777 010100 170406 15: MOV #10100,BAR ;LOAD TEST NUMBER IN BAR
2090 011136 105077 170746 CLRB BAR ;TEST THAT ONLY ODD BYTE CLEARED
2091 011142 001407 BEQ 25
2092 011144 012737 000100 001570 MOV #00100,XMTDAT ;LOAD CORRECT RESULT
2093 011152 017737 170364 001566 MOV CSR,RCVDAT ;LOAD ACTUAL RESULT
2094 011160 104011 ERROR1 ;ERROR! ODD BYTE INSTRUCTION FAILED
2095 011162 104006 25: SCOPE ;SCOPE
2096 ;*****
2097 011164 000023 RT23: 23 ;ROUTINE # 23
2098 011166 011272 RT24 ;ADDR OF NEXT ROUTINE.
2099 011170 000144 100. ;ITERATION COUNT
2100 011172 011174 RT23A ;SCOPE ENTRY POINT.
2101 000023 X=X+1
2102 ;*****
2103
2104 ;TEST THAT BKCSR RESPONDS PROPERLY TO BYTE COMMANDS
2105 011174 012777 010100 170344 RT23A: MOV #10100,BKCSR ;LOAD TEST NUMBER IN BKCSR
2106 011202 105077 170340 CLRB BKCSR ;CLEAR EVEN BYTE
2107 011206 022777 010000 170332 CMP #10000,BKCSR ;TEST THAT ONLY EVEN BYTE CLEARED
2108 011214 001410 BEQ 15
2109 011216 012737 010100 001570 MOV #10100,XMTDAT ;LOAD CORRECT RESULT
2110 011224 017737 170316 001566 MOV BKCSR,RCVDAT ;GET ACTUAL RESULT

```

```
2111 011232 104011 ERROR1  
2112 011234 000415 BR 25 ;ERROR! EVEN BYTE INSTRUCTION FAILED  
2113 011236 012777 010100 170302 15: MOV #10100,2BKCSR ;GO TO SCOPE  
2114 011244 105077 170642 CLR B 2, BKCSR ;LOAD TEST NUMBER IN BKCSR  
2115 011250 001407 BEQ 25 ;TEST THAT ONLY ODD BYTE CLEARED  
2116 011252 012737 000100 001570 MOV #00100,XMTDAT ;LOAD CORRECT RESULT  
2117 011260 017737 170256 001566 MOV #CSR,RCVDAT ;LOAD ACTUAL RESULT  
2118 011266 104011 ERROR1  
2119 011270 104006 25: SCOPE ;ERROR! ODD BYTE INSTRUCTION FAILED  
2120 ;*****  
2121 011272 000024 RT24: 24 ;ROUTINE # 24 *  
2122 011274 011336 RT25 ;ADDR OF NEXT ROUTINE. *  
2123 011276 000144 100 ;ITERATION COUNT *  
2124 011300 011302 RT24A ;SCOPE ENTRY POINT. *  
2125 000024 X=X+1  
2126 ;*****  
2127  
2128 ;TEST THAT OVER RUN BIT (CSR BIT13) CAUSES AN INTERRUPT WHEN SET  
2129 011302 012777 011334 170246 RT24A: MOV #15,2XMTINT ;LOAD TRANSMITTER INTERRUPT VECTOR  
2130 011310 012777 010000 170224 MOV #BIT12,2CSR ;SET TRANSMITTER IE BIT  
2131 011316 052777 020000 170216 BIS #BIT13,2CSR ;SET OVER RUN BIT  
2132 011324 005037 177776 CLR 2APSW ;ENABLE INTERRUPTS  
2133 011330 000240 NOP  
2134 011332 104001 ERROR  
2135 ;ERROR!OVERRUN FAILED TO CAUSE AN  
2136 ;INTERRUPT,OR INTERRUPTED TO INCOR-  
2137 011334 104006 15: SCOPE ;RECT ADDRESS  
2138 ;*****  
2139 011336 000025 RT25: 25 ;ROUTINE # 25 *  
2140 011340 011444 RT26 ;ADDR OF NEXT ROUTINE *  
2141 011342 000100 100 ;ITERATION COUNT *  
2142 011344 011346 RT25A ;SCOPE ENTRY POINT *  
2143 000025 X=X+1  
2144 ;*****  
2145  
2146 ;TEST THAT THE DM11 INTERRUPTS AT THE CORRECT LEVEL  
2147 011346 012737 000340 177776 RT25A: MOV #PRTY7,2APSW  
2148 011354 012777 011404 170174 MOV #15,2XMTINT ;LOAD TRANSMITTER INTERRUPT VECTOR  
2149 011362 012777 030000 170152 MOV #30000,2CSR ;SET OVER RUN & IE BITS  
2150 011370 012737 000200 177776 MOV #PRTY4,2APSW ;ALLOW INTERRUPTS ON LEVEL 5 & ABOVE  
2151 011376 000240 NOP  
2152 011400 104001 ERROR  
2153 011402 000417 BR 35 ;ERROR!DM11 FAILED TO INTERRUPT  
2154 011404 022626 15: CMP (6)+,(6)+ ;GO TO EXIT  
2155 011406 013737 001560 177776 MOV XMTLVL,2APSW ;RESET STACK POINTER  
2156 011414 012777 011440 170134 MOV #25,2XMTINT ;LOAD DM11 INTERRUPT LEVEL  
2157 011422 005077 170114 CLR 2CSR ;LOAD TRANSMITTER INTERRUPT VECTOR  
2158 011426 012777 030000 170106 MOV #30000,2CSR  
2159 011434 000240 NOP  
2160 011436 000401 BR 35 ;GO TO EXIT  
2161 011440 104001 25: ERROR ;ERROR! DM11 INTERRUPTED ON HIGHER  
2162 ;PRIORITY LEVEL THAN SET FOR  
2163 011442 104006 35: SCOPE  
2164 ;*****  
2165 011444 000026 RT26: 26 ;ROUTINE # 26 *  
2166 011446 011462 RT27 ;ADDRESS OF NEXT TEST. *
```

2167	011450	000144		100.		; ITERATION COUNT	*
2168	011452	011454		LTST0		; SCOPE ENTRY POINT	*
2169		000026		X=X+1			
2170				;*****			
2171				; TRANSMITTER LINE TEST LINE 0			
2172	011454	004537	004622	LTST0: JSR	5, XMTTST	; GO TEST TRANSMITTER LINE 0	
2173	011460	000000		LINE0			
2174		000001		Y=Y+1			
2175				;*****			
2176	011462	000027		RT27:	27	; ROUTINE # 27	*
2177	011464	C11500		RT30		; ADDRESS OF NEXT TEST.	*
2178	011466	000144		100.		; ITERATION COUNT	*
2179	011470	011472		LTST1		; SCOPE ENTRY POINT	*
2180		000027		X=X+1			
2181				;*****			
2182				; TRANSMITTER LINE TEST LINE 1			
2183	011472	004537	004622	LTST1: JSR	5, XMTTST	; GO TEST TRANSMITTER LINE 1	
2184	011476	000002		LINE1			
2185		000002		Y=Y+1			
2186				;*****			
2187	011500	000030		RT30:	30	; ROUTINE # 30	*
2188	011502	011516		RT31		; ADDRESS OF NEXT TEST.	*
2189	011504	000144		100.		; ITERATION COUNT	*
2190	011506	011510		LTST2		; SCOPE ENTRY POINT	*
2191		000030		X=X+1			
2192				;*****			
2193				; TRANSMITTER LINE TEST LINE 2			
2194	011510	004537	004622	LTST2: JSR	5, XMTTST	; GO TEST TRANSMITTER LINE 2	
2195	011514	000004		LINE2			
2196		000003		Y=Y+1			
2197				;*****			
2198	011516	000031		RT31:	31	; ROUTINE # 31	*
2199	011520	011534		RT32		; ADDRESS OF NEXT TEST.	*
2200	011522	000144		100.		; ITERATION COUNT	*
2201	011524	011526		LTST3		; SCOPE ENTRY POINT	*
2202		000031		X=X+1			
2203				;*****			
2204				; TRANSMITTER LINE TEST LINE 3			
2205	011526	004537	004622	LTST3: JSR	5, XMTTST	; GO TEST TRANSMITTER LINE 3	
2206	011532	000006		LINE3			
2207		000004		Y=Y+1			
2208				;*****			
2209	011534	000032		RT32:	32	; ROUTINE # 32	*
2210	011536	011552		RT33		; ADDRESS OF NEXT TEST	*
2211	011540	000144		100.		; ITERATION COUNT	*
2212	011542	011544		LTST4		; SCOPE ENTRY POINT	*
2213		000032		X=X+1			
2214				;*****			
2215				; TRANSMITTER LINE TEST LINE 4			
2216	011544	004537	004622	LTST4: JSR	5, XMTTST	; GO TEST TRANSMITTER LINE 4	
2217	011550	000010		LINE4			
2218		000005		Y=Y+1			
2219				;*****			
2220	011552	000033		RT33:	33	; ROUTINE # 33	*
2221	011554	011570		RT34		; ADDRESS OF NEXT TEST.	*
2222	011556	000144		100.		; ITERATION COUNT	*

2223	011560	011562		LTST5		; SCOPE ENTRY POINT	*
2224		000033		X=X+1			
2225				;*****			
2226				; TRANSMITTER LINE TEST LINE 5			
2227	011562	004537	004622	LTST5: JSR	5, XMTTST	; GO TEST TRANSMITTER LINE 5	
2228	011566	000012		LINE5			
2229		000006		Y=Y+1			
2230				;*****			
2231	011570	000034		RT34:	34	; ROUTINE # 34	*
2232	011572	011606		RT35		; ADDRESS OF NEXT TEST	*
2233	011574	000144		100.		; ITERATION COUNT	*
2234	011576	011600		LTST6		; SCOPE ENTRY POINT	*
2235		000034		X=X+1			
2236				;*****			
2237				; TRANSMITTER LINE TEST LINE 6			
2238	011600	004537	004622	LTST6: JSR	5, XMTTST	; GO TEST TRANSMITTER LINE 6	
2239	011604	000011		LINE6			
2240		000007		Y=Y+1			
2241				;*****			
2242	011606	000035		RT35:	35	; ROUTINE # 35	*
2243	011610	011624		RT36		; ADDRESS OF NEXT TEST	*
2244	011612	000144		100.		; ITERATION COUNT	*
2245	011614	011616		LTST7		; SCOPE ENTRY POINT	*
2246		000035		X=X+1			
2247				;*****			
2248				; TRANSMITTER LINE TEST LINE 7			
2249	011616	004537	004622	LTST7: JSR	5, XMTTST	; GO TEST TRANSMITTER LINE 7	
2250	011622	000016		LINE7			
2251		000010		Y=Y+1			
2252				;*****			
2253	011624	000036		RT36:	36	; ROUTINE # 36	*
2254	011626	011642		RT37		; ADDRESS OF NEXT TEST	*
2255	011630	000144		100.		; ITERATION COUNT	*
2256	011632	011634		LTST10		; SCOPE ENTRY POINT	*
2257		000036		X=X+1			
2258				;*****			
2259				; TRANSMITTER LINE TEST LINE 10			
2260	011634	004537	004622	LTST10: JSR	5, XMTTST	; GO TEST TRANSMITTER LINE 10	
2261	011640	000020		LINE10			
2262		000011		Y=Y+1			
2263				;*****			
2264	011642	000037		RT37:	37	; ROUTINE # 37	*
2265	011644	011660		RT40		; ADDRESS OF NEXT TEST	*
2266	011646	000144		100.		; ITERATION COUNT	*
2267	011650	011652		LTST11		; SCOPE ENTRY POINT	*
2268		000037		X=X+1			
2269				;*****			
2270				; TRANSMITTER LINE TEST LINE 11			
2271	011652	004537	004622	LTST11: JSR	5, XMTTST	; GO TEST TRANSMITTER LINE 11	
2272	011656	000022		LINE11			
2273		000012		Y=Y+1			
2274				;*****			
2275	011660	000040		RT40:	40	; ROUTINE # 40	*
2276	011662	011676		RT41		; ADDRESS OF NEXT TEST	*
2277	011664	000144		100.		; ITERATION COUNT	*
2278	011666	011670		LTST12		; SCOPE ENTRY POINT	*

```
2279      000040
2280
2281      X=X+1
2282 011670 004537 004622 ;*****
2283 011674 000024 ;TRANSMITTER LINE TEST LINE 12
2284      000013 ;LTST12: JSR 5,XMTTST ;GO TEST TRANSMITTER LINE 12
2285      Y=Y+1
2286 011676 000041 ;*****
2287 011700 011714 RT41: 41 ;ROUTINE # 41
2288 011702 000144 ;ADDRESS OF NEXT TEST.
2289 011704 011706 ;ITERATION COUNT
2290      000041 ;SCOPE ENTRY POINT
2291
2292 ;*****
2293 011706 004537 004622 ;TRANSMITTER LINE TEST LINE 13
2294 011712 000026 ;LTST13: JSR 5,XMTTST ;GO TEST TRANSMITTER LINE 13
2295      000014 ;LINE13
2296      Y=Y+1
2297 011714 000042 ;*****
2298 011716 011732 RT42: 42 ;ROUTINE # 42
2299 011720 000144 ;ADDRESS OF NEXT TEST.
2300 011722 011724 ;ITERATION COUNT
2301      000042 ;SCOPE ENTRY POINT
2302      X=X+1
2303 ;*****
2304 011724 004537 004622 ;TRANSMITTER LINE TEST LINE 14
2305 011730 000030 ;LTST14: JSR 5,XMTTST ;GO TEST TRANSMITTER LINE 14
2306      000015 ;LINE14
2307      Y=Y+1
2308 011732 000043 ;*****
2309 011734 011750 RT43: 43 ;ROUTINE # 43
2310 011736 000144 ;ADDRESS OF NEXT TEST.
2311 011740 011742 ;ITERATION COUNT
2312      000043 ;SCOPE ENTRY POINT
2313      X=X+1
2314 ;*****
2315 011742 004537 004622 ;TRANSMITTER LINE TEST LINE 15
2316 011746 000032 ;LTST15: JSR 5,XMTTST ;GO TEST TRANSMITTER LINE 15
2317      000016 ;LINE15
2318      Y=Y+1
2319 011750 000044 ;*****
2320 011752 011766 RT44: 44 ;ROUTINE # 44
2321 011754 000144 ;ADDRESS OF NEXT TEST.
2322 011756 011760 ;ITERATION COUNT
2323      000044 ;SCOPE ENTRY POINT
2324      X=X+1
2325 ;*****
2326 011760 004537 004622 ;TRANSMITTER LINE TEST LINE 16
2327 011764 000034 ;LTST16: JSR 5,XMTTST ;GO TEST TRANSMITTER LINE 16
2328      000017 ;LINE16
2329      Y=Y+1
2330 011766 000045 ;*****
2331 011770 012004 RT45: 45 ;ROUTINE # 45
2332 011772 000144 ;ADDRESS OF NEXT TEST.
2333 011774 011776 ;ITERATION COUNT
2334      000045 ;SCOPE ENTRY POINT
2335      X=X+1
```

```
2335  
2336  
2337 011776 004537 004622  
2338 012002 000036  
2339 000020  
2340 000000  
2341 000000  
2342  
2343 012004 000046  
2344 012006 012022  
2345 012010 000144  
2346 012012 012014  
2347 000046  
2348  
2349  
2350 012014 004537 005034  
2351 012020 000000  
2352 000001  
2353  
2354 012022 000047  
2355 012024 012040  
2356 012026 000144  
2357 012030 012032  
2358 000047  
2359  
2360  
2361 012032 004537 005034  
2362 012036 000002  
2363 000002  
2364  
2365 012040 000050  
2366 012042 012056  
2367 012044 000144  
2368 012046 012050  
2369 000050  
2370  
2371  
2372 012050 004537 005034  
2373 012054 000004  
2374 000003  
2375  
2376 012056 000051  
2377 012060 012074  
2378 012062 000144  
2379 012064 012066  
2380 000051  
2381  
2382  
2383 012066 004537 005034  
2384 012072 000006  
2385 000004  
2386  
2387 012074 000052  
2388 012076 012112  
2389 012100 000144  
2390 012102 012104
```

```
*****  
; TRANSMITTER LINE TEST LINE 17  
LTST17: JSR 5,XMTTST ; GO TEST TRANSMITTER LINE 17  
LINE17  
Y=Y+1  
Y=0  
A=0  
*****  
RT46: 46 ; ROUTINE # 46  
RT47 ; ADDRESS OF NEXT TEST  
100. ; ITERATION COUNT  
RCV0 ; SCOPE ENTRY POINT  
X=X+1  
*****  
; RECEIVER LINE TEST LINE 0  
RCV0: JSR 5,RCVTST ; GO TEST RECEIVER LINE 0  
LINE0  
Y=Y+1  
*****  
RT47: 47 ; ROUTINE # 47  
RT50 ; ADDRESS OF NEXT TEST  
100. ; ITERATION COUNT  
RCV1 ; SCOPE ENTRY POINT  
X=X+1  
*****  
; RECEIVER LINE TEST LINE 1  
RCV1: JSR 5,RCVTST ; GO TEST RECEIVER LINE 1  
LINE1  
Y=Y+1  
*****  
RT50: 50 ; ROUTINE # 50  
RT51 ; ADDRESS OF NEXT TEST  
100. ; ITERATION COUNT  
RCV2 ; SCOPE ENTRY POINT  
X=X+1  
*****  
; RECEIVER LINE TEST LINE 2  
RCV2: JSR 5,RCVTST ; GO TEST RECEIVER LINE 2  
LINE2  
Y=Y+1  
*****  
RT51: 51 ; ROUTINE # 51  
RT52 ; ADDRESS OF NEXT TEST  
100. ; ITERATION COUNT  
RCV3 ; SCOPE ENTRY POINT  
X=X+1  
*****  
; RECEIVER LINE TEST LINE 3  
RCV3: JSR 5,RCVTST ; GO TEST RECEIVER LINE 3  
LINE3  
Y=Y+1  
*****  
RT52: 52 ; ROUTINE # 52  
RT53 ; ADDRESS OF NEXT TEST  
100. ; ITERATION COUNT  
RCV4 ; SCOPE ENTRY POINT  
X=X+1
```

2391		000052			X=X+1	
2392					*****	
2393					RECEIVER LINE TEST LINE 4	
2394	012104	004537	005034		RCV4: JSR 5,RCVTST	GO TEST RECEIVER LINE 4
2395	012110	000010			LINE4	
2396		000005			Y=Y+1	
2397					*****	
2398	012112	000053			RT53: 53	ROUTINE # 53
2399	012114	012130			RT54	ADDRESS OF NEXT TEST
2400	012116	000144			100.	ITERATION COUNT
2401	012120	012122			RCV5	SCOPE ENTRY POINT
2402		000053			X=X+1	
2403					*****	
2404					RECEIVER LINE TEST LINE 5	
2405	012122	004537	005034		RCV5: JSR 5,RCVTST	GO TEST RECEIVER LINE 5
2406	012126	000012			LINE5	
2407		000006			Y=Y+1	
2408					*****	
2409	012130	000054			RT54: 54	ROUTINE # 54
2410	012132	012146			RT55	ADDRESS OF NEXT TEST
2411	012134	000144			100.	ITERATION COUNT
2412	012136	012140			RCV6	SCOPE ENTRY POINT
2413		000054			X=X+1	
2414					*****	
2415					RECEIVER LINE TEST LINE 6	
2416	012140	004537	005034		RCV6: JSR 5,RCVTST	GO TEST RECEIVER LINE 6
2417	012144	000014			LINE6	
2418		000007			Y=Y+1	
2419					*****	
2420	012146	000055			RT55: 55	ROUTINE # 55
2421	012150	012164			RT56	ADDRESS OF NEXT TEST
2422	012152	000144			100.	ITERATION COUNT
2423	012154	012156			RCV7	SCOPE ENTRY POINT
2424		000055			X=X+1	
2425					*****	
2426					RECEIVER LINE TEST LINE 7	
2427	012156	004537	005034		RCV7: JSR 5,RCVTST	GO TEST RECEIVER LINE 7
2428	012162	000016			LINE7	
2429		000010			Y=Y+1	
2430					*****	
2431	012164	000056			RT56: 56	ROUTINE # 56
2432	012166	012202			RT57	ADDRESS OF NEXT TEST
2433	012170	000144			100.	ITERATION COUNT
2434	012172	012174			RCV10	SCOPE ENTRY POINT
2435		000056			X=X+1	
2436					*****	
2437					RECEIVER LINE TEST LINE 10	
2438	012174	004537	005034		RCV10: JSR 5,RCVTST	GO TEST RECEIVER LINE 10
2439	012200	000020			LINE10	
2440		000011			Y=Y+1	
2441					*****	
2442	012202	000057			RT57: 57	ROUTINE # 57
2443	012204	012220			RT60	ADDRESS OF NEXT TEST
2444	012206	000144			100.	ITERATION COUNT
2445	012210	012212			RCV11	SCOPE ENTRY POINT
2446		000057			X=X+1	

```
2447
2448
2449 012212 004537 005034
2450 012216 000022
2451 000012
2452
2453 012220 000060
2454 012222 012236
2455 012224 000144
2456 012226 012230
2457 000060
2458
2459
2460 012230 004537 005034
2461 012234 000024
2462 000013
2463
2464 012236 000061
2465 012240 012254
2466 012242 000144
2467 012244 012246
2468 000061
2469
2470
2471 012246 004537 005034
2472 012252 000026
2473 000014
2474
2475 012254 000062
2476 012256 012272
2477 012260 000144
2478 012262 012264
2479 000062
2480
2481
2482 012264 004537 005034
2483 012270 000030
2484 000015
2485
2486 012272 000063
2487 012274 012310
2488 012276 000144
2489 012300 012302
2490 000063
2491
2492
2493 012302 004537 005034
2494 012306 000032
2495 000016
2496
2497 012310 000064
2498 012312 012326
2499 012314 000144
2500 012316 012320
2501 000064
2502
```

```
*****
;RECEIVER LINE TEST LINE 11
RCV11: JSR 5,RCVTST ;GO TEST RECEIVER LINE 11
        LINE11
        Y=Y+1
*****
RT60: 60 ;ROUTINE # 60
      100. ;ADDRESS OF NEXT TEST
      RCV12 ;ITERATION COUNT
      X=X+1 ;SCOPE ENTRY POINT
*****
;RECEIVER LINE TEST LINE 12
RCV12: JSR 5,RCVTST ;GO TEST RECEIVER LINE 12
        LINE12
        Y=Y+1
*****
RT61: 61 ;ROUTINE # 61
      100. ;ADDRESS OF NEXT TEST
      RCV13 ;ITERATION COUNT
      X=X+1 ;SCOPE ENTRY POINT
*****
;RECEIVER LINE TEST LINE 13
RCV13: JSR 5,RCVTST ;GO TEST RECEIVER LINE 13
        LINE13
        Y=Y+1
*****
RT62: 62 ;ROUTINE # 62
      100. ;ADDRESS OF NEXT TEST
      RCV14 ;ITERATION COUNT
      X=X+1 ;SCOPE ENTRY POINT
*****
;RECEIVER LINE TEST LINE 14
RCV14: JSR 5,RCVTST ;GO TEST RECEIVER LINE 14
        LINE14
        Y=Y+1
*****
RT63: 63 ;ROUTINE # 63
      100. ;ADDRESS OF NEXT TEST
      RCV15 ;ITERATION COUNT
      X=X+1 ;SCOPE ENTRY POINT
*****
;RECEIVER LINE TEST LINE 15
RCV15: JSR 5,RCVTST ;GO TEST RECEIVER LINE 15
        LINE15
        Y=Y+1
*****
RT64: 64 ;ROUTINE # 64
      100. ;ADDRESS OF NEXT TEST
      RCV16 ;ITERATION COUNT
      X=X+1 ;SCOPE ENTRY POINT
*****
```

Address	Op-Code	Op-Code Hex	Op-Code Dec	Comments
2503				RECEIVER LINE TEST LINE 16
2504	012320	004537	005034	RCV16: JSR 5,RCVTST ;GO TEST RECEIVER LINE 16
2505	012324	000034		LINE16
2506		000017		Y=Y+1
2507				*****
2508	012326	000065		RT65: 65 ;ROUTINE # 65
2509	012330	012346		RT66 ;ADDRESS OF NEXT TEST
2510	012332	000144		100. ;ITERATION COUNT
2511	012334	012336		RCV17 ;SCOPE ENTRY POINT
2512		000065		X=X+1
2513				*****
2514				RECEIVER LINE TEST LINE 17
2515	012336	004537	005034	RCV17: JSR 5,RCVTST ;GO TEST RECEIVER LINE 17
2516	012342	000036		LINE17
2517		000020		Y=Y+1
2518	012344	000240		NOP
2519				*****
2520	012346	000066		RT66: 66 ;ROUTINE # 66
2521	012350	012606		RT67 ;ADDR OF NEXT ROUTINE.
2522	012352	000012		10. ;ITERATION COUNT
2523	012354	012362		RT66A ;SCOPE ENTRY POINT.
2524		000066		X=X+1
2525				*****
2526				NOP
2527	012356	000240		NOP
2528	012360	000240		NOP
2529				TEST THAT NEX BIT (CSR BIT 14) SETS WHEN THE TRANSMITTER REFERENCES
2530				NON-EXISTANT MEMORY, THE CORRESPONDING BAR BIT CLEARS
2531				AND THAT AN INTERRUPT OCCURS. ALL LINES ARE USED FOR THE TEST
2532	012362	004737	004216	RT66A: JSR 7,TIMER ;GO CALCULATE MACHINE TIME TO TRANSMIT
2533				ONE CHARACTER
2534	012366	012701	001400	MOV 0LAT,X1 ;GET CAT ADDRESS
2535	012372	012702	160000	MOV 0160000,X2 ;GET A NON-EXISTANT ADDRESS
2536	012376	012703	000020	MOV 016,X3 ;GET COUNTER
2537	012402	010221		15 MOV X2,(1)+ ;LOAD THE CURRENT ADDRESS
2538	012404	005303		DEC X3 ;TABLE WITH NON-EXISTANT
2539	012406	001375		BNE 15 ;ADDRESSES
2540	012410	012701	000001	MOV 0LB10,X1 ;GET LINE BIT
2541	012414	012777	012552 167134	MOV 065,XMTINT ;LOAD TRANSMITTER INT. VECTOR
2542	012422	052777	000060 167112	25: BIS 060,0CSR ;SET EXTENDED ADDRESS BITS
2543	012430	050177	167110	BIS X1,0BAR ;START TRANSMITTER
2544	012434	013737	004370 012444	MOV TIME14,J5 ;LOAD DELAY TIME TO
2545	012442	104400		DELAY ;DELAY FOR 1/4TH OF A CHARACTER
2546	012444	000000		35: OPEN ;TO RESPOND TO NEX
2547	012446	017737	167072 001566	MOV 0BAR,RCV DAT ;GET BAR DATA & TEST
2548	012454	001406		BEQ 45 ;THAT IT IS CLEAR
2549	012456	005037	001570	CLR XMT DAT
2550	012462	104011		ERROR1 ;ERROR! BAR BIT DID NOT CLEAR
2551	012464	005077	167054	CLR 0BAR
2552	012470	000440		BR 75 ;GO TO SCOPE
2553	012472	032777	040000 167042	45: BIT 0BIT14,0CSR ;TEST THAT NEX BIT IS SET
2554	012500	001002		BNE 55 ;BRANCH IF SET
2555	012502	104001		ERROR ;ERROR! NEX BIT FAILED TO SET
2556	012504	000432		BR 75 ;GO TO SCOPE
2557	012506	042777	100000 167026	55: BIC 0BIT15,0CSR ;CLEAR TRANSMITTER READY FLAG
2558	012514	052777	010000 167020	BIS 0BIT12,0CSR ;SET TRANSMITTER IE BIT

```

2559 012522 005037 177776          CLR      20PSW          ,ALLOW INTERRUPTS
2560 012526 000240          NOP
2561 012530 012737 000340 177776    MOV      #PRTY7,20PSW    ,LOCK OUT INTERRUPTS
2562 012536 010137 001570          MOV      X1,XMTDAT    ,LOAD LINE THAT FAILED
2563 012542 005037 001566          CLR      RCVDAT
2564 012546 104011          ERROR1
2565                                ,ERROR! NEX FAILED TO CAUSE INTERRUPT
2566 012550 000410          BR      75
2567 012552 005077 166764 65:        CLR      20CSR
2568 012556 012737 000340 177776    MOV      #PRTY7,20PSW    ,LOCK OUT INTERRUPTS
2569 012564 022626          CMP      (6)+,(6)+    ,ADJUST STACK PTR
2570 012566 006301          ASL      X1          ,SHIFT LINE BIT
2571 012570 103314          BCC      25          ,DO NEXT LINE
2572 012572 013737 004366 012602 75:  MOV      TIME1,85    ,WAIT FOR TRANSMITTER TO RUN
2573 012600 104400          DELAY
2574 012602 000000 85:        OPEN
2575 012604 104006          SCOPE    ,EXITING TEST
2576                                ,SCOPE
2577 012606 000067          ;*****
2578 012610 012716          RT67:  67          ,ROUTINE # 67
2579 012612 000012          RT70          ,ADDR OF NEXT ROUTINE
2580 012614 012616          100          ,ITERATION COUNT
2581          RT67A          ,SCOPE ENTRY POINT.
2582          X=X+1
2583          ;*****
2584          ;TEST THAT NEX BIT SETS IF THE DM11 TABLES ARE IN NON-EXISTANT CORE
2585 012616 012777 160000 166724  RT67A:  MOV      #160000,20ASREG  ,SET BASE REGISTER TO NON-EXISTANT ADRS
2586 012624 012737 012706 000004    MOV      #45,20ERRVEC    ,SET TIME OUT TRAP VECTOR
2587 012632 005737 160000          TST      20160000    ,CHECK THAT ADDRESS TIMES OUT
2588 012636 013737 004370 012654    MOV      TIME14,15    ,GET TIME TO TRANSMIT 1/4 CHAR
2589 012644 062777 000001 166672    BIS      @BIT0,20AR    ,START TO TRANSMIT ON LINE 0
2590 012652 104400          DELAY    ,DELAY 1/4TH OF A CHARACTER
2591 012654 000000 15:        OPEN    ,TIME
2592 012656 005077 166662          CLR      20AR          ,STOP TRANSMITTER
2593 012662 022777 040060 166652    CMP      @BIT14+60,20CSR  ,TEST THAT ONLY NEX IS SET
2594 012670 001401          BEQ      25
2595 012672 104001          ERROR    ,ERROR! EITHER NEX FALED TO SET
2596                                ,OR OTHER BITS SET
2597 012674 013737 004366 012704 25:  MOV      TIME1,35    ,DELAY 1 CHARACTER TIME TO ALLOW
2598 012702 104400          DELAY    ,TRANSMITTER TO RUN TO
2599 012704 000000 35:        OPEN    ,COMPLETION
2600 012706 012737 000006 000004 45:  MOV      20ERRVEC+2,20ERRVEC  ,RESTORE TIME OUT TRAP
2601 012714 104006          SCOPE
2602          ;*****
2603 012716 000070          RT70:  70          ,ROUTINE # 70
2604 012720 013032          RT71          ,ADDR OF NEXT ROUTINE.
2605 012722 000144          100          ,ITERATION COUNT
2606 012724 012726          RT70A          ,SCOPE ENTRY POINT.
2607          X=X+1
2608          ;*****
2609          ;TEST THAT WHEN THE GO BIT IS CLEAR THAT THE RECEIVERS DO NOT RECEIVE
2610          ;DATA. EACH LINE IN TURN IS TRANSMITTED ON, AND WHEN TEN CHARACTERS
2611          ;HAVE BEEN TRANSMITTED THE RECEIVER DONE FLAG IS TESTED. IF IT IS SET
2612          ;AN ERROR IS INDICATED ON THE LINE DATA WAS RECEIVED ON
2613          RT70A:  CLR      LINE          ,SET UP TO TRANSMIT
2614 012726 005037 016730

```

2615	012732	004537	006246	15:	JSR	5, 20XMITD	; 10 CHARACTERS
2616	012736	177766			-10.		; ON EACH LINE
2617	012740	005777	166576		TST	2CSR	; WAIT FOR 10 CHARACTERS
2618	012744	100375			BPL	-4	; TO BE TRANSMITTED
2619	012746	042777	100000	166566	BIC	#100000, 2CSR	
2620	012754	105777	166562		TSTB	2CSR	; TEST RECEIVER DONE FLAG
2621	012760	100010			BPL	25	
2622	012762	013737	001564	001566	MOV	LINBIT, RCVDAT	; GET LINE BIT OF ACTIVE LINE
2623	012770	013737	001564	001570	MOV	LINBIT, XMTDAT	; THAT ERROR OCCURED ON
2624	012776	104011			ERROR1		; ERROR! DATA WAS RECEIVED ON LINE INDICATED
2625	013000	000413			BR	45	; GO TO SCOPE
2626	013002	062737	000002	016730	25:	ADD	#2, LINE
2627	013010	006337	001564		ASL	LINBIT	; SET UP NEXT LINE NUMBER
2628	013014	103346			BCC	15	; GET READY TO TRANSMIT ON NEXT LINE
2629	013016	013737	004366	013026	MOV	TIME1, 35	; GO TRANSMIT ON NEXT LINE
2630	013024	104400			DELAY		; DELAY 1 CHARACTER
2631	013026	000000			35:	0	; TIME BEFORE ENTERING NEXT TEST
2632	013030	104006			45:	SCOPE	; SCOPE
2633					; *****		
2634	013032	000071			RT71:	71	; ROUTINE # 71
2635	013034	013226				RT72	; ADDR OF NEXT ROUTINE
2636	013036	000024				20.	; ITERATION COUNT
2637	013040	013042				RT71A	; SCOPE ENTRY POINT.
2638		000071				X=X+1	
2639					; *****		
2640					; TEST THAT CURRENT ADDRESS INCREMENTS PROPERLY WHEN A CHAR-		
2641					; ACTER IS TRANSMITTED. LINE 0 IS USED FOR THE TEST.		
2642					RT71A:	CLR	X0
2643	013042	005000			15:	MOV	X0, X1
2644	013044	010001				INC	X1
2645	013046	005201				MOV	#35, 20ERRVEC
2646	013050	012737	013212	000004		MOV	#PRTY7, 20ERRVEC+2
2647	013056	012737	000340	000006		MOV	2CAT, 20ASREG
2648	013064	012777	001400	166456		MOV	X0, CAT
2649	013072	010037	001400			MOV	(0)
2650	013076	105710				TSTB	(0)
2651	013100	012737	177776	001440		MOV	#-2, WCT
2652	013106	012777	000005	166426		MOV	#5, 2CSR
2653	013114	012777	000001	166422		MOV	0LB10, 20AR
2654	013122	105777	166414			TSTB	2CSR
2655	013126	100375				BPL	-4
2656	013130	042777	000200	166404		BIC	#200, 2CSR
2657	013136	105777	166400			TSTB	2CSR
2658	013142	100375				BPL	-4
2659	013144	023701	001400			CMP	CAT, X1
2660							
2661	013150	001413				BEQ	25
2662	013152	010137	001570			MOV	X1, XMTDAT
2663	013156	013737	001400	001566		MOV	CAT, RCVDAT
2664	013164	104011				ERROR1	
2665	013166	032777	040000	166006		BIT	#BIT14, 2SWR
2666	013174	001323				BNE	15
2667	013176	000405				BR	35
2668	013200	005701			25:	TST	X1
2669	013202	001403				BEQ	35
2670	013204	000261				SEC	

2671	013206	006100				ROL	X0	
2672	013210	100715				BMI	15	
2673	013212	012737	000006	000004	35:	MOV	#ERRVEC+2, @ERRVEC	; RESTORE TIME OUT TRAP
2674	013220	005037	000006			CLR	@ERRVEC+2	
2675	013224	104006				SCOPE		; SCOPE
2676						; *****		
2677	013226	000072				RT72:	72	; ROUTINE # 72
2678	013230	013572					RT73	; ADDR OF NEXT ROUTINE.
2679	013232	000024					20.	; ITERATION COUNT
2680	013234	013236					RT72A	; SCOPE ENTRY POINT.
2681		000072					X=X+1	
2682						; *****		
2683						; TEST THAT DATA CAN BE TRANSMITTED FROM ALL AVAILABLE CORE.		
2684						; LINE 0 IS USED FOR THE TEST AND ONLY ONE WORD IS TRANSMITTED		
2685						; AT A TIME.		
2686						RT72A:	CLR	X0
2687	013236	005000					RESET	; CLEAR INDEX REGISTER
2688	013240	000005				15:	MOV	AREA(0), CAT
2689	013242	016037	013526	001400			MOV	#5, @CSR
2690	013250	012777	000005	166264			MOV	#-1, WCT
2691	013256	012737	177777	001440			MOV	@LBTO, @BAR
2692	013264	012777	000001	166252			MOV	@LBTO, @BAR
2693	013272	005777	166244				TST	@CSR
2694	013276	100375					BPL	-4
2695	013300	105777	166236				TSTB	@CSR
2696	013304	100375					BPL	-4
2697	013306	005077	166230				CLR	@CSR
2698	013312	005037	001566				CLR	RCVDT
2699	013316	113737	001600	001566			MOV	TUNTAB, RCVDT
2700	013324	117037	013526	001570			MOV	@AREA(0), XMTDT
2701	013332	043737	001572	001570			BIC	CARNSK, XMTDT
2702	013340	123737	001566	001570			CMPS	RCVDT, XMTDT
2703	013346	001402					BEQ	25
2704	013350	104011					ERROR1	
2705						; ((CAT)-1 IS THE MEMORY LOCATION WHERE THE DATA WAS TRANSMITTED FROM		
2706	013352	000464					BR	65
2707	013354	020027	000006			25:	CMPS	X0, #6
2708	013360	001402					BEQ	35
2709	013362	005720					TST	(0)+
2710	013364	000726					BR	15
2711						; GO REPEAT TEST		
2712								
2713	013366					35:	MOV	#55, @ERRVEC
2714	013366	012737	013514	000004				; BEGIN TESTING ABOVE 4K
2715								; SET TIME OUT TRAP TO EXIT
2716	013374	005001					CLR	X1
2717	013376	005201				45:	INC	X1
2718	013400	005720					TST	(0)+
2719	013402	110170	013526				MOV	X1, @AREA(0)
2720	013406	016037	013526	001400			MOV	AREA(0), CAT
2721	013414	012777	000005	166120			MOV	#5, @CSR
2722	013422	012737	177777	001440			MOV	#-1, WCT
2723	013430	012777	000001	166106			MOV	@LBTO, @BAR
2724	013436	005777	166100				TST	@CSR
2725	013442	100375					BPL	-4
2726	013444	105777	166072				TSTB	@CSR
								; TEST FOR CHARACTER DONE

2727	013450	100375			BPL	-4	
2728	013452	005077	166064		CLR	2CSR	
2729	013456	113737	001600	001566	MOVB	TUMTAB,RCVDAT	;GET THE RECEIVED CHARACTER
2730	013464	117037	013526	001570	MOVB	2AREA(0),XMTDAT	;GET THE TRANSMITTED CHARACTER
2731	013472	043737	001572	001570	BIC	CARMSK,XMTDAT	;CLEAR NON-TRANSMITTED BITS
2732	013500	123737	001566	001570	CMPB	RCVDAT,XMTDAT	;COMPARE CHARACTERS
2733	013506	001773			BEQ	45	;BRANCH IF VALID COMPARISON
2734	013510	104011			ERROR1		;ERROR! DATA COMPARISON ERROR NUMBER
2735	013512	000404			BR	65	;IN S/B GIVES MEMORY LOCATION (SEE TABLE
2736	013514	022626			POPSP2		;RESET THE STACK
2737	013516	012737	000006	000004	MOV	#6,2ERRVEC	;RESTORE TIME OUT TRAP
2738	013524	104006			SCOPE		;EXIT TEST
2739					MEMORY		
2740	013526	000000			LOCATIONS TRANSMITTED FROM TABLE		
2741	013530	005252			AREA:	0	;FOR DATA IN FIRST
2742	013532	012525				5252	;4K SEE THE LISTING
2743	013534	017777				12525	;CONTENTS OF THESE LOCATIONS (BYTE)
2744	013536	020000				17777	;IS THE DATA TRANSMITTED
2745	013540	026314			A8K:	20000	;CONTENTS =1 (IF AVAILABLE)
2746	013542	031463				26314	
2747						31463	
2748	013544	037477					
2749	013546	040000			A12K:	37477	
2750	013550	057477				40000	
2751	013552	060000			A16K:	57477	
2752	013554	077477				60000	
2753	013556	100000				77477	
2754	013560	117477			A20K:	100000	
2755	013562	120000				117477	
2756	013564	137477			A24K:	120000	
2757	013566	140000				137477	
2758	013570	173000			A28K:	140000	
2759						173000	
2760	013572	000073			RT73:	73	;ROUTINE # 73
2761	013574	014006				RT74	;ADDR OF NEXT ROUTINE.
2762	013576	000012				10.	;ITERATION COUNT
2763	013600	013602				RT73A	;SCOPE ENTRY POINT.
2764		000073				X=X+1	
2765							
2766							
2767							
2768							
2769	013602	005037	016730		RT73A:	CLR	LINE
2770	013606	012777	000001	165726	15:	MOV	#1,2CSR
2771	013614	005037	001566			CLR	RCVDAT
2772	013620	013737	016730	001570		MOV	2LINE,XMTDAT
2773	013626	004537	006246			JSR	5,2XMITD
2774	013632	177634				-100.	
2775	013634	106777	165702		25:	TSTB	2CSR
2776	013640	100375				BPL	25
2777	013642	042777	000200	165672		BIC	#BIT7,2CSR
2778	013650	005237	001566			INC	RCVDAT
2779	013654	023727	001566	000144		CMF	RCVDAT,#100
2780	013662	001416				BEQ	45
2781	013664	005777	165652			TST	2CSR
2782	013670	100002				BPL	35

```

2783 013672 104011
2784
2785 013674 000443
2786
2787 013676 023777 001564 165640 35: CMP @L INBIT, @BAR ;TEST THAT BAR BIT IS SET
2788 013704 001753 BEQ 25 ;BRANCH IF SET
2789 013706 017737 165632 001570 MOV @BAR, XMTDAT ;GET BAR CONTENTS
2790 013714 104011
2791
2792
2793 013716 000432
2794 013720 013737 004370 013730 45: BR 85 ;ERROR! READY BIT SET TOO SOON
;TYPEOUT SHOWS HOW MANY CHARS WERE RECEIVED WHEN READY SET AND THE LINE # (X2)
;GO TO EXIT
2795 013726 104400
2796 013730 000000
2797 013732 005777 165604 55: OPEN ;EXIT TEST
;DELAY 1/4 CHARACTER TIME
;TO ALLOW TRANSMITTER TO FINISH
2798 013736 100402
2799 013740 104001
2800 013742 000420
2801 013744 005777 165574 65: TST @CSR ;TEST READY FLAG (SHOULD BE SET)
;GO TEST BAR
;ERROR! READY FLAG FAILED TO SET
;GO TO EXIT
2802 013750 001407 BEQ 75 ;TEST THAT BAR BIT IS CLEAR
;GO TO 75 IF CLEAR
2803 013752 017737 165566 001566 MOV @BAR, RCVDAT
2804 013760 005037 001570 CLR XMTDAT
2805 013764 104011
2806 013766 000406
2807 013770 062737 000002 016730 75: BR 85 ;ERROR! BAR BIT FAILED TO CLEAR
2808 013776 006337 001564
2809 014002 103301
2810 014004 104006
2811
2812 014006 000074
2813 014010 014174
2814 014012 000012
2815 014014 014016
2816 000074
2817
2818
2819
2820
2821
2822 014016 012701 001600
2823 014022 012702 000100
2824 014026 005021
2825 014030 005302
2826 014032 001375
2827 014034 012701 001600
2828 014040 012777 000004 165474
2829 014046 005037 001570
2830 014052 005037 001566
2831 014056 012737 177677 001440
2832 014064 062777 000001 165450
2833 014072 012777 000001 165444
2834 014100 105777 165436
2835 014104 100375
2836 014106 042777 000200 165426
2837 014114 005237 001566
2838 014120 005237 001570

;*****
RT74: 74 ;ROUTINE # 74
RT75 ;ADDR OF NEXT ROUTINE
10 ;ITERATION COUNT
RT74A ;SCOPE ENTRY POINT
X=X+1
;*****

;TEST THAT THE TUMBLE TABLE POINTER INCREMENTS PROPERLY AND
;RETURNS TO THE BEGINNING AFTER 64 CHARACTERS HAVE BEEN RECEIVED
;LINE 0 IS USED FOR THE TEST
RT74A: MOV @TUMTAB, X1 ;CLEAR THE
MOV @64, X2 ;TUMBLE TABLE
15: CLR (1)+
DEC X2
BNC 15
MOV @TUMTAB, X1
MOV @BIT2, @CSR ;SET MAINT BIT & CLEAR GO BIT
CLR XMTDAT
CLR RCVDAT
MOV @-65, WCT ;SET UP TO TRANSMIT 65 CHARACTERS
BIS @BIT0, @CSR ;SET THE GO BIT
MOV @LBIT0, @BAR ;TRANSMIT ON LINE 0
25: TSTB @CSR ;WAIT FOR CHAR DONE FLAG
BPL 25
BIC @BIT7, @CSR ;CLEAR CHAR DONE FLAG
INC RCVDAT ;INCREMENT CHARACTERS
INC XMTDAT ;RECEIVED COUNT

```

2839	014124	005711			TST (1)		; TEST TT ENTRY FOR VALID
2840	014126	100402			BMI 35		; DATA ENTRY
2841	014130	104011			ERROR1		; ERROR! NO VALID DATA ENTRY
2842							; TYPEOUT SHOWS # OF CHARS RCVD WHEN ERROR OCCURED
2843	014132	000417			BR 45		; GO TO SCOPE
2844	014134	005021			CLR (1)+		; CLEAR TT ENTRY
2845	014136	023727	001570	000100	CMP XMTDAT, #64.		; HAVE 64. CHARACTERS BEEN RECEIVED
2846	014144	001355			BNE 25		
2847	014146	005777	165370		TST @CSR		; WAIT FOR THE LAST CHARACTER
2848	014152	100375			BPL -4		; TO BE TRANSMITTED
2849	014154	105777	165362		TSTB @CSR		; TEST FOR DONE
2850	014160	100375			BPL -4		
2851	014162	005737	001600		TST TUMTAB		; TEST FIRST TT ENTRY
2852	014166	100401			BMI 45		; FOR VALID DATA
2853	014170	104001			ERROR		; ERROR! POINTER DID NOT RETURN
2854	014172	104006			SCOPE		; SCOPE
2855		000000			A=0		
2856		000000			Y=0		
2857					; *****		
2858	014174	000075			RT75: 75		; ROUTINE # 75
2859	014176	014212			RT76		; ADDRESS OF NEXT TEST
2860	014200	000144			100.		; ITERATION COUNT
2861	014202	014204			BRK0		; SCOPE ENTRY POINT
2862		000075			X=X+1		
2863					; *****		
2864					; BREAK TEST ON LINE 0.		
2865	014204	004537	005456		BRK0: JSR 5, BRKTST		; GO DO BREAK TEST
2866	014210	000001			LBIT0		; ON LINE 0
2867		000001			Y=Y+1		
2868					; *****		
2869	014212	000076			RT76: 76		; ROUTINE # 76
2870	014214	014230			RT77		; ADDRESS OF NEXT TEST
2871	014216	000144			100.		; ITERATION COUNT
2872	014220	014222			BRK1		; SCOPE ENTRY POINT
2873		000076			X=X+1		
2874					; *****		
2875					; BREAK TEST ON LINE 1.		
2876	014222	004537	005456		BRK1: JSR 5, BRKTST		; GO DO BREAK TEST
2877	014226	000002			LBIT1		; ON LINE 1
2878		000002			Y=Y+1		
2879					; *****		
2880	014230	000077			RT77: 77		; ROUTINE # 77
2881	014232	014246			RT100		; ADDRESS OF NEXT TEST
2882	014234	000144			100.		; ITERATION COUNT
2883	014236	014240			BRK2		; SCOPE ENTRY POINT
2884		000077			X=X+1		
2885					; *****		
2886					; BREAK TEST ON LINE 2		
2887	014240	004537	005456		BRK2: JSR 5, BRKTST		; GO DO BREAK TEST
2888	014244	000004			LBIT2		; ON LINE 2
2889		000003			Y=Y+1		
2890					; *****		
2891	014246	000100			RT100: 100		; ROUTINE # 100
2892	014250	014264			RT101		; ADDRESS OF NEXT TEST
2893	014252	000144			100.		; ITERATION COUNT
2894	014254	014256			BRK3		; SCOPE ENTRY POINT

2895		000100							
2896									
2897									
2898	014256	004537	005456						
2899	014262	000010							
2900		000004							
2901									
2902	014264	000101							
2903	014266	014302							
2904	014270	000144							
2905	014272	014274							
2906		000101							
2907									
2908									
2909	014274	004537	005456						
2910	014300	000020							
2911		000005							
2912									
2913	014302	000102							
2914	014304	014320							
2915	014306	000144							
2916	014310	014312							
2917		000102							
2918									
2919									
2920	014312	004537	005456						
2921	014316	000040							
2922		000006							
2923									
2924	014320	000103							
2925	014322	014336							
2926	014324	000144							
2927	014326	014330							
2928		000103							
2929									
2930									
2931	014330	004537	005456						
2932	014334	000100							
2933		000007							
2934									
2935	014336	000104							
2936	014340	014354							
2937	014342	000144							
2938	014344	014346							
2939		000104							
2940									
2941									
2942	014346	004537	005456						
2943	014352	000200							
2944		000010							
2945									
2946	014354	000105							
2947	014356	014372							
2948	014360	000144							
2949	014362	014364							
2950		000105							


```

      X=X+1
;*****
;BREAK TEST ON LINE 3.
BRK3: JSR      5,BRKTST      ;GO DO BREAK TEST
      LBIT3      ;ON LINE 3
      Y=Y+1
;*****
RT101: 101      ;ROUTINE # 101
      RT102      ;ADDRESS OF NEXT TEST
      100.      ;ITERATION COUNT
      BRK4      ;SCOPE ENTRY POINT
      X=X+1
;*****
;BREAK TEST ON LINE 4.
BRK4: JSR      5,BRKTST      ;GO DO BREAK TEST
      LBIT4      ;ON LINE 4
      Y=Y+1
;*****
RT102: 102      ;ROUTINE # 102
      RT103      ;ADDRESS OF NEXT TEST
      100.      ;ITERATION COUNT
      BRK5      ;SCOPE ENTRY POINT
      X=X+1
;*****
;BREAK TEST ON LINE 5.
BRK5: JSR      5,BRKTST      ;GO DO BREAK TEST
      LBIT5      ;ON LINE 5
      Y=Y+1
;*****
RT103: 103      ;ROUTINE # 103
      RT104      ;ADDRESS OF NEXT TEST
      100.      ;ITERATION COUNT
      BRK6      ;SCOPE ENTRY POINT
      X=X+1
;*****
;BREAK TEST ON LINE 6.
BRK6: JSR      5,BRKTST      ;GO DO BREAK TEST
      LBIT6      ;ON LINE 6
      Y=Y+1
;*****
RT104: 104      ;ROUTINE # 104
      RT105      ;ADDRESS OF NEXT TEST
      100.      ;ITERATION COUNT
      BRK7      ;SCOPE ENTRY POINT
      X=X+1
;*****
;BREAK TEST ON LINE 7.
BRK7: JSR      5,BRKTST      ;GO DO BREAK TEST
      LBIT7      ;ON LINE 7
      Y=Y+1
;*****
RT105: 105      ;ROUTINE # 105
      RT106      ;ADDRESS OF NEXT TEST
      100.      ;ITERATION COUNT
      BRK10     ;SCOPE ENTRY POINT
      X=X+1

```

2951				*****
2952				;BREAK TEST ON LINE 10.
2953	014364	004537	005456	BRK10: JSR 5, BRKTST ;GO DO BREAK TEST
2954	014370	000400		LBIT10 ;ON LINE 10
2955		000011		Y=Y+1
2956				*****
2957	014372	000106		RT106: 106 ;ROUTINE # 106
2958	014374	014410		RT107 ;ADDRESS OF NEXT TEST
2959	014376	000144		100. ;ITERATION COUNT
2960	014400	014402		BRK11 ;SCOPE ENTRY POINT
2961		000106		X=X+1
2962				*****
2963				;BREAK TEST ON LINE 11.
2964	014402	004537	005456	BRK11: JSR 5, BRKTST ;GO DO BREAK TEST
2965	014406	001000		LBIT11 ;ON LINE 11
2966		000012		Y=Y+1
2967				*****
2968	014410	000107		RT107: 107 ;ROUTINE # 107
2969	014412	014426		RT110 ;ADDRESS OF NEXT TEST
2970	014414	000144		100. ;ITERATION COUNT
2971	014416	014420		BRK12 ;SCOPE ENTRY POINT
2972		000107		X=X+1
2973				*****
2974				;BREAK TEST ON LINE 12.
2975	014420	004537	005456	BRK12: JSR 5, BRKTST ;GO DO BREAK TEST
2976	014424	002000		LBIT12 ;ON LINE 12
2977		000013		Y=Y+1
2978				*****
2979	014426	000110		RT110: 110 ;ROUTINE # 110
2980	014430	014444		RT111 ;ADDRESS OF NEXT TEST
2981	014432	000144		100. ;ITERATION COUNT
2982	014434	014436		BRK13 ;SCOPE ENTRY POINT
2983		000110		X=X+1
2984				*****
2985				;BREAK TEST ON LINE 13.
2986	014436	004537	005456	BRK13: JSR 5, BRKTST ;GO DO BREAK TEST
2987	014442	004000		LBIT13 ;ON LINE 13
2988		000014		Y=Y+1
2989				*****
2990	014444	000111		RT111: 111 ;ROUTINE # 111
2991	014446	014462		RT112 ;ADDRESS OF NEXT TEST
2992	014450	000144		100. ;ITERATION COUNT
2993	014452	014454		BRK14 ;SCOPE ENTRY POINT
2994		000111		X=X+1
2995				*****
2996				;BREAK TEST ON LINE 14.
2997	014454	004537	005456	BRK14: JSR 5, BRKTST ;GO DO BREAK TEST
2998	014460	010000		LBIT14 ;ON LINE 14
2999		000015		Y=Y+1
3000				*****
3001	014462	000112		RT112: 112 ;ROUTINE # 112
3002	014464	014500		RT113 ;ADDRESS OF NEXT TEST
3003	014466	000144		100. ;ITERATION COUNT
3004	014470	014472		BRK15 ;SCOPE ENTRY POINT
3005		000112		X=X+1
3006				*****

Line	Address	Hex	Label	Code	Comment
3007					,BREAK TEST ON LINE 15
3008	014472	004537	005456	BRK15:	JSR 5,BRKTST ;GO DO BREAK TEST
3009	014476	020000			LBIT15 ;ON LINE 15
3010		000016			Y=Y+1
3011					*****
3012	014500	000113		RT113:	113 ;ROUTINE # 113
3013	014502	014516			RT114 ;ADDRESS OF NEXT TEST
3014	014504	000144			100. ;ITERATION COUNT
3015	014506	014510			BRK16 ;SCOPE ENTRY POINT
3016		000113			X=X+1
3017					*****
3018					,BREAK TEST ON LINE 16.
3019	014510	004537	005456	BRK16:	JSR 5,BRKTST ;GO DO BREAK TEST
3020	014514	040000			LBIT16 ;ON LINE 16
3021		000017			Y=Y+1
3022					*****
3023	014516	000114		RT114:	114 ;ROUTINE # 114
3024	014520	014534			RT115 ;ADDRESS OF NEXT TEST
3025	014522	000144			100. ;ITERATION COUNT
3026	014524	014526			BRK17 ;SCOPE ENTRY POINT
3027		000114			X=X+1
3028					*****
3029					,BREAK TEST ON LINE 17.
3030	014526	004537	005456	BRK17:	JSR 5,BRKTST ;GO DO BREAK TEST
3031	014532	100000			LBIT17 ;ON LINE 17
3032		000020			Y=Y+1
3033		000000			A=0
3034		000000			Y=0
3035					*****
3036	014534	000115		RT115:	115 ;ROUTINE #115
3037	014536	014552			RT116 ;ADDRESS OF NEXT TEST
3038	014540	000144			100. ;ITERATION COUNT
3039	014542	014544			DATO ;SCOPE ENTRY POINT
3040		000115			X=X+1
3041					*****
3042					,DATA TEST 100 CHARACTERS LINE0
3043	014544	004537	006312	DATO:	JSR 5,DATTST ;GO RUN DATA TEST
3044	014550	000000			LINE0 ;ON LINE0
3045		000001			Y=Y+1
3046					*****
3047	014552	000116		RT116:	116 ;ROUTINE #116
3048	014554	014570			RT117 ;ADDRESS OF NEXT TEST
3049	014556	000144			100. ;ITERATION COUNT
3050	014560	014562			DAT1 ;SCOPE ENTRY POINT
3051		000116			X=X+1
3052					*****
3053					,DATA TEST 100 CHARACTERS LINE1
3054	014562	004537	006312	DAT1:	JSR 5,DATTST ;GO RUN DATA TEST
3055	014566	000002			LINE1 ;ON LINE1
3056		000002			Y=Y+1
3057					*****
3058	014570	000117		RT117:	117 ;ROUTINE #117
3059	014572	014606			RT120 ;ADDRESS OF NEXT TEST
3060	014574	000144			100. ;ITERATION COUNT
3061	014576	014600			DAT2 ;SCOPE ENTRY POINT
3062		000117			X=X+1

3063				*****
3064				; DATA TEST 100 CHARACTERS LINE2
3065	014600	004537	006312	DAT2: JSR 5, DATTST ; GO RUN DATA TEST
3066	014604	000004		LINE2 ; ON LINE2
3067		000003		Y=Y+1
3068				*****
3069	014606	000120		RT120: 120 ; ROUTINE #120
3070	014610	014624		RT121 ; ADDRESS OF NEXT TEST
3071	014612	000144		100. ; ITERATION COUNT
3072	014614	014616		DAT3 ; SCOPE ENTRY POINT
3073		000120		X=X+1
3074				*****
3075				; DATA TEST 100 CHARACTERS LINE3
3076	014616	004537	006312	DAT3 JSR 5, DATTST ; GO RUN DATA TEST
3077	014622	000006		LINE3 ; ON LINE3
3078		000004		Y=Y+1
3079				*****
3080	014624	000121		RT121: 121 ; ROUTINE #121
3081	014626	014642		RT122 ; ADDRESS OF NEXT TEST
3082	014630	000144		100. ; ITERATION COUNT
3083	014632	014634		DAT4 ; SCOPE ENTRY POINT
3084		000121		X=X+1
3085				*****
3086				; DATA TEST 100 CHARACTERS LINE4
3087	014634	004537	006312	DAT4: JSR 5, DATTST ; GO RUN DATA TEST
3088	014640	000010		LINE4 ; ON LINE4
3089		000005		Y=Y+1
3090				*****
3091	014642	000122		RT122: 122 ; ROUTINE #122
3092	014644	014660		RT123 ; ADDRESS OF NEXT TEST
3093	014646	000144		100. ; ITERATION COUNT
3094	014650	014652		DAT5 ; SCOPE ENTRY POINT
3095		000122		X=X+1
3096				*****
3097				; DATA TEST 100 CHARACTERS LINE5
3098	014652	004537	006312	DAT5: JSR 5, DATTST ; GO RUN DATA TEST
3099	014656	000012		LINE5 ; ON LINE5
3100		000006		Y=Y+1
3101				*****
3102	014660	000123		RT123: 123 ; ROUTINE #123
3103	014662	014676		RT124 ; ADDRESS OF NEXT TEST
3104	014664	000144		100. ; ITERATION COUNT
3105	014666	014670		DAT6 ; SCOPE ENTRY POINT
3106		000123		X=X+1
3107				*****
3108				; DATA TEST 100 CHARACTERS LINE6
3109	014670	004537	006312	DAT6: JSR 5, DATTST ; GO RUN DATA TEST
3110	014674	000014		LINE6 ; ON LINE6
3111		000007		Y=Y+1
3112				*****
3113	014676	000124		RT124: 124 ; ROUTINE #124
3114	014700	014714		RT125 ; ADDRESS OF NEXT TEST
3115	014702	000144		100. ; ITERATION COUNT
3116	014704	014706		DAT7 ; SCOPE ENTRY POINT
3117		000124		X=X+1
3118				*****

3119					
3120	014706	004537	006312		
3121	014712	000016			
3122		000010			
3123					
3124	014714	000125			
3125	014716	014732			
3126	014720	000144			
3127	014722	014724			
3128		000125			
3129					
3130					
3131	014724	004537	006312		
3132	014730	000020			
3133		000011			
3134					
3135	014732	000126			
3136	014734	014750			
3137	014736	000144			
3138	014740	014742			
3139		000126			
3140					
3141					
3142	014742	004537	006312		
3143	014746	000022			
3144		000012			
3145					
3146	014750	000127			
3147	014752	014766			
3148	014754	000144			
3149	014756	014760			
3150		000127			
3151					
3152					
3153	014760	004537	006312		
3154	014764	000024			
3155		000013			
3156					
3157	014766	000130			
3158	014770	015004			
3159	014772	000144			
3160	014774	014776			
3161		000130			
3162					
3163					
3164	014776	004537	006312		
3165	015002	000026			
3166		000014			
3167					
3168	015004	000131			
3169	015006	015022			
3170	015010	000144			
3171	015012	015014			
3172		000131			
3173					
3174					


```

; DATA TEST 100 CHARACTERS LINE7
DAT7: JSR 5, DATTST ; GO RUN DATA TEST
      LINE7 ; ON LINE7
      Y=Y+1
; *****
RT125: 125 ; ROUTINE #125
      RT126 ; ADDRESS OF NEXT TEST
      100 ; ITERATION COUNT
      DAT10 ; SCOPE ENTRY POINT
      X=X+1
; *****
; DATA TEST 100 CHARACTERS LINE10
DAT10: JSR 5, DATTST ; GO RUN DATA TEST
      LINE10 ; ON LINE10
      Y=Y+1
; *****
RT126: 126 ; ROUTINE #126
      RT127 ; ADDRESS OF NEXT TEST
      100 ; ITERATION COUNT
      DAT11 ; SCOPE ENTRY POINT
      X=X+1
; *****
; DATA TEST 100 CHARACTERS LINE11
DAT11: JSR 5, DATTST ; GO RUN DATA TEST
      LINE11 ; ON LINE11
      Y=Y+1
; *****
RT127: 127 ; ROUTINE #127
      RT130 ; ADDRESS OF NEXT TEST
      100 ; ITERATION COUNT
      DAT12 ; SCOPE ENTRY POINT
      X=X+1
; *****
; DATA TEST 100 CHARACTERS LINE12
DAT12: JSR 5, DATTST ; GO RUN DATA TEST
      LINE12 ; ON LINE12
      Y=Y+1
; *****
RT130: 130 ; ROUTINE #130
      RT131 ; ADDRESS OF NEXT TEST
      100 ; ITERATION COUNT
      DAT13 ; SCOPE ENTRY POINT
      X=X+1
; *****
; DATA TEST 100 CHARACTERS LINE13
DAT13: JSR 5, DATTST ; GO RUN DATA TEST
      LINE13 ; ON LINE13
      Y=Y+1
; *****
RT131: 131 ; ROUTINE #131
      RT132 ; ADDRESS OF NEXT TEST
      100 ; ITERATION COUNT
      DAT14 ; SCOPE ENTRY POINT
      X=X+1
; *****
; DATA TEST 100 CHARACTERS LINE14

```

3175	015014	004537	006312	DATA: JSR 5, DATTST ; GO RUN DATA TEST
3176	015020	000030		LINE14 ; ON LINE14
3177		000015		Y=Y+1
3178				*****
3179	015022	000132		RT132: 132 ; ROUTINE #132
3180	015024	015040		RT133 ; ADDRESS OF NEXT TEST
3181	015026	000144		100. ; ITERATION COUNT
3182	015030	015032		DATA15 ; SCOPE ENTRY POINT
3183		000132		X=X+1
3184				*****
3185				DATA TEST 100 CHARACTERS LINE15
3186	015032	004537	006312	DATA15: JSR 5, DATTST ; GO RUN DATA TEST
3187	015036	000032		LINE15 ; ON LINE15
3188		000016		Y=Y+1
3189				*****
3190	015040	000133		RT133: 133 ; ROUTINE #133
3191	015042	015056		RT134 ; ADDRESS OF NEXT TEST
3192	015044	000144		100. ; ITERATION COUNT
3193	015046	015050		DATA16 ; SCOPE ENTRY POINT
3194		000133		X=X+1
3195				*****
3196				DATA TEST 100 CHARACTERS LINE16
3197	015050	004537	006312	DATA16: JSR 5, DATTST ; GO RUN DATA TEST
3198	015054	000034		LINE16 ; ON LINE16
3199		000017		Y=Y+1
3200				*****
3201	015056	000134		RT134: 134 ; ROUTINE #134
3202	015060	015074		RT135 ; ADDRESS OF NEXT TEST
3203	015062	000144		100. ; ITERATION COUNT
3204	015064	015066		DATA17 ; SCOPE ENTRY POINT
3205		000134		X=X+1
3206				*****
3207				DATA TEST 100 CHARACTERS LINE17
3208	015066	004537	006312	DATA17: JSR 5, DATTST ; GO RUN DATA TEST
3209	015072	000036		LINE17 ; ON LINE17
3210		000020		Y=Y+1
3211				*****
3212	015074	000135		RT135: 135 ; ROUTINE #135
3213	015076	015474		RT136 ; ADDR OF NEXT ROUTINE
3214	015100	000144		100. ; ITERATION COUNT
3215	015102	015104		RT135A ; SCOPE ENTRY POINT
3216		000135		X=X+1
3217				*****
3218				TEST THAT DATA (ALL 1'S) CAN BE TRANSMITTED ON LINES SIMULTANEOUSLY
3219				THE FOLLOWING TESTS ARE PERFORMED:
3220				THERE ARE 16 DATA ENTRIES
3221				THERE ISN'T A 17TH ENTRY
3222				DATA RECEIVED IS CORRECT
3223				ONE DATA ENTRY PER LINE
3224				
3225				
3226	015104	005037	001600	RT135A: CLR TUMTAB ; CLEAR THE
3227	015110	004537	006224	JSR 5, BMOVE ; TUMBLE
3228	015114	001600		TUMTAB ; TABLE
3229	015116	001601		TUMTAB+1 ; (200
3230	015120	000177		177 ; ENTRIES)

3231	015122	012737	177777	017102	MOV	#-1,OUTBUF	;LOAD CHAR INTO OUTPUT BUFFER	
3232	015130	005000			CLR	X0	;SET RO = LINE 0	
3233	015132	012737	000001	001564	MOV	#LBITO,LINBIT	;GET LINE BIT	
3234	015140	012777	000001	164374	MOV	#BIT0,ACSR	;SET THE GO BIT	
3235	015146	010037	016730		MOV	X0,LINE	;GET LINE NUMBER	
3236	015152	004537	006246		JSR	5,XMITD	;TRANSMIT 1 CHAR.	
3237	015156	177777			-1		;ON EACH LINE	
3238	015160	005720			TST	(0)+	;INCREMENT LINE NUMBER (+2)	
3239	015162	006337	001564		ASL	LINBIT	;SHIFT LINE BIT TO NEXT LINE	
3240	015166	103367			BCC	15	;BRANCH IF ALL LINES NOT DONE	
3241	015170	013737	004366	015200	MOV	TIME1,25	;PUT TIME TO TRANSMIT 1 CHAR	
3242	015176	104400			DELAY		;DELAY 1	
3243	015200	000000			OPEN		;CHARACTER TIME	
3244	015202	017737	164336	001566	MOV	BAR,RCVDAT	;GET & TEST BAR CONTENTS	
3245	015210	001410			BEQ	35	;BRANCH IF 0	
3246	015212	005037	001570		CLR	XMTDAT		
3247	015216	104011			ERROR1		;ERROR! BAR NOT CLEAR AFTER ALL	
3248	015220	005077	164320		CLR	BAR	;LINES FINISHED	
3249	015224	005077	164312		CLR	ACSR		
3250	015230	000520			BR	165	;GO TO EXIT	
3251	015232	032777	020000	164302	BIT	#BIT13,ACSR	;TEST THAT OVER RUN DID NOT SET	
3252	015240	001404			BEQ	45		
3253	015242	104001			ERROR		;ERROR! OVER RUN BIT SET	
3254	015244	005077	164272		CLR	ACSR		
3255	015250	000510			BR	165	;GO TO EXIT	
3256								
3257							;TEST THAT THERE ARE 16 VALID DATA ENTRIES	
3258	015252	005077	164264		45:	CLR	ACSR	;CLEAR THE CSR
3259	015256	012702	000020		MOV	#16,X2	;GET TT SCAN COUNT	
3260	015262	012701	001600		MOV	#TUMTAB,X1	;GET FIRST TT ADDRESS	
3261	015266	005302			55:	DEC	X2	;DECREMENT SCAN COUNTER
3262	015270	100404			BMI	65	;BRANCH IF 16 ENTRIES SCANNED	
3263	015272	005721			TST	(1)+	;TEST FOR VALID DATA ENTRY	
3264	015274	100774			BMI	55	;BRANCH IF FOUND	
3265	015276	104001			ERROR		;ERROR! MISSING DATA ENTRY	
3266	015300	000474			BR	165	;GO TO EXIT	
3267	015302	005721			65:	TST	(1)+	;TEST 17TH ENTRY (SHOULD BE = TO 0)
3268	015304	001402			BEQ	75	;BRANCH IF 0	
3269	015306	104301			ERROR		;ERROR! EXTRA DATA ENTRY	
3270	015310	000470			BR	165	;GO TO EXIT	
3271								
3272							;TEST THAT THE DATA IS CORRECT IN ALL 16 ENTRIES	
3273	015312	012701	001600		75:	MOV	#TUMTAB,X1	;GET FIST TT ADDRESS
3274	015316	012702	000020		MOV	#16,X2	;GET SCAN COUNT	
3275	015322	005302			85:	DEC	X2	;DECREMENT SCAN COUNT
3276	015324	100421			BMI	105	;BRANCH IF 16 ENTRIES SCANNED	
3277	015326	013737	017102	001570	MOV	OUTBUF,XMTDAT	;GET TRANSMITTED DATA	
3278	015334	043737	001572	001570	BIC	CARMSK,XMTDAT	;CLEAR NON-TRANSMITTED BITS	
3279	015342	113737	001600	001566	MOVB	TUMTAB,RCVDAT	;GET RECEIVED DATA	
3280	015360	123737	001570	001566	CPB	XMTDAT,RCVDAT	;COMPARE DATA	
3281	015364	001402			BEQ	95		
3282	015360	104011			ERROR1		;ERROR INCORRECT DATA	
3283	015362	000443			BR	165	;GO TO EXIT	
3284	015364	005721			95:	TST	(1)+	;INCREMENT TT ADDRESS
3285	015366	000755			BR	85	;TEST NEXT ENTRY	
3286								

```

3287
3288 015370 012701 001600      ; CLEAR ALL BUT LINE NUMBER IN TUMBLE TABLE ENTRY
3289 015374 012702 000020      105  MOV  #TUMTAB,%1      ; GET FIRST TT ADDRESS
3290 015400 005302              115  MOV  #16,%2      ; GET SCAN COUNT
3291 015402 100403              DEC  %2      ; DECREMENT SCAN COUNT
3292 015404 042721 160777      BMI  125      ; BRANCH IF ALL LINES TESTED
3293 015410 000773              BIC  #160777,(1)+ ; CLEAR ALL BUT LINE NUMBER IN TT
3294
3295                          BR  115      ; DO NEXT TT ADDRESS
3296
3296 015412 005037 001570      ; TEST THAT THERE IS AN ENTRY FOR EACH OF THE 16 LINES
3297 015416 012701 000020      125  CLR  XMTDAT
3298 015422 012702 000020      MOV  #16,%1
3299 015426 012700 001600      135  MOV  #16,%2
3300 015432 023720 001570      MOV  #TUMTAB,%0
3301 015436 001406              145  CMP  XMTDAT,(0)+ ; TEST FOR LINE ENTRY
3302 015440 005302              BEQ  155      ; BRANCH IF FOUND
3303 015442 001373              DEC  %2      ; DECREMENT SCAN COUNT
3304 015444 005037 001566      BNE  145      ; LOOK AT NEXT ENTRY
3305 015450 104011              CLR  RCVDAT
3306 015452 000407              ERROR1
3307 015454 005301              BR  165      ; ERROR! NO ENTRY FOUND FOR THIS LINE
3308 015456 005701              155  DEC  %1      ; GO TO EXIT
3309 015460 001404              TST  %1      ; DECREMENT LINES FOUND COUNT
3310 015462 062737 001000 001570 BEQ  165      ; BRANCH IF ALL LINE TESTED
3311 015470 000754              ADD  #1000,XMTDAT ; INCREMENT LINE NUMBER
3312 015472 104006              BR  135      ; GO DO NEXT LINE
3313
3313      SCOPE ; SCOPE
3314 015474 000136      RT136: 136 ; ROUTINE # 136
3315 015476 016010      RT137 ; ADDR OF NEXT ROUTINE
3316 015500 000144      100 ; ITERATION COUNT
3317 015502 015504      RT136A ; SCOPE ENTRY POINT.
3318 000136      X=X+1
3319
3319      ; *****
3320
3321
3321      ; TEST THAT THE DM11 CAN TRANSMIT A BREAK ON ALL LINES SIMULTANEOUSLY
3322 015504 013737 004366 015560 RT136A: MOV  #RTIME1,15 ; GET TIME TO TRANSMIT ONE CHARACTER
3323 015512 005037 001600      CLR  TUMTAB ; CLEAR
3324 015516 004537 006224      JSR  5,BMOVE ; THE
3325 015522 001600      TUMTAB ; TUMBLE
3326 015524 001601      TUMTAB+1 ; TABLE
3327 015526 000177      177
3328 015530 012777 000001 164004 MOV  #B1TO,%CSR ; SET GO
3329 015536 012777 177777 164002 MOV  #B-1,%KCSR ; SET BREAK BIT FOR ALL LINES
3330 015544 106777 163772      TSTB %CSR ; WAIT FOR THE RECEIVER
3331 015550 100375      BPL  -4 ; TO RECEIVE A BREAK
3332 015552 005077 163770      CLR  %KCSR ; CLEAR ALL BREAK BITS
3333 015556 104400      DELAY ; WAIT ONE CHARACTER
3334 015560 000000      OPEN ; TIME
3335 015562 022777 000201 163752 15: CMP  #201,%CSR ; TEST THAT ONLY GO AND DONE ARE SET
3336 015570 081410      BEQ  25
3337 015572 017737 163744 001566 MOV  %CSR,RCVDAT ; GET CSR ENTRY
3338 015600 012737 000201 001570 MOV  #201,XMTDAT ; GET CORRECT RESULT
3339 015606 104011      ERROR1 ; ERROR! INCORRECT CSR DATA
3340 015610 000476      BR  135 ; EXIT
3341
3342
3342      ; TEST THAT THERE IS 16. VALID DATA ENTRIES

```

```

3343 015612 012701 001600      25      MOV      #TUMTAB,X1      ;GET TUMBLE TABLE BASE ADDRESS
3344 015616 012702 000020      MOV      #16,X2      ;GET SCAN COUNT
3345 015622 005721      35      TST      (1)+      ;TEST FOR VALID DATA ENTRY
3346 015624 100402      BMI      45      ;BRANCH IF VALID DATA ENTRY FOUND
3347 015626 104001      ERROR      ;ERROR! MISSING VALID DATA ENTRY
3348 015630 000466      BR      135      ;EXIT
3349 015632 005302      45      DEC      X2      ;DECREMENT SCAN COUNT
3350 015634 001372      BNE      35      ;BRANCH IF 16. ENTRIES NOT SCANNED
3351
3352      ;TEST THAT THE BREAK BIT IS SET IN 16. TUMBLE TABLE ENTRIES
3353 015636 012701 001600      MOV      #TUMTAB,X1
3354 015642 012702 000020      MOV      #16,X2
3355 015646 032721 040000      55      BIT      #BIT14,(1)+      ;BREAK BIT SET?
3356 015652 001002      BNE      65      ;BRANCH IF SET
3357 015654 104001      ERROR      ;ERROR! MISSING BREAK BIT
3358 015656 000453      BR      135      ;EXIT
3359 015660 005302      65      DEC      X2      ;DECREMENT SCAN COUNT
3360 015662 001371      BNE      55
3361
3362      ;TEST THAT THE TUMBLE TABLE DATA BYTE IS ALL 0'S
3363 015664 012701 001600      MOV      #TUMTAB,X1
3364 015670 012702 000020      MOV      #16,X2
3365 015674 105721      75      TSTB      (1)+      ;TEST DATA BYTE
3366 015676 001402      BEQ      85      ;BRANCH IF 0'S
3367 015700 104001      ERROR      ;ERROR! INCORRECT DATA
3368 015702 000441      BR      135      ;EXIT
3369 015704 105721      85      TSTB      (1)+      ;STEP TABLE POINTER TO NEXT DATA BYTE
3370 015706 005302      DEC      X2
3371 015710 001371      BNE      75
3372
3373      ;CLEAR ALL BUT LINE NUMBER IN TUMBLE TABLE ENTRY
3374 015712 012701 001600      MOV      #TUMTAB,X1
3375 015716 012702 000020      MOV      #16,X2
3376 015722 042721 160777      95      BIC      #160777,(1)+      ;CLEAR ALL BUT LINE NUMBER
3377 015726 005302      DEC      X2
3378 015730 001374      BNE      95
3379
3380      ;TEST THAT THERE IS A TUMBLE TABLE ENTRY FOR EACH LINE
3381 015732 005004      CLR      X4      ;CLEAR LINE NUMBER
3382 015734 012703 000020      MOV      #16,X3
3383 015740 012702 000020      105      MOV      #16,X2
3384 015744 012701 001600      MOV      #TUMTAB,X1
3385 015750 020421      115      CMP      X4,(1)+      ;TEST FOR LINE ENTRY FOR THIS LINE
3386 015752 001410      BEQ      125      ;BRANCH IF FOUND
3387 015754 005302      DEC      X2
3388 015756 001374      BNE      115
3389 015760 010437 001570      MOV      X4,XMTDAT
3390 015764 010437 001566      MOV      X4,RCVDAT
3391 015770 104011      ERROR1      ;ERROR! NO LINE ENTRY FOUND FOR THIS LINE
3392 015772 000405      BR      135      ;EXIT
3393 015774 005303      125      DEC      X3      ;ALL LINES BEEN FOUND
3394 015776 001403      BEQ      135      ;EXIT IF YES
3395 016000 062704 001000      ADD      #1000,X4      ;SEARCH FOR
3396 016004 000755      BR      105      ;NEXT LINE
3397 016006 104006      135      SCOPE      ;SCOPE
3398
;*****

```

3399 016010 000137
3400 016012 016026
3401 016014 000002
3402 016016 016020
3403 000137
3404
3405
3406
3407
3408 016020 004537 005602
3409 016024 000040
3410
3411 016026 000140
3412 016030 016044
3413 016032 000002
3414 016034 016036
3415 000140
3416
3417
3418
3419
3420 016036 004537 005602
3421 016042 000020
3422
3423 016044 000141
3424 016046 016062
3425 016050 000002
3426 016052 016054
3427 000141
3428
3429
3430
3431
3432 016054 004537 005602
3433 016060 000010
3434
3435 016062 000142
3436 016064 016100
3437 016066 000002
3438 016070 016072
3439 000142
3440
3441
3442
3443
3444 016072 004537 005602
3445 016076 000004
3446
3447 016100 000143
3448 016102 016116
3449 016104 000002
3450 016106 016110
3451 000143
3452
3453
3454

```
RT137: 137          ; ROUTINE # 137
        RT140        ; ADDR OF NEXT ROUTINE
        2            ; ITERATION COUNT
        RT137A       ; SCOPE ENTRY POINT.
        X=X+1
; *****
; TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
; NEXT LINE.
RT137A: JSR          5, @NDLYXMT ; GO DO TEST. DELAY
        32.          ; THIS MUCH BETWEEN LINES
; *****
RT140: 140          ; ROUTINE # 140
        RT141        ; ADDR OF NEXT ROUTINE.
        2            ; ITERATION COUNT
        RT140A       ; SCOPE ENTRY POINT.
        X=X+1
; *****
; TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
; NEXT LINE.
RT140A: JSR          5, @NDLYXMT ; GO DO TEST. DELAY
        16.          ; THIS MUCH BETWEEN LINES
; *****
RT141: 141          ; ROUTINE # 141
        RT142        ; ADDR OF NEXT ROUTINE.
        2            ; ITERATION COUNT
        RT141A       ; SCOPE ENTRY POINT.
        X=X+1
; *****
; TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
; NEXT LINE.
RT141A: JSR          5, @NDLYXMT ; GO DO TEST. DELAY
        8.           ; THIS MUCH BETWEEN LINES
; *****
RT142: 142          ; ROUTINE # 142
        RT143        ; ADDR OF NEXT ROUTINE.
        2            ; ITERATION COUNT
        RT142A       ; SCOPE ENTRY POINT.
        X=X+1
; *****
; TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
; NEXT LINE.
RT142A: JSR          5, @NDLYXMT ; GO DO TEST. DELAY
        4            ; THIS MUCH BETWEEN LINES
; *****
RT143: 143          ; ROUTINE # 143
        RT144        ; ADDR OF NEXT ROUTINE.
        2            ; ITERATION COUNT
        RT143A       ; SCOPE ENTRY POINT.
        X=X+1
; *****
; TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
```

```

3455
3456 016110 004537 005602 ;NEXT LINE.
3457 016114 000002 RT143A: JSR 5, @DLYXMT ;GO DO TEST. DELAY
3458 ;*****
3459 016116 000144 ;THIS MUCH BETWEEN LINES
3460 016120 016134 RT144: 144 ;ROUTINE # 144
3461 016122 000002 RT145 ;ADDR OF NEXT ROUTINE.
3462 016124 016126 2 ;ITERATION COUNT
3463 000144 RT144A ;SCOPE ENTRY POINT
3464 X=X+1
3465 ;*****
3466 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
3467 ;NEXT LINE.
3468 016126 004537 005602 RT144A: JSR 5, @DLYXMT ;GO DO TEST. DELAY
3469 016132 000001 1 ;THIS MUCH BETWEEN LINES
3470 ;*****
3471 016134 000145 RT145: 145 ;ROUTINE # 145
3472 016136 016302 RT146 ;ADDR OF NEXT ROUTINE.
3473 016140 000144 100 ;ITERATION COUNT
3474 016142 016144 RT145A ;SCOPE ENTRY POINT.
3475 000145 X=X+1
3476 ;*****
3477 ;TEST THAT THE DM11 WORKS PROPERLY WHEN THE HALF-DUPLEX BIT (CSR BIT 1)
3478 ;IS SET. THE TEST TRANSMITS DATA ON LINE 0, AND 'BREAKS' ON LINE 1. ONLY
3479 ;THE BREAK SHOULD BE RECEIVED ON LINE 0 IN THE TUMBLE TABLE.
3480
3481 016144 005037 001600 RT145A: CLR TUMTAB ;CLEAR THE FIRST TWO
3482 016150 005037 001602 CLR TUMTAB+2 ;TUMBLE TABLE ENTRIES
3483 016154 012737 017102 001400 MOV @OUTBUF,CAT ;SET UP TO
3484 016162 012737 177777 001440 MOV @-1,WCT ;TRANSMIT 1 CHARACTER
3485 016170 012777 000007 163344 MOV @7,@CSR ;SET GO, HALF DUPLEX & MAINT BITS
3486 016176 012777 000001 163340 MOV @LBITO,@BAR ;TRANSMIT 1 CHAP. ON LINE 0
3487 016204 012777 000002 163334 MOV @LBITI,@BKCSR ;SET BREAK ON LINE 1
3488 016212 105777 163324 TSTB @CSR ;WAIT FOR THE CHARACTER
3489 016216 100375 BPL -4 ;TO BE RECEIVED
3490 016220 005077 163322 CLR @BKCSR ;CLEAR THE BREAK BIT ON LINE 1
3491
3492 ;TEST THAT ONLY THE BREAK WAS RECEIVED
3493 016224 022737 141000 001600 CMP #141000,TUMTAB ;TOST FOR BREAK ENTRY (LINE 1)
3494 016232 001410 BEQ 15
3495 016234 013737 001600 001566 MOV TUMTAB,RCVDAT ;GET ACTUAL ENTRY
3496 016242 012737 141000 001570 MOV #141000,XMTDAT ;GET CORRECT ENTRY
3497 016250 104011 ERROR1 ;ERROR! INCORRECT BREAK ENTRY
3498 016252 000407 BR 25 ;GO TO EXIT
3499 016254 013737 001602 001566 15: MOV TUMTAB+2,RCVDAT ;TEST THAT NEXT ENTRY IS CLEAR
3500 016262 001403 BEQ 25 ;EXIT IF CORRECT
3501 016264 005037 001570 CLR XMTDAT
3502 016270 104011 ERROR1 ;ERROR! SECOND ENTRY WAS NOT CLEAR
3503 016272 005777 163244 25: TST @CSR ;WAIT FOR THE TRANSMITTER
3504 016276 100375 BPL -4 ;TO FINISH
3505 016300 104006 SCOPE ;SCOPE
3506 ;*****
3507 016302 000146 RT146: 146 ;ROUTINE # 146
3508 016304 177777 RTLAST ;ADDR OF NEXT ROUTINE.
3509 016306 000144 100 ;ITERATION COUNT
3510 016310 016312 RT146A ;SCOPE ENTRY POINT.
  
```

```

3511          000146
3512          X=X+1
3513          ;*****
3514          ;TEST THAT THE DM11 RESPONDS CORRECTLY TO A RESET
3515 016312 012737 017102 001400 RT146A: MOV #OUTBUF,CAT ;SET UP TO TRANSMIT 10
3516 016320 012737 177770 001440      MOV 8-10,WCT ;CHARACTERS ON LINE 0
3517 016326 013737 004366 016376      MOV TIME1,25 ;GET TIME TO TRANSMIT 2 CHARACTERS
3518 016334 013737 004366 016442      MOV TIME1,65
3519 016342 005037 001570      CLR XMTDAT
3520 016346 012777 000007 163166      MOV #7,@CSR ;SET MAINT., HALF DUPLEX & GO BITS
3521 016354 012777 000001 163162      MOV #LBIT0,@BAR ;START TO TRANSMIT ON LINE 0
3522 016362 012777 000002 163156      MOV #LAIT1,@BKCSR ;BREAK ON LINE 1
3523 016370 012704 000002      MOV #2,%4
3524 016374 104400      15: DELAY ;WAIT 2 CHARACTER
3525 016376 000000      25: OPEN ;TIMES
3526 016400 005304      DEC %4
3527 016402 001374      BNE 15
3528 016404 104005      SRESET ;RESET
3529 016406 017737 163130 001566      MOV @CSR,RCVDAT ;GET CSR CONTENTS
3530 016414 001401      BEQ 35 ;BRANCH IF 0
3531 016416 104011      ERROR1 ;ERROR! CSR DID NOT CLEAR
3532 016420 017737 163120 001566 35: MOV @BAR,RCVDAT ;GET BAR CONTENTS
3533 016426 001402      BEQ 45 ;BRANCH IF 0
3534 016430 104011      ERROR1 ;ERROR! BAR IS NOT CLEAR
3535 016432 000427      BR 55 ;EXIT
3536 016434 012704 000010      45: MOV #8,%4
3537 016440 104400      55: DELAY ;WAIT 8 MORE CHARACTER TIMES
3538 016442 000000      65: OPEN
3539 016444 005304      DEC %4
3540 016446 001374      BNE 55
3541 016450 017737 163066 001566      MOV @CSR,RCVDAT ;TEST THAT CSR IS CLEAR
3542 016456 001402      BEQ 75
3543 016460 104011      ERROR1 ;ERROR! CSR WAS NOT CLEAR
3544 016462 000413      BR 95 ;GO TO EXIT
3545 016464 017737 163054 001566 75: MOV @BAR,RCVDAT ;TES THAT BAR IS CLEAR
3546 016472 001402      BEQ 85
3547 016474 104011      ERROR1 ;ERROR! BAR DID NOT CLEAR
3548 016476 000405      BR 95
3549 016500 017737 163042 001566 85: MOV @BKCSR,RCVDAT ;TEST THAT BKCSR IS CLEAR
3550 016506 001401      BEQ 95
3551 016510 104011      ERROR1 ;ERROR! BKCSR DID NOT CLEAR
3552 016512 104006      95: SCOPE
  
```

Address	Hex	ASCII	Label	Operation	Comments
3553					
3554	016514				
3555	016514	104000			
3556	016516	020117			
3557	016520	004737	016720		
3558	016524	004737	016772		
3559	016530	005777	163010		
3560	016534	001375			
3561	016536	005077	163000		
3562	016542	005037	016570		
3563	016546	017737	162430	016570	
3564	016554	042737	000377	016570	
3565	016562	000337	016570		
3566	016566	104400			
3567	016570	000000			
3568	016572	000754			
3569					
3570					

Label	Operation	Comments
PRG1-	TRANSMITTER SCOPE LOOP	
PRG1:		
	TYPE	
	PRG1M	
	JSR	7, PARAM
PRG1R:	JSR	7, LOOP
PRG1B:	TST	2BAR
	BNE	PRG1B
PRG1C:	CLR	2CSR
	CLR	PRG1D
	MOV	2SWR, PRG1D
	BIC	#377, PRG1D
	SWAB	PRG1D
	DELAY	
PRG1D:	OPEN	
	BR	PRG1R

Comments
; BEGIN
; TYPE PROGRAM TITLE
; GO GET USER PARAMETERS
; GO LOOP TRANSMITTER
; WAIT FOR ALL LINES TO FINISH
; BRANCH IF NOT DONE
; CLEAR THE CSR
; CLEAR DELAY TIME
; GET DELAY
;; ++D
;; ++D
;; ++D
; DELAY AS SPECIFIED
; BY USER
; LOOP BACK

```

3571
3572
3573 016574          ,PRG2- RECEIVER SCOPE LOOP
3574 016574 104000    PRG2:
3575 016576 020135    TYPE
3576 016600 004737 016720    PRG2M
3577 016604 004737 016772    JSR 7,PARAM
3578 016610 012777 000001 162724    PRG2R: JSR 7,LOOP
3579 016616 005777 162722    PRG2A: MOV #BIT0, @CSR
3580 016622 001415    PRG2AA: TST @BAR
3581 016624 105777 162712    BEQ PRG2B
3582 016630 100372    TSTB @CSR
3583 016632 042777 000200 162702    BPL PRG2AA
3584 016640 020127 001776    BIC #BIT7, @CSR
3585 016644 001002    CMP #1, @TUMTAB+176
3586 016646 012701 001576    BNE +6
3587 016652 005721    MOV @TUMTAB-2, %1
3588 016654 000755    TST (1)+
3589 016656 005077 162660    BR PRG2A
3590 016662 005077 162656    PRG2B: CLR @CSR
3591 016666 005037 016714    CLR @BAR
3592 016672 017737 162304 016714    CLR PRG2C
3593 016700 042737 000377 016714    MOV @SWR, PRG2C
3594 016706 000337 016714    BIC #377, PRG2C
3595 016712 104400    SWAB PRG2C
3596 016714 000000    DELAY
3597 016716 000732    PRG2C: OPEN
3598
3599
3600
3601 016720 104000    ,SUBROUTINE TO GET USER PARAMETERS (FOR PRG1 & 2)
3602 016722 020154    PARAM: TYPE
3603 016724 004537 004416    LINPAR
3604 016730 000000    JSR 5, RECD
3605 016732 104000    LINE: 0
3606 016734 020177    PARAM: TYPE
3607 016736 004537 004416    HOWMAN
3608 016742 000000    JSR 5, RECD
3609 016744 023727 016742 000310    CHARS. 0
3610 016752 101403    CMP CHARS, #200
3611 016754 104000    BLOS PARAMB
3612 016756 020017    TYPE
3613 016760 000764    M1
3614 016762 104000    BR PARAMA
3615 016764 020043    PARAMB: TYPE
3616 016766 104015    M4
3617 016770 000207    CNTRLU
3618
3619
3620
3621 016772 117737 162204 017102    ,SUBROUTINE TO TRANSMIT DATA FROM THE SR
3622 017000 004537 006224    LOOP: MOV @SWR, OUTBUF
3623 017004 017102    JSR 5, BMOVE
3624 017006 017103    OUTBUF
3625 017010 000307    OUTBUF+1
3626 017012 012777 001400 162530    199.
    MOV #CAT, @BASREG

```

```

; BEGIN
; TYPE PROGRAM
; TITLE
; GO GET USER PARAMETERS
; GO START TRANSMITTER
; SET GO, CLEAR THE OTHERS
; HAVE ALL LINES SELECTED FINISHED
; BRANCH IF FLAG SHED TRANSMITTING
; WAIT FOR THE RECEIVER TO
; RECEIVE A CHARACTER
; CLEAR RECEIVER FLAG
; IS THE POINTER AT THE END OF THE TT
; BRANCH IF NOT
; RESET POINTER
; INCREMENT POINTER
; GO BACK & TEST TRANSMITTER FLAG
; CLEAR THE CSR
; CLEAR THE BAR
; CLEAR USER DELAY
; GET USER DELAY
; DELAY AS SPECIFIED
; BY USER
; REPEAT LOOP

```

.. ++D
.. ++D
.. ++D

```

; ASK USER WHICH LINE
; TO TEST
; GET LINE AND PUT IT
; HERE
; ASK USER HOW MANY
; CHARACTERS TO TRANSMIT
; GET CHARS AND PUT IT
; HERE
; LIMIT RESPONSE TO 200.
; (CORE LIMITATION)
; RE-REQUEST PARAMETER
; TYPE INSTRUCTIONS
; GO GET VALUE
; EXIT

```

```

; FILL OUTPUT
; BUFFER
; WITH
; DATA TO BE
; TRANSMITTED
; INITIALIZE BASE REGISTER

```

```

3627 017020 012737 017102 001400      MOV      #OUTBUF,CAT      ;LOAD CURRENT
3628 017026 004537 006224      JSR      5,BMOVE      ;ADDRESS TABLE
3629 017032 001400      CAT      ;WITH ADDRESS
3630 017034 001402      CAT+2      ;OF OUTPUT BUFFER
3631 017036 000040      32.
3632 017040 013737 016742 001440      MOV      CHARS,WCT      ;LOAD WORD COUNT
3633 017046 005437 001440      NEG      WCT      ;FORM TWO'S COMPLEMENT
3634 017052 004537 006224      JSR      5,BMOVE      ;TABLE WITH
3635 017056 001440      WCT      ;NUMBER OF
3636 017060 001442      WCT+2      ;CHARACTERS TO BE
3637 017062 000040      32.      ;TRANSMITTED
3638 017064 013737 016730 001564      MOV      LINE,LINBIT      ;SAVE LINES TO BE TRANSMITTED ON
3639 017072 013777 016730 162444      MOV      LINE,ABAR      ;START TRANSMITTING ON SELECTED LINES
3640 017100 000207      RTS      7      ;EXIT
3641
3642
3643 017102 000000      OUTBUF: 0      ;FIRST ADDRESS OF 100
3644      017246      ;=OUTBUF+100.      ;CHARACTER OUTPUT BUFFER
3645 017246 000000      INBUF: 0      ;FIRST ADDRESS OF 100.
3646      017412      ;=INBUF+100.      ;CHARACTER INPUT BUFFER (WHERE RECEIVED
3647      ;DATA IS STORED)
3648
3649 017412 013746 000006      SUSWRR: MOV      @B6,-(SP)      ;SAVE VECTORS
3650 017416 013746 000004      MOV      @B4,-(SP)
3651 017422 012737 017442 000004      MOV      @B5,@B4
3652 017430 022777 177777 161544      CMP      B-1,@SWR      ;SET UP FOR TIMEOUT
3653 017436 001402      BEQ      25      ;REFERENCE HARDWARE SWITCH REGISTER
3654 017440 000407      BR      35
3655 017442 022626      15:      CMP      (SP)+,(SP)+      ;ADJUST STACK
3656 017444 012737 000176 001202      25:      MOV      @SWREG,SWR      ;POINT TO SOFTWARE SWITCH REG
3657 017452 012737 000174 001204      MOV      @DISPREG,DISPLAY      ;POINT TO SOFT DISPLAY REG
3658 017460 012637 000004      35:      MOV      (SP)+,@B4
3659 017464 012637 000006      MOV      (SP)+,@B6      ;RESTORE VECTORS
3660 017470 000002      RTI
3661
3662
3663      ;ROUTINE TO CHECK FOR G BEING TYPED
3664
3665 017472 022737 000176 001202      KBDINTT: CMP      @SWREG,SWR
3666 017500 001021      BNE      15
3667 017502 023737 000042 000046      CMP      @B42,@B46      ;ACT11?
3668 017510 001415      BEQ      15      ;BR IF YES
3669 017512 005037 017604      CLR      TMP1      ;CLEAR TEMP AREA
3670 017516 117737 162346 017604      MOV      @TKDBR,TMP1      ;FETCH THE BUFFER
3671 017524 142737 000200 017604      BICB      @200,TMP1      ;STRIP OFF PARITY
3672 017532 122737 000007 017604      CNPB      @7,TMP1      ;WAS IT G
3673 017540 001001      BNE      15      ;NO
3674 017542 104015      CNTLU      ;GO CHANGE IT
3675 017544 000002      15:      RTI      ;EXIT
3676
3677
3678      ;ROUTINE TO CHANGE CONTENTS OF SWREG(LOC 176)
3679
3680 017546 022737 000176 001202      CNTLUU: CMP      @SWREG,SWR
3681 017554 001023      BNE      FJX
3682 017556 104000      TYPE
  
```


Line	Address	Offset	Value	Field	Encoding	Text
3697						MESSAGES
3698	017626	053045	041505	020124	WHERE	ASCII '%VECT ADR?'
3699	017634	042101	037522	100		
3700	017641	045	053523	036522	SSWREG:	ASCII '%SWR='
3701	017646	100				
3702	017647	040	020040	020040	SVALUE:	ASCII 'NEW='
3703	017654	020040	020040	047040		
3704	017662	053505	040075			
3705	017666	044445	041516	051117	SCTLU	ASCII '%INCORRECT INPUT, TRY AGAIN'
3706	017674	042522	052103	044440		
3707	017702	050116	052125	020054		
3708	017710	051124	020131	043501		
3709	017716	044501	020516			
3710	017722	036445	100			
3711	017725	045	047125	052111	WHICH:	ASCII '%='
3712	017732	024043	024470	040077		ASCII '%UNIT#(8)?'
3713	017740	041445	040510	020122	LEVEL	ASCII '%CHAR LGTH'
3714	017746	047114	052107	040110		
3715	017754	042445	051122	051440	ERDAT	ASCII '%ERR S/B'
3716	017762	041057	020072			
3717	017766	020040	020040	020040	AASB:	ASCII 'WAS'
3718	017774	053440	051501	020072		
3719	020002	020040	020040	020040	AHAS	ASCII 'a'
3720	020010	100				
3721	020011	045	051120	021507	MO:	ASCII '%PRGA'
3722	020016	100				
3723	020017	045	040077		M1:	ASCII '%?'
3724	020022	042445	042116	040040	M2:	ASCII '%END a'
3725	020030	051445	036522	037460	M3:	ASCII '%SR=0? GO a'
3726	020036	043440	027117	100		
3727	020043	045	042114	041440	M4:	ASCII '%LD CHAR IN SRO-7, DLY IN SR8-15a'
3728	020050	040510	020122	047111		
3729	020056	051440	030122	033455		
3730	020064	042073	054514	044440		
3731	020072	020116	051123	026470		
3732	020100	032461	100			
3733	020103	045	047514	044507	PRGOM:	ASCII '%LOGIC TSTSa'
3734	020110	020103	051524	051524		
3735	020116	100				
3736	020117	045	046530	052111	PRGIM:	ASCII '%XMITTER LOOPa'
3737	020124	042524	020122	047514		
3738	020132	050117	100			
3739	020135	045	046530	052111	PRG2M:	ASCII '%XMIT/REC LOOPa'
3740	020142	051057	041505	046040		
3741	020150	047517	040120			
3742	020154	052045	050131	046040	LINPAR:	ASCII '%TYP LINES TO TST a'
3743	020162	047111	051505	052040		
3744	020170	020117	051524	020124		
3745	020176	100				
3746	020177	045	047443	020106	HOWMAN:	ASCII '%#OF CHARS?a'
3747	020204	044103	051101	037523		
3748	020212	100				
3749	020213	045	020122		EMO:	ASCII '%R'
3750	020216	020040	050040	036503	ATNUMB:	ASCII 'PC='
3751	020224	020040	020040	020040	APC:	ASCII 'a'
3752	020232	100				

3753	020233	015	012	
3754	020235	124	042510	050440
3755	020242	044525	045503	041040
3756	020250	047522	047127	043040
3757	020256	054117	045040	046525
3758	020264	042520	020104	053117
3759	020272	051105	052040	042510
3760	020300	046040	055101	020131
3761	020306	047504	051507	041040
3762	020314	041501	020113	031061
3763	020322	032063	033065	034067
3764	020330	030071		
3765		000001		

MSG1 . BYTE 15,12
ASCII 'THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 1234567890'

END

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

PRTY6 = 000300	473#												
PRTY7 = 000340	472#	541	543	843	847	1010	1130	1138	1264	1268	1301	1321	1324
PRVCNT 002104	2147	2561	2568	2647									
PSH = 177776	737#	1907#	1918	1982#	1994								
	433#	1010#	1032#	1133#	1138#	1262#	1264#	1268#	1301#	1319#	1321#	1324#	2132#
PTO 006100	2147#	2150#	2155#	2559#	2561#	2568#							
PT1 006102	1470#	1474	1480#										
PWRUP 004606	1471#	1474#	1475#	1478#	1479#	1480	1481						
RCV DAT 001566	1214	1218#											
	697#	751	805	1245#	1252#	1256#	1257	1331#	1332#	1337	1344#	1356#	1361#
	1362#	1367	1399#	1400#	1435#	1442#	1579#	1598#	1604#	1610#	1617#	1854#	1860#
	1877#	1878	1886#	1911#	1912	1936#	1952#	1953	1961#	1987#	1988	2021#	2038#
	2044#	2062#	2069#	2086#	2093#	2110#	2117#	2547#	2563#	2622#	2663#	2698#	2699#
	2702	2729#	2732	2771#	2778#	2779	2803#	2830#	2837#	3244#	3279#	3280	3304#
	3337#	3390#	3495#	3499#	3529#	3532#	3541#	3545#	3549#				
RCVTST 005034	1298#	2350	2361	2372	2383	2394	2405	2416	2427	2438	2449	2460	2471
	2482	2493	2504	2515									
RCV0 012014	2346	2350#											
RCV1 012032	2357	2361#											
RCV10 012174	2434	2438#											
RCV11 012212	2445	2449#											
RCV12 012230	2456	2460#											
RCV13 012246	2467	2471#											
RCV14 012264	2478	2482#											
RCV15 012302	2489	2493#											
RCV16 012320	2500	2504#											
RCV17 012336	2511	2515#											
RCV2 012050	2368	2372#											
RCV3 012066	2379	2383#											
RCV4 012104	2390	2394#											
RCV5 012122	2401	2405#											
RCV6 012140	2412	2416#											
RCV7 012156	2423	2427#											
RECD 004416	838	1047	1070	1099	1173#	3603	3607	3690					
RIND 006076	1463#	1465	1466	1469#	1476#								
RNGEN 003250	968#	1908	1983										
RORPAR 006764	1571	1597	1619	1633#	1641								
RP1 003316	968	972#	979	981#									
RP2 003320	971	975	978#	982#									
RSTART 002024	713#	851											
RSTAT1 002426	670	832#											
RSTAT2 002514	672	846#											

RT102	014302	2903	2913#
RT103	014320	2914	2924#
RT104	014336	2925	2935#
RT105	014354	2936	2946#
RT106	014372	2947	2957#
RT107	014410	2958	2968#
RT11	007720	1824	1842#
RT11A	007730	1845	1851#
RT110	014426	2969	2979#
RT111	014444	2980	2990#
RT112	014462	2991	3001#
RT113	014500	3002	3012#
RT114	014516	3013	3023#
RT115	014534	3024	3036#
RT116	014552	3037	3047#
RT117	014570	3048	3058#
RT12	010010	1843	1866#
RT12A	010020	1869	1875#
RT120	014606	3059	3069#
RT121	014624	3070	3080#
RT122	014642	3081	3091#
RT123	014660	3092	3102#
RT124	014676	3103	3113#
RT125	014714	3114	3124#
RT126	014732	3125	3135#
RT127	014750	3136	3146#
RT13	010144	1867	1898#
RT13A	010154	1901	1906#
RT130	014766	3147	3157#
RT131	015004	3158	3168#
RT132	015022	3169	3179#
RT133	015040	3180	3190#
RT134	015056	3191	3201#
RT135	015074	3202	3212#
RT135A	015104	3215	3226#
RT136	015474	3213	3314#
RT136A	015504	3317	3322#
RT137	016010	3315	3399#
RT137A	016020	3402	3408#
RT14	010256	1899	1924#
RT14A	010266	1927	1933#
RT140	016026	3400	3411#
RT140A	016036	3414	3420#
RT141	016044	3412	3423#
RT141A	016054	3426	3432#
RT142	016062	3424	3435#
RT142A	016072	3438	3444#
RT143	016100	3436	3447#
RT143A	016110	3450	3456#
RT144	016116	3448	3459#
RT144A	016126	3462	3468#
RT145	016134	3460	3471#
RT145A	016144	3474	3481#
RT146	016302	3472	3507#
RT146A	016312	3510	3515#
RT15	010316	1925	1941#

1857

RT15A	010326	1944	1950#
RT16	010452	1942	1973#
RT16A	010462	1976	1981#
RT17	010570	1974	2000#
RT17A	010600	2003	2008#
RT2	007252	1691	1709#
RT2A	007262	1712	1717#
RT20	010662	2001	2027#
RT20A	010672	2030	2035#
RT21	010750	2028	2049#
RT21A	010760	2052	2057#
RT22	011056	2050	2073#
RT22A	011066	2076	2081#
RT23	011164	2074	2097#
RT23A	011174	2100	2105#
RT24	011272	2098	2121#
RT24A	011302	2124	2129#
RT25	011336	2122	2139#
RT25A	011346	2142	2147#
RT26	011444	2140	2165#
RT27	011462	2166	2176#
RT3	007324	1710	1728#
RT3A	007334	1731	1736#
RT30	011500	2177	2187#
RT31	011516	2188	2198#
RT32	011534	2199	2209#
RT33	011552	2210	2220#
RT34	011570	2221	2231#
RT35	011606	2232	2242#
RT36	011624	2243	2253#
RT37	011642	2254	2264#
RT4	007376	1729	1747#
RT4A	007406	1750	1755#
RT40	011660	2265	2275#
RT41	011676	2276	2286#
RT42	011714	2287	2297#
RT43	011732	2298	2308#
RT44	011750	2309	2319#
RT45	011766	2320	2330#
RT46	012004	2331	2343#
RT47	012022	2344	2354#
RT5	007450	1748	1766#
RT5A	007460	1769	1774#
RT50	012040	2355	2365#
RT51	012056	2366	2376#
RT52	012074	2377	2387#
RT53	012112	2388	2398#
RT54	012130	2399	2409#
RT55	012146	2410	2420#
RT56	012164	2421	2431#
RT57	012202	2432	2442#
RT6	007522	1767	1785#
RT6A	007532	1788	1793#
RT60	012220	2443	2453#
RT61	012236	2454	2464#
RT62	012254	2465	2475#

RT63	012272	2476	2486#																
RT64	012310	2487	2497#																
RT65	012326	822#	2498	2508#															
RT66	012346	2509	2520#																
RT66A	012362	2523	2532#																
RT67	012606	2521	2577#																
RT67A	012616	2580	2585#																
RT7	007574	1786	1804#																
RT7A	007604	1807	1812#																
RT70	012716	822	2578	2603#															
RT70A	012726	2606	2614#																
RT71	013032	2604	2634#																
RT71A	013042	2637	2643#																
RT72	013226	2635	2677#																
RT72A	013236	2680	2687#																
RT73	013572	2678	2760#																
RT73A	013602	2763	2769#																
RT74	014006	2761	2812#																
RT74A	014016	2815	2822#																
RT75	014174	2813	2858#																
RT76	014212	2859	2869#																
RT77	014230	2870	2880#																
SAVREG=	104007	513#	1486	1505															
SAVRG	003132	723	932#																
SCOPE =	104006	512#	1274	1371	1403	1445	1630	1684	1707	1726	1745	1764	1783	1802					
		1821	1840	1864	1896	1922	1939	1971	1998	2019	2047	2071	2095	2119					
		2137	2163	2575	2601	2632	2675	2738	2810	2854	3312	3397	3505	3552					
SCOPEA	002736	889#	897																
SCOPEB	002742	888	890#																
SCOPEd	002776	868	891	895	898#														
SCOPEr	002012	708#	886	916#															
SPBOT	001200	576#	818	825	832	848	1011#	1012	1219										
SRESET=	104005	511#	1853	1935	2037	3528													
SRSET	002546	852#	1668																
SRSETT	003232	721	961#	963#															
START	002360	669	818#																
SUSAR =	104013	517#	819																
SUSARR	017412	727	3649#																
SVRPC	003166	932#	940	942#															
SVRPSW	003170	933#	939	943#															
SWR	001202	677#	767	782	791	852	865	869	887	890	898	1881	1889	1915					
SWREG	000176	1956	1964	1991	2665	3563	3592	3621	3652	3656#	3665	3680	3694#						
TIMEA	004272	666#	852	3656	3665	3680	3685												
TIMEB	004326	1134#	1143																
TIMEC	004336	1136	1141#																
TIMER	004216	1129	1144#																
TIME1	004366	1126#	1140	2532															
TIME14	004370	1126#	1135#	1145	1152#	1420	1430	2572	2597	2629	3241	3322	3517	3518					
TKCSR	002066	1145#	1146#	1148#	1153#	2544	2588	2794											
TKDSR	002070	730#	1176																
TMP1	017604	731#	1178	3670															
TPCSR	002072	3669#	3670#	3671#	3672	3691#	3694												
TPDSR	002074	732#	784	787	996	1195	1198	1201											
TTOAT	001562	733#	786#	995#	1180#	1197#	1200#												
TUNTAS	001600	695#	1574#	1575#	1576	1579													
		701#	702	1299#	1300#	1331	1343#	1344	1356	1361	1389	1393	1397	1400					

TYP	003322	1537	1538*	1540	1541	1583	1585	2699	2729	2822	2827	2851	3226*	3228
TYPA	003332	3229	3260	3273	3279	3288	3299	3323*	3325	3326	3343	3353	3363	3374
TYPC	003350	3384	3481*	3482*	3493	3495	3499	3584	3586					
TYPD	003366	716	984*											
TYPDAT	003432	987*	994	1003										
TYPE =	104000	989	991*											
		993	995*	1000	1002									
		987*	988	991	995	999*	1001*	1004*						
		506*	746	777	780	836	856	903	1045	1054	1068	1074	1097	1106
TYPEF	003404	1206	1208	1664	3555	3574	3601	3605	3611	3614	3682	3688		
TYPEG	003416	992	999*											
UNIT	003766	1001*												
UNTOKA	004006	1067	1071*	1072	1077*	1078*	1079*	1083						
UNTOKB	004037	1073	1077*											
UNTOKC	004062	1082*	1085											
VAC	001540	1090*	1093											
VECOK	003666	686*	700											
VECOKA	003676	1043	1050	1052*										
VECOKB	003704	1054*	1058	1060										
VECTOR	003650	1053	1057*											
WCT	001440	1040*	1048*	1049	1051*	1052	1057	1059	1061	1062*	1063			
		682*	683	1029*	1127*	1252	1522*	1589	2651*	2691*	2722*	2831*	3484*	3516*
WHERE	017626	3632*	3633*	3635	3636									
WHICH	017725	1046	3698*											
X =	000146	1067	3711*											
		525*	1672	1677*	1689	1694*	1708	1713*	1727	1732*	1746	1751*	1765	1770*
		1784	1789*	1803	1808*	1822	1827*	1841	1846*	1865	1870*	1897	1902*	1923
		1928*	1940	1945*	1972	1977*	1999	2004*	2026	2031*	2048	2053*	2072	2077*
		2096	2101*	2120	2125*	2138	2143*	2164	2169*	2175	2180*	2186	2191*	2197
		2202*	2208	2213*	2219	2224*	2230	2235*	2241	2246*	2252	2257*	2263	2268*
		2274	2279*	2285	2290*	2296	2301*	2307	2312*	2318	2323*	2329	2334*	2342
		2347*	2353	2358*	2364	2369*	2375	2380*	2386	2391*	2397	2402*	2408	2415*
		2419	2424*	2430	2435*	2441	2446*	2452	2457*	2463	2468*	2474	2479*	2485
		2490*	2496	2501*	2507	2512*	2518	2519	2524*	2576	2581*	2602	2607*	2633
		2638*	2676	2681*	2759	2764*	2811	2816*	2857	2862*	2868	2873*	2879	2884*
		2890	2895*	2901	2906*	2912	2917*	2923	2928*	2934	2939*	2945	2950*	2956
		2961*	2967	2972*	2978	2983*	2989	2994*	3000	3005*	3011	3016*	3022	3027*
		3035	3040*	3046	3051*	3057	3062*	3068	3073*	3079	3084*	3090	3095*	3101
		3106*	3112	3117*	3123	3128*	3134	3139*	3145	3150*	3156	3161*	3167	3172*
		3178	3183*	3189	3194*	3200	3205*	3211	3216*	3313	3318*	3398	3403*	3410
		3415*	3422	3427*	3434	3439*	3446	3451*	3458	3463*	3470	3475*	3506	3511*
XMITD	006246	1237	1303	1349	1418	1517*	1545	2615	2773	3236				
XMTDAT	001570	698*	751	801	1236*	1255*	1257	1333*	1334*	1335*	1336*	1337	1346*	1358*
		1363*	1364*	1365*	1366*	1367	1401*	1410*	1441*	1578*	1599*	1602*	1603*	1852*
		1875*	1876	1878	1883	1884*	1891*	1892	1894*	1909*	1910	1912	1934*	1950*
		1951	1953	1958	1959*	1966*	1967	1969*	1985*	1986	1988	2020*	2024*	2035*
		2061*	2068*	2085*	2092*	2109*	2116*	2549*	2562*	2623*	2662*	2700*	2701*	2702
		2730*	2731*	2732	2772*	2789*	2804*	2829*	2838*	2845	3246*	3277*	3278*	3280
		3296*	3300	3310*	3338*	3389*	3496*	3501*	3519*					
XMTINT	001556	693*	1063*	1260*	2129*	2148*	2156*	2541*						
XMTLVL	001560	694*	2155											
XMTTST	004622	1234*	2172	2183	2194	2205	2216	2227	2238	2249	2260	2271	2282	2293
		2304	2315	2326	2337									
		524*	2164	2174*	2175	2185*	2186	2196*	2197	2207*	2208	2218*	2219	2229*
Y =	000020	2230	2240*	2241	2251*	2252	2262*	2263	2273*	2274	2284*	2285	2295*	2296
		2306*	2307	2317*	2318	2328*	2329	2339*	2340*	2342	2352*	2353	2363*	2364

		2374#	2375	2385#	2386	2396#	2397	2407#	2408	2418#	2419	2429#	2430	2440#
		2441	2451#	2452	2462#	2463	2473#	2474	2484#	2485	2495#	2496	2506#	2507
		2517#	2856#	2857	2867#	2868	2878#	2879	2889#	2890	2900#	2901	2911#	2912
		2922#	2923	2933#	2934	2944#	2945	2955#	2956	2966#	2967	2977#	2978	2988#
		2989	2999#	3000	3010#	3011	3021#	3022	3032#	3034#	3035	3045#	3046	3056#
		3057	3067#	3068	3078#	3079	3089#	3090	3100#	3101	3111#	3112	3122#	3123
		3133#	3134	3144#	3145	3155#	3156	3166#	3167	3177#	3178	3188#	3189	3199#
		3200	3210#											
		1209	3705#											
		659	908#											
		3683	3700#											
		3686	3689	3702#										
		527#	528	530	532	534	536	538	544	546	548	550	552	554
		556	558	560	562	564	566	568	570	572	574	576	578	580
		582	584	586	588	590	592	594	596	598	600	602	604	606
		608	610	612	614	616	618	620	622	624	626	628	630	632
		634	636	638	640	642	644	646	648	650	652	654	658#	660#
		664#	668#	675#	679#	681#	683#	685#	700#	702#	785	788	824	997
		1142	1242	1307	1352	1354	1383	1386	1456	1477	1558	1561	1590	2618
		2655	2658	2694	2696	2725	2727	2848	2850	3331	3489	3504	3585	3644#
		3646#												
		739#	2090#											
		741#												
		740#	2114#											
		738#	1089	2066#										
SCTLU	017666													
SENDAD	003036													
SSUREG	017641													
SVALUE	017647													
=	020332													
BAR	002110													
BASRE	002114													
BKCSR	002112													
CSR	002106													

ABS. 020332 000

ERRORS DETECTED: 0

DSKZ: CZDMAD, DSKZ: CZDMAD. SEQ=DSKZ: CZDMAD. P11

RUN-TIME: 7 11 1 SECONDS

RUN-TIME RATIO: 322/20=15.8

CORE USED: 13K (25 PAGES)

DOCUMENT PAGES: 85