

DM11A

LOGIC TEST
CZDMA D0

AH-8537D-MC

COPYRIGHT © 72-78

FICHE 1 OF 1

AUG 1978

digital

MADE IN USA

This microfiche card contains a grid of frames, each displaying logic test data. The data is organized into columns and rows, with some frames containing additional headers or footers. The frames are arranged in a grid that is approximately 10 frames wide and 15 frames high. The data appears to be organized into several sections, with some frames containing more detailed information than others. The overall layout is consistent with a microfiche card used for storing digital data.

. 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 CLRB 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 LINES.

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 80 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 DZDMA 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 CZDMADO DM11A LGC TST
. LIST ME, BIN, SEQ
. ENABLE ABS, AMA

413		;CZDMADO 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	
435	000240	;ADDRESS OF ERROR TRAP VECTOR
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	POPS=5726
471	022626	POPS2=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	R0=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	.+2	
635	000326	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
636	000330	000332	.+2	
637	000332	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
638	000334	000336	.+2	
639	000336	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
640	000340	000342	.+2	
641	000342	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
642	000344	000346	.+2	
643	000346	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
644	000350	000352	.+2	
645	000352	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
646	000354	000356	.+2	
647	000356	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
648	000360	000362	.+2	
649	000362	000300	HALT	; TRAPPED TO PREVIOUS ADDRESS.
650	000364	000366	.+2	
651	000366	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
652	000370	000372	.+2	
653	000372	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
654	000374	000376	.+2	
655	000376	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
656				

657						
658		000046				
659	000046	003036		. =46		
660		000052		SENDAD		
661	000052	060000		. =52		
662				60000		
663						
664		000174		. =174		
665	000174	000000		DISPREG: 0		
666	000176	000000		SWREG: 0		
667						
668		000200		. =200		
669	000200	000137	002360	JMP	2#START	
670	000204	000137	002426	JMP	2#RSTAT1	; GO TO START OF DIAGNOSTIC.
671						; GO GET PROGRAM # & RESTART PROGRAM
672	000210	000137	002514	JMP	2#RSTAT2	; USING PREVIOUS DM11 PARAMETERS
673						; RESTART PREVIOUS PROGRAM USING
674						; PREVIOUS DM11 PARAMETERS
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: PRTYS		
693	001556	000000		XMTINT: OPEN		
694	001560	000240		XMTLVL: PRTYS		
695	001562	000000		TTOAT: OPEN		
696	001564	000000		LINBIT: OPEN		
697	001566	000000		RCVDT: OPEN		
698	001570	000000		XMTDT: OPEN		
699	001572	000000		CARMSK: OPEN		
700		001600		. =VAC+32		
701	001600	000000		TUNTAB: OPEN		
702		002000		. =TUNTAB+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		

; 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
; TUNBLE TABLE DATA
; LINE BIT (FOR BAR)

; STARTING ADDRESS OF
; TUNBLE 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
 EMTTAB: 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 ;ROUTINE TO TYPE OUT INCORRECT ROUTINE SELECTED
746 002120 104000 INCRTN: TYPE
747 002122 020017 M1
748 002124 000207 RTS %7 ;TYPE INCORRECT ROUTINE SELECTED.
749 ;EXIT.
750
751 002126 123737 001566 001570 ;DATA COMPARISON ROUTINE
752 002134 001403 DTCHK: CMPB RCVDAT,XMTDAT ;COMPARE RECEIVED & TRANSMITTED DATA
753 002136 004737 002332 BEQ 15 ;CHARS. BRANCH IF SAME.
754 002142 104011 JSR 7,CNV DAT ;CONVERT RCVDAT & XMTDAT TO ASCII
755 002144 000002 ERROR1 ;EXIT.
756 RTI
757
758 ;ERROR ROUTINE WHENEVER THE PROGRAM DETECTS AN ERROR THE ERROR
759 ;AND ERROR1 ENT INSTRUCTIONS ENTER HERE. ERROR AT ERR:,AND
760 002146 012737 000402 002246 ERR: MOV #402,ERRB ;MOV BR.+6 TO ERRB
761 002154 013737 002100 002102 MOV PCADD,APCADD ;GET PC WHERE ERROR OCCURRED
762 002162 000410 BR ERRA
763 002164 012737 000240 002246 ERR1: MOV #240,ERRB ;MOVE NOP TO ERRB
764 002172 013737 002100 002102 MOV PCADD,APCADD ;GET PC WHERE ERROR OCCURRED
765 002200 004737 002332 JSR 7,CNV DAT ;CONVERT RCVDAT & XMT DAT TO ASCII
766 002204 104014 KBDIN ;GO CHECK FOR G
767 002206 032777 020000 176766 BIT #BIT13,ASHR ;ERROR PRINTOUT DESIRED
768 002214 001017 BNE ERRC ;BRANCH IF NO PRINTOUT
769 002216 004537 006154 JSR 5,OACNV ;CONVERT
770 002222 002102 APCADD ;DATA
771 002224 020224 APC ;TO
772 002226 000006 6 ;ASCII
773 002230 004537 006154 JSR 5,OACNV ;FOR
774 002234 002004 RTNNO ;PRINTOUT
775 002236 020216 ATNUMB
776 002240 000003 3
777 002242 104000 TYPE
778 002244 020213 EMO ;TYPE ERROR
779 002246 000000 OPEN ;MESSAGE
780 002250 104000 TYPE ;NOP IF ERROR1, BR.+6 IF ERROR
781 002252 017754 ERDAT ;TYPE ANOTHER MESSAGE
782 002254 032777 000400 176720 ERRC: BIT #BIT8,ASHR ;IF ERROR 1
783 002262 001411 BEQ ERRD ;RING BELL ON ERROR?
784 002264 105777 177602 TSTB @TPCSR ;BRANCH IF NO BELL ON ERROR
785 002270 100375 BPL -4 ;TELEPRINTER
786 002272 012777 000007 177574 MOV #BELL,@TPDBR ;READY?
787 002300 105777 177566 TSTB @TPCSR ;RING THE BELL
788 002304 100375 BPL -4 ;WAIT FOR THE BELL TO RING
789 002306 023737 000042 000046 ERRD: CMP #42,#46 ;ACT11?
790 002314 001403 BEQ ERRLT ;HALT ON ERROR
791 002316 005777 176660 TST ASHR ;GO TO EXIT IF NO HALT ON ERROR
792 002322 100001 BPL ERREX
793 002324 000000 ERRHLT: HALT
794 002326 104014 ERREX: KBDIN
795 002330 000002 RTI ;CHECK FOR G
796 ;RETURN
797
798 ;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, OACNV
XMTDAT
AASB
6
JSR 5, OACNV
RCVDAT
AWAS
6
RTS 7 ;EXIT

```

811 ;THE FIRST PART OF THE START ROUTINE CONTAINS A SHORT
812 ;ROUTINE TO CHECK FOR MEMORY MANAGEMENT. ALTHOUGH THIS
813 ;DIAGNOSTIC DOES NOT USE MEMORY MANAGEMENT, ITS PRESENCE
814 ;INDICATES THAT OVER 28K OF MEMORY MAY BE PRESENT IN WHICH
815 ;CASE TESTS RT66 AND RT67 MAY FAIL. IF MEM. MAN. IS
816 ;PRESENT THESE TESTS ARE SKIPPED BY THE PROGRAM.
817
818 002360 012706 001200 START: MOV #SPBOT,%6
819 002364 104013 SUSWR ;SEE IF SWITCH-LESS PROCESSOR
820 002366 012737 002410 000004 MOV #15,%ERRVEC ;SET UP FOR ERROR TRAP
821 002374 005737 172300 TST #172300 ;TEST FOR KT11
822 002400 012737 012716 012330 MOV #RT70,%RT65+2 ;KT11 PRESENT, SET UP TO
823 ;SKIP RT66 AND RT67
824 002406 000402 BR +6
825 002410 012706 001200 15: MOV #SPBOT,%6 ;TRAP OCCURRED, NO KT11 PRESENT,
826 ;RESET STACK
827 002414 012737 000006 000004 MOV #ERRVEC+2,%ERRVEC ;RESET ERROR TRAP
828
829
830 ;OUT INTERRUPTS (SET PRIORITY LEVEL 7)
831 002422 004737 003464 RSTAT1: JSR 7,%DMPAR ;GET DM11 PARAMETERS
832 002426 012706 001200 MOV #SPBOT,%6
833 002432 104012 INITIALIZE
834 002434 023737 000042 000046 CMP #42,%46 ;ACT11?
835 002442 001405 BEQ PRGNUM+2
836 002444 104000 TYPE
837 002446 020011 NO
838 002450 004537 004416 JSR 5,RECD ;GET PRGNUM AND PUT IT
839 002454 000000 PRGNUM: 0 ;HERE
840 002456 043737 002014 002454 BIC PRGLIM,PRGNUM ;MASK OFF UNUSED BITS
841 002464 006337 002454 ASL PRGNUM ;SHIFT PROGRAM #
842 002470 012737 004576 000024 MOV #PFAIL,%24
843 002476 012737 000340 000026 MOV #PRTY7,%26
844 002504 013700 002454 MOV PRGNUM,%20 ;GET PROGRAM #
845 002510 000170 002016 JMP #PRGTAB(0) ;GO START PROGRAM
846 002514 012737 004576 000024 RSTAT2: MOV #PFAIL,%24
847 002522 012737 000340 000026 MOV #PRTY7,%26
848 002530 012706 001200 MOV #SPBOT,%6
849 002534 104012 INITIALIZE
850 002536 013700 002454 MOV PRGNUM,%20 ;GET PROGRAM #
851 002542 000170 002024 JMP #RSTART(0) ;GO RESTART PROGRAM
852 002546 022737 000176 001202 SRSET: CMP #SHREG,%SR
853 002554 001410 BEQ 15
854 002556 023737 000042 000046 CMP #42,%46 ;ACT11?
855 002564 001405 BEQ GETRDY
856 002566 104000 TYPE
857 002570 020030 M3
858 002572 000000 HALT
859 002574 000401 BR GETRDY
860 002576 104015 15: CNTLU
861 002600 013737 002000 002006 GETRDY: MOV KSTART,NXTST ;GO GET SHREG SETTINGS
862 002606 012737 000006 000004 GTRDYX: MOV #6,%ERRVEC ;ADDR OF 1ST ROUTINE TO NXTST
863 002614 104012 INITIALIZE ;RESET ERROR TRAP VECTOR
864 002616 004737 003046 GTRDYA: JSR %7,FORWD ;ROLL FORWARD TO "NEXT" ROUTINE.
865 002622 032777 001000 176352 BIT #BIT9,%SR ;CHECK SELECT ROUTINE SWITCH
866 002630 001003 BNE GTRDYC ;BRANCH IF SELECT ROUTINE SWITCH IS SET.

```


867	002632	000177	177144		JMP	@CURTST	;GO RUN CURRENT ROUTINE.	
868	002636	000457			BR	SCOPE	;NO GO. MANUAL RTN BYPASSED.	
869	002640	017700	176336	GTRDYC:	MOV	@SWR,%0	;(SR) TO R0	
870	002644	042700	177600		BIC	#177600,%0	;MASK UNDESIRED BITS	
871	002650	123700	002004		CMPB	RTRNO,%0	;COMPARE RTRNO TO (R0)	
872	002654	001002			BNE	GTRDYD	;BRANCH IF ROUTINE NOT FOUND YET.	
873	002656	000177	177120		JMP	@CURTST	;GO RUN ROUTINE.	
874	002662	022737	177777	002006	GTRDYD:	CMP	#-1,NXTST	;NO. CHECK FOR LAST ROUTINE.
875	002670	001352			BNE	GTRDYA	;BRANCH IF NOT LAST ROUTINE.	
876	002672	004737	002120		JSR	%7, INCRTN	;YES. INCORRECT ROUTINE SELECTED.	
877	002676	000740			BR	GETRDY	;START OVER.	
878								
879								
880	002700	000240			; SCOPE SERVICE ROUTINE			
881	002702	104014		ESCOPE:	NOP			
882	002704	005077	176634		KBDIN			
883	002710	005077	176626		CLR	@BAR	;CHECK FOR G	
884	002714	005077	176626		CLR	@CSR	;CLEAR ALL DM11 REGISTERS	
885	002720	104012			CLR	@KCSR	;AND SET BASE REGISTER	
886	002722	013716	002012		INITIALIZE		;AT THE STARTING ADDRESS	
887	002726	032777	040000	176246	MOV	SCOPTR,(SP)		
888	002734	001402			BIT	@BIT14,@SWR	;CHECK FOR SCOPE OPTION.	
889	002736	000176	000000		BEQ	SCOPE	;BRANCH IF SCOPE SW NOT SET.	
890	002742	032777	004000	176232	JMP	@(SP)	;RETURN TO ROUTINE	
891	002750	001012		SCOPEA:	BIT	@BIT11,@SWR	;TEST INHIBIT ITERATION SWITCH	
892	002752	023737	000042	000046	BNE	SCOPE	;BRANCH IF INHIBIT ITERATION SW SET.	
893	002760	001003			CMP	@#42,@#46	;ACT11?	
894	002762	005737	002116		BNE	IS	;BR IF NO	
895	002766	001403			TST	@PASS	;1ST PASS?	
896	002770	005337	002010		BEQ	SCOPE	;BR IF YES	
897	002774	001360		IS:	DEC	ICTR	;DECREMENT ITERATION COUNT.	
898	002776	032777	001000	176176	BNE	SCOPEA	;BRANCH IF COUNT NOT 0.	
899	003004	001275		SCOPE:	BIT	@BIT9,@SWR	;CHECK SELECT ROUTINE SWITCH	
900	003006	022737	177777	002006	BNE	GETRDY	;BRANCH IF SELECT RTN SW SET	
901	003014	001274			CMP	#-1,NXTST	;LAST TEST?	
902	003016	005237	002116		BNE	GTRDYX	;BRANCH IF NOT LAST TEST.	
903	003022	104000			INC	@PASS		
904	003024	020022			TYPE		;TYPE	
905	003026	013700	000042		M2		; 'PRGEND'	
906	003032	001662			MOV	@#42,%0	;CHECK XXDP/ACT11 MONITOR HOOK	
907	003034	000005			BEQ	GETRDY		
908	003036	004710		SENDAD:	RESET			
909	003040	000240			JSR	7,(0)	;RETURN TO XXDP/ACT11 MONITOR	
910	003042	000240			NOP			
911	003044	000240			NOP			
912	003046	013705	002006	FORWD:	NOP			
913	003052	012537	002004		MOV	NXTST,%5	;ADDR OF NEXT ROUTINE TO R5.	
914	003056	012537	002006		MOV	(%5)+,RTNO	;GET NEXT ROUTINE NUMBER.	
915	003062	012537	002010		MOV	(%5)+,NXTST	;GET ADDR OF NEXT "NEXT" ROUTINE.	
916	003066	012537	002012		MOV	(%5)+,ICTR	;GET ITERATION COUNT.	
917	003072	010537	002002	FORWDA:	MOV	(%5)+,SCOPTR	;GET SCOPE LOOP ENTRY POINTER.	
918	003076	000207			MOV	%5,CURTST	;ADDR OF NOW CURRENT TEST TO CURTST.	
919					RTS	%7	;EXIT FORWD SUBROUTINE.	
920								
921								
922	003100	011646			; 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	; SAVE REGS 0 TO 4 SUBROUTINE.		
933	003136	012637	003170	SAVRG:	MOV (6)+, SVRPC	; SAVE PC AND PSW.
934	003142	010446			MOV (6)+, SVRPSW	
935	003144	010346			MOV X4, -(6)	; SAVE REGS 0 - 4
936	003146	010246			MOV X3, -(6)	; IN STACK.
937	003150	010146			MOV X2, -(6)	
938	003152	010046			MOV X1, -(6)	
939	003154	013746	003170		MOV X0, -(6)	
940	003160	013746	003166		MOV SVRPSW, -(6)	; RESTORE PC AND PSW.
941	003164	000002			MOV SVRPC, -(6)	
942	003166	000000			RTI	; EXIT.
943	003170	000000		SVRPC:	OPEN	
944				SVRPSW:	OPEN	
945						
946	003172	012637	003226	; RESTORE REGS 0 TO 4 SUBROUTINE.		
947	003176	012637	003230	RSTRG:	MOV (6)+, RSTPC	; SAVE PC AND PSW.
948	003202	012600			MOV (6)+, RSTPSW	
949	003204	012601			MOV (6)+, X0	; RESTORE REGS 0 - 4
950	003206	012602			MOV (6)+, X1	; FROM STACK.
951	003210	012603			MOV (6)+, X2	
952	003212	012604			MOV (6)+, X3	
953					MOV (6)+, X4	
954	003214	013746	003230		MOV RSTPSW, -(6)	; RESTORE PC AND PSW.
955	003220	013746	003226		MOV RSTPC, -(6)	
956	003224	000002			RTI	; EXIT
957	003226	000000		RSTPC:	OPEN	
958	003230	000000		RSTPSW:	OPEN	
959						
960				; ROUTINE TO ISSUE RESET.		
961	003232	012700	052525	SRSETT:	MOV #52525, X0	; DATA TO R0.
962	003236	005100			COM X0	; COMPLEMENT (R0).
963	003240	010037	003234		MOV X0, SRSETT+2	; (R0) TO SRSETT+2.
964	003244	000005			RESET	; ISSUE RESET. (R0) IS
965	003246	000002			RTI	; DISPLAYED. EXIT.
966						
967				; RANDOM NUMBER GENERATOR. ROUTINE EXITS WITH NUMBER IN REGISTER 0.		
968	003250	013700	003316	RNGEN:	MOV RP1, X0	
969	003254	006100			R0: X0	
970	003256	006100			ROL X0	
971	003260	063700	003320		ADD RP2, X0	
972	003264	010037	003316		MOV X0, RP1	
973	003270	006100			ROL X0	
974	003272	006100			ROL X0	
975	003274	063700	003320		ADD RP2, X0	
976	003300	006100			ROL X0	
977	003302	006100			ROL X0	
978	003304	010037	003320		MOV X0, RP2	


```

979 003310 013700 003316      MOV      RP1,%0
980 003314 000207              RTS      %7
981 003316 001233              ;EXIT. NUMBER IN RO
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      TYP:    MOV      %X0,%0      ;ADDRESS OF MESSAGE TO RO.
988 003336 122737 000100 003432      TYP:    MOVB    (0)+,TYPDAT      ;GET CHARACTER
989 003344 001001              CMPB    #100,TYPDAT      ;CHECK FOR "a" CHARACTER
990 003346 000002              BNE     TYPC      ;BRANCH IF NOT "a".
991 003350 122737 000045 003432      TYP:    RTI      ;TERMINATOR CHAR. DONE. EXIT.
992 003356 001412              CMPB    #45,TYPDAT      ;CHECK FOR "x".
993 003360 004737 003366              BEQ     TYPF      ;BRANCH IF "x".
994 003364 000762              JSR     %7,TYPD      ;TYPE CHAR IN TYPDAT
995 003366 113777 003432 176500      TYP:    BR      TYPD      ;TYPE CHAR IN TYPDAT
996 003374 105777 176472      TYP:    MOVB    TYPDAT,%TPDBR      ;OUTPUT CHARACTER TO PRINTER
997 003400 100375              TSTB    %TPCSR      ;WAIT FOR DONE FLAG.
998 003402 000207              BPL     .-4
999 003404 112737 000015 003432      TYP:    RTS      %7      ;EXIT
1000 003412 004737 003366              MOVB    #15,TYPDAT      ;MOVE CARRIAGE RETURN CODE TO TYPDAT
1001 003416 112737 000012 003432      TYP:    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      TYPD
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      ;INITIALIZE THE BASE REGISTER
1010 003442 012737 000340 177776      INIT:  MOV      %PRTY7,PSW      ;SET PRIORITY LEVEL 7
1011 003450 011637 001200              MOV      (SP),SPBOT      ;GET RETURN ADDRESS
1012 003454 012706 001200              MOV      %SPBOT,SP      ;SET BOTTOM OF THE STACK
1013 003460 000176 000000              JMP     %SP      ;RETURN
1014
1015
1016      ;SUBROUTINE TO GET DM11 PARAMETERS
1017      ;VECTOR ADDRESS
1018 003464 023737 000042 000046      DMVAR:  CMP      %X42,%X46      ;ACT11?
1019 003472 001060              BNE     %65      ;BR IF NO
1020                      ;SIZE FOR INTERRUPT VECTOR IN AUTO MODE
1021 003474 012700 000302              MOV      %302,RO      ;SET UP FLOATING VECT AREA
1022 003500 010060 177776      45:    MOV      RO,-2(RO)
1023 003504 012720 000003              MOV      %3,(RO)+
1024 003510 005720              TST     (RO)+
1025 003512 022700 000776              CMP      %776,RO
1026 003516 100370              BPL     %45
1027 003520 012737 003610 000014      35:    MOV      %55,%A14      ;SET BPT VECT
1028 003526 012737 000340 000016      35:    MOV      %340,%A16      ;& PSW
1029 003534 012737 177777 001440      35:    MOV      #1,%CT      ;SET TO XMIT 1 CHAR
1030 003542 012737 017102 001400      35:    MOV      %OUTBUF,%CAT
1031 003560 012777 000105 175764      35:    MOV      %BIT6+BIT2+BIT0,%CSR      ;SET IE
1032 003566 005037 177776              CLR     %PSW      ;LVL 0
1033 003562 012777 000001 175754      35:    MOV      %LBITO,%BAR      ;XMIT
1034 003570 012737 177777 002076      35:    MOV      #1,COUNT      ;WAIT

```

1035	003576	005337	002076	25:	DEC	COUNT	
1036	003602	001375			BNE	25	
1037	003604	104001			ERROR		
1038	003606	000752			BR	35	; NO INT OCCURRED
1039	003610	162716	000004	55:	SUB	#4, (SP)	; REPEAT IT
1040	003614	011637	003650		MOV	(SP), @#VECTOR	; CALC INT VECT
1041	003620	012737	000016		MOV	#16, @#14	; STORE IT
1042	003626	004737	004372		JSR	7, OVLAY	; RESTORE BPT VECT
1043	003632	000415			BR	VECOK	; +2, HALT IN VECT AREA
1044	003634	004737	004372	65:	JSR	7, OVLAY	
1045	003640	104000			TYPE		; PUT HALT, +2 IN VECTOR AREA
1046	003642	017626			WHERE		; ASK USER FOR RECEIVER INT. VECTOR
1047	003644	004537	004416		JSR	5, RECD	; OF UNIT UNDER TEST
1048	003650	000000		VECTOR:	O		; GET VECTOR AND PUT IT
1049	003652	005737	003650		TST	VECTOR	; HERE
1050	003656	001003			BNE	VECOK	
1051	003660	012737	000300	003650	MOV	#300, VECTOR	; SET VECTOR = TO 0300
1052	003666	023727	003650	000300	VECOK:	CMP	; IS VECTOR HIGHER OR
1053	003674	103003			BHIS	VECTOR, #300	; EQUAL TO 0300
1054	003676	104000			VECOKA:	TYPE	; TYPE '?'
1055	003700	020017			M1		
1056	003702	000670			BR	DMPAR	; ASK FOR ANOTHER VECTOR
1057	003704	023727	003650	000770	VECOKB:	CMP	; IS VECTOR = TO OR
1058	003712	101371			BHI	VECTOR, #770	; 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		
1067	003754	001405			DMPARB:	CMP	; ACT11?
1068	003756	104000			BEQ	@#42, @#46	; BR IF YES
1069	003760	017725			TYPE	UNIT+2	
1070	003762	004537	004416		WHICH		
1071	003766	000000			JSR	5, RECD	; GET UNIT AND PUT IT
1072	003770	023727	003766	000017	UNIT:	O	; HERE
1073	003776	101403			CMP	UNIT, #17	; UNIT SELECTED MUST BE
1074	004000	104000			BLOS	UNTOKA	; BETWEEN 0 & 17
1075	004002	020017			TYPE		
1076	004004	000760			M1		
1077	004006	006337	003766		BR	DMPARB	
1078	004012	006337	003766		UNTOKA:	ASL	UNIT
1079	004016	006337	003766		ASL	UNIT	
1080	004022	012702	000004		ASL	UNIT	
1081	004026	012701	001542		MOV	#4, X2	
1082	004032	042711	000370		UNTOKB:	MOV	; FORM ADDRESSES OF
1083	004036	063721	003766		BIC	#370, (1)	; REGISTERS OF UNIT SELECTED
1084	004042	005302			ADD	UNIT, (1)+	
1085	004044	001372			DEC	X2	
1086					BNE	UNTOKB	
1087	004046	012702	000004		MOV	#4, X2	
1088	004052	012703	001542		MOV	#CSR, X3	
1089	004056	012701	002106		UNTOKC:	MOV	; FORM ODD BYTE ADDRESSES
1090	004062	012311			MOV	#, CSR, X1	
						(3)+, (1)	


```
1091 004064 005221
1092 004066 005302
1093 004070 001374
1094
1095 004072 023737 000042 000046 ; CHARACTER LENGTH
1096 004100 001405 DMPARC: CMP #42, #46 ; ACT11?
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 BHS LENOKB ; BE BETWEEN 5-8
1107 004142 020017 TYPE ; CARRIAGE RETURN SELECTS 8
1108 004144 000752 M1
1109 004146 023727 004112 000010 LENOKC: CMP DMPARC
1110 004154 101371 BHI LENGTH, #8.
1111 004156 162737 000005 004112 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), #CARMASK ; SET CHARACTER LENGTH MASK
1116 004204 000207 NOP
1117 RTS 7 ; EXIT PARAMETERS ROUTINE
1118
1119 ; THE BELOW TABLE REPRESENTS THE CHARACTER LENGTH MASK FOR 5, 6, 7, AND 8
1120 004206 177740 ; BITS PER CHARACTER RESPECTIVELY.
1121 004210 177700 LENOKD: 177740
1122 004212 177600 177700
1123 004214 177400 177600
1124 177400
1125
1126 004216 005037 004366 ; CALCULATE MACHINE TIME TO TRANSMIT ONE CHARACTER
1127 004222 012737 177777 001440 TIMER: CLR TIME1
1128 004230 012737 017102 001400 MOV #1, WCT ; SET UP TO TRANSMIT
1129 004236 012777 004336 175306 MOV #OUTBUF, CAT ; 1 CHARACTER ON LINE 1
1130 004244 012777 000340 175302 MOV #TIMEC, #CLKINT ; LOAD RECEIVER INTERRUPT
1131 004252 012777 000001 175264 MOV #PRTY7, #CLKLVL ; AND PRIORITY
1132 004260 012777 000105 175254 MOV #LBITO, #BAR ; START TRANSMITTING
1133 004266 005037 177776 CLR #BIT6+BIT2+BIT0, #CSR ; SET IE BIT
1134 004272 012737 000044 002076 TIMEA: MOV #44, COUNT ; SET PROCESSOR PRIORITY LEVEL = 0
1135 004300 062737 000001 004366 ADD #1, TIME1 ; INCREMENT MACH. TIME TO TRANSMIT
1136 004306 001007 BNE TIMEB
1137 004310 005077 175226 CLR #CSR
1138 004314 012737 000340 177776 MOV #PRTY7, PSW ; SET PROCESSOR PRIORITY LEVEL = 7
1139 004322 104001 ERROR ; TRANSMITTER FAILED TO INTERRUPT
1140 004324 000734 BR TIMER
1141 004326 005337 002076 TIMEB: DEC COUNT
1142 004332 001375 BNE -4
1143 004334 000756 BR TIMEA
1144 004336 005077 175200 TIMEC: CLR #CSR
1145 004342 013737 004366 004370 MOV TIME1, TIME14
1146 004350 006037 004370 ROR TIME14
```

```

CLC
ROR
POPSP2
RTS
TIME14
7
;RESTORE STACK POINTER
;EXIT TIME CALCULATION ROUTINE

TIME1: OPEN
TIME14: OPEN
;CONTAINS MACHINE TIME TO XMIT 1 CHAR
;CONTAIN TIME TO XMIT 1/4 CHAR

;SUBROUTINE TO PUT MALT.. +2 IN VECTOR AREA (0300-1000)
OVLAY: MOV #302, R2
15: MOV R2, -2(R2)
CLR (R2)+
TST (R2)+
CMP #776, R2
BPL 15
RTS 7

;SUBROUTINE TO RECEIVE DATA
;THIS SUBROUTINE RECEIVES DATA FROM THE KEYBOARD (UP TO SIX OCTAL
;DIGITS AND PLACES THEM INTO THE ADDRESS FOLLOWING THE SUBROUTINE
;CALL (JSR 5, RECD). NO REGISTER CONTENTS ARE DISTURBED.

;SUBROUTINE TO INPUT DATA FROM TTY

RECD: MOV RO, -(SP)
15: CLR (5)
MOV #7, CNT
25: TSTB @TKCSR
BPL 25
MOVB @TKDOR, RO
BICB #200, RO
NOVB RO, @TPDOR
CMPB #35, RO
BEQ 55
CMPB #15, RO
BEQ 65
BICB #60, RO
BITB #110, RO
BNE 75
ASL (5)
ASL (5)
ASL (5)
BISB RO, (5)
DEC CNT
BEQ 75
BR 25
65: TSTB @TPCSR
BPL 65
NOVB #12, @TPDOR
85: TSTB @TPCSR
BPL 85
CLR @TPDOR
95: TSTB @TPCSR
BPL 95

;CLEAR OLD DATA
;SET CHAR COUNT
;WAIT FOR CHAR

;STRIP OFF PARITY
;ECHO CHARACTER
;IS IT A U
;BRANCH IF YES
;IS IT A <CR>
;BRANCH IF YESS

;CHECK FOR 0-7 (8)
;BRANCH IF NOT

;SHIFT DATA
;INSET NEW CHAR

;ONLY 6 CHAR'S PLEASE
;NEXT CHARACTER

;WAIT FOR READY
;TYPE <LF>

;WAIT FOR READY
;LOAD CHAR

```


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	; POWER FAIL ROUTINE				
1215	004604	000000			PFAIL:	MOV	#PHRUP, 24		
1216						HALT			
1217									
1218	004606	000005			; POWER UP SUBROUTINE				
1219	004610	012706	001200		PHRUP:	RESET			; GIVE TELEPRINTER TIME TO START
1220	004614	104001				MOV	#SPBOT, %6		
1221	004616	000137	002514			ERROR			; TYPE POWER FAIL ERROR
1222						JMP	#RSTAT2		; GO RESTART PROGRAM
1223									
1224									
1225									
1226									
1227									
1228									
1229									
1230									
1231									
1232									
1233									
1234	004622	012537	016730						
1235	004626	004737	007044						
1236	004632	005037	001570						
1237	004636	004537	006246						
1238	004642	177777							
1239	004644	012703	000010						
1240	004650	005002							
1241	004652	005302							
1242	004654	001376							
1243	004656	005303							
1244	004660	001374							
1245	004662	017737	174656	001566					
1246	004670	001401							
1247	004672	104011							
1248	004674	005777	174642						
1249	004700	104001							
1250	004702	104001							
1251	004704	013701	016730						
1252	004710	016137	001440	001566					
1253	004716	001401							
1254	004720	104011							
1255	004722	012737	017102	001570					
1256	004730	016137	001400	001566					
1257	004736	023737	001566	001570					
1258	004744	001401							

; LINE TEST SUBROUTINE: THIS LINE TEST PROVIDES SEVERAL TESTS ON A DM11 LINE.
 ; THE SUBROUTINE IS CALLED BY JSR 5, LNTST. THIS INSTRUCTION PROVIDES THE
 ; LINE BIT AND LINE NUMBER. THE FOLLOWING LINE TESTS ARE PERFORMED:
 ; WAITS UNTIL CHARACTER SHOULD HAVE BEEN TRANSMITTED, THEN TESTS
 ; THAT BAR BIT CLEARED ; DO NEXT TEST IF ERROR
 ; READY SET ; DO NEXT TEST IF ERROR
 ; WORD COUNT WENT TO 0 ; DO NEXT TEST IF ERROR
 ; CURRENT ADDRESS DID NOT INCREMENT ; DO NEXT TEST IF ERROR
 ; INTERRUPTS TO CORRECT VECTOR ; DO NEXT TEST IF ERROR (NO INTERRUPT)
 ; READY BIT CAN BE CLEARED ; END OF TEST
 XMTTST: MOV (5)+, #LINE ; GET LINE NUMBER
 JSR 7, GTLINE ; GO FOR LINE BIT (FOR BAR)
 CLR XMTDAT
 15: JSR 5, #XMITD ; GO TO TRANSMIT SUBROUTINE
 -1 ; TRANSMIT ONE CHARACTER
 MOV #10, X3 ; WAIT IN
 CLR X2 ; THIS
 25: DEC X2 ; LOOP
 BNE -2 ; UNTIL THE
 DEC X3 ; TRANSMITTER
 BNE 25 ; IS FINISHED
 MOV #BAR, RCVDAT ; BAR SHOULD NOW BE CLEAR
 BEQ 35 ; BRANCH IF IT IS
 ERROR1 ; ERROR! BAR BIT FAILED TO CLEAR
 35: TST #CSR ; TEST READY BIT SHOULD BE SET
 BMI 45 ; BRANCH IF SET
 ERROR ; ERROR! READY NOT SET
 45: MOV #LINE, X1 ; GET LINE NUMBER
 MOV WCT(1), RCVDAT ; WORD COUNT SHOULD BE 0
 BEQ 55
 ERROR1 ; ERROR! WORD COUNT NOT EQUAL TO 0
 55: MOV #OUTBUF, XMTDAT ;
 MOV CAT(1), RCVDAT ; CURRENT ADDRESS SHOULD NOT HAVE INCREMENTED
 CMP RCVDAT, XMTDAT ;
 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								; AND HALTED WITH ADDRESS +2 DISPLAYED.
1267	005002	022626				CMP	(6)+, (6)+	; RESET STACK PTR
1268	005004	012737	000340	177776	75:	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

;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.

005034	012737	177777	017102	RCVTST:	MOV	8-1,OUTBUF	;LOAD ALL 1'S INTO OUTPUT BUFFER
005042	005037	001600			CLR	TUMTAB	;CLEAR THE FIRST
005046	005037	001602			CLR	TUMTAB+2	;TWO TUMBLE TABLE ADDRESSES
005052	012737	000340	177776		MOV	8PTY7,8PSW	;LOCK OUT INTERRUPTS
005060	012537	016730			MOV	(5)+,LINE	;GET LINE NUMBER
005064	004537	006246			JSR	5,8XMITD	;TRANSMIT 1 CHARACTER (0'S)
005070	177777				-1		;ON LINE SPECIFIED BY JSR
005072	052777	000001	174442		BIS	8BIT0,8CSR	;SET GO BIT
005100	005777	174436			TST	8CSR	;WAIT FOR TRANSMITTER
005104	100375				BPL	-4	;TO TRANSMIT 1 CHAR.
005106	042777	100000	174426		BIC	8BIT15,8CSR	;CLEAR TRANSMITTER READY FLAG
005114	005046				CLR	-(SP)	;SET WATCH DOG TIMER
005116	105777	174420		15:	TSTB	8CSR	;TEST CHAR. DONE FLAG
005122	100404				BMI	25	;BRANCH IF SET
005124	005216				INC	(SP)	;WAIT FOR THE FLAG
005126	001373				BNE	15	
005130	104001				ERROR		;ERROR! CHAR. DONE FLAG FAILED TO SET
005132	000550				BR	85	;GO TO EXIT
005134	005726			25:	TST	(SP)+	;RESTORE STACK PTR
005136	012777	005172	174406		MOV	835,8CLKINT	;LOAD RECEIVER INTERRUPT VEC. ADRS.
005144	052777	000100	174370		BIS	8BIT6,8CSR	;SET RECEIVER IE BIT
005152	005037	177776			CLR	8PSW	;ENABLE INTERRUPTS
005156	000240				NOP		
005160	012737	000340	177776		MOV	8PTY7,PSW	;LOCK OUT INTERRUPTS
005166	104001				ERROR		;RECEIVER FAILED TO INTERRUPT
005170	000531				BR	85	;GO TO EXIT
005172	012737	000340	177776	35:	MOV	8PTY7,8PSW	;LOCK OUT INTERRUPTS
005200	022626				CMP	(6)+,(6)+	
005202	042777	000300	174332		BIC	8BIT7+BIT6,8CSR	;CLEAR CHAR. DONE FLAG
005210	105777	174326			TSTB	8CSR	;TEST THAT CHAR DONE FLAG CLEARED
005214	100002				BPL	45	;BRANCH IF CHAR. DONE FLAG CLEARED
005216	104001				ERROR		;ERROR! CHAR. DONE FAILED TO CLEAR
005220	000515				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
1342 005300 000465 BR 85 ;ERROR! INCORRECT TUMBLE TABLE
1343 005302 005037 001600 55: CLR TUMTAB ;ENTRY; GO TO EXIT
1344 005306 013737 001602 001566 MOV TUMTAB+2,RCV DAT ;ET NEXT ENTRY
1345 005314 001404 BEQ 65 ;BRANCH IF ALL 0'S
1346 005316 005037 001570 CLR XMT DAT
1347 005322 104011 ERROR1
1348 005324 000453 BR 85 ;ERROR! FALSE ENTRY IN NEXT
1349 005326 004537 006246 65: JSR 5, @XMTD ;TUMBLE TABLE ADDRESS
1350 005332 177777 -1 ;TRANSMIT 1 CHARACTER (ALL 1'S)
1351 005334 005777 174202 TST @CSR ;ON LINE SPECIFIED BY JSR
1352 005340 100375 BPL -4 ;WAIT FOR TRANSMITTER
1353 005342 105777 174174 TSTB @CSR ;READY FLAG
1354 005346 100375 BPL -4 ;TEST FOR THE DONE FLAG
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
1360 005374 000427 BR 85 ;ERROR! TUMBLE TABLE POINTER DID
1361 005376 013737 001602 001566 75: MOV TUMTAB+2,RCV DAT ;NOT INCREMENT; GO TO EXIT
1362 005404 042737 020000 001566 BIC #BIT13,RCV DAT ;GET TUMBLE TABLE ENTRY
1363 005412 012737 000377 001570 MOV #377,XMT DAT ;CLEAR PARITY INDICATOR
1364 005420 043737 001572 001570 BIC CARMSK,XMT DAT ;LOAD XMT DAT WITH TRANSMITTED DATA
1365 005426 153737 016730 001571 BISB LINE,XMT DAT+1 ;CLEAR NON-TRANSMITTED BITS
1366 005434 052737 100000 001570 BIS #BIT15,XMT DAT ;LOAD LINE # INTO XMT DAT
1367 005442 023737 001566 001570 CMP RCV DAT,XMT DAT ;SET VALID DATA ENTRY BIT INTO XMT DAT
1368 ;COMPARE TUMBLE TABLE ENTRY (RCV DAT) &
1369 ;CORRECT RESULT (XMT DAT)
1369 005450 001401 BEQ 85
1370 005452 104011 ERROR1
1371 005454 104006 85: SCOPE ;ERROR! 2ND TUMBLE TABLE ENTRY
1372 ;WAS INCORRECT; SCOPE
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
1380 005456 012777 000001 174056 BRKTST: MOV #1,@CSR ;SET THE GO BIT
1381 005464 011577 174056 MOV (5),@BKCSR ;SET THE BREAK BIT
1382 005470 105777 174046 TSTB @CSR ;WAIT FOR THE RECEIVER TO
1383 005474 100375 BPL -4 ;RECEIVE BREAK
1384 005476 042777 000200 174036 BIC #BIT7,@CSR ;CLEAR FLAG
1385 005504 105777 174032 TSTB @CSR ;WAIT FOR THE RECEIVER TO
1386 005510 100375 BPL -4 ;TO RECEIVE BREAK
```


1387	005512	042777	000200	174022		BIC	#BIT7, @CSR	; CLEAR FLAG
1388	005520	005077	174022			CLR	@BKCSR	; CLEAR BREAK BIT
1389	005524	005737	001600		15:	TST	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	005724				
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
DLYXMT: MOV      (5)+, @COUNT      ; GET CHARACTER DELAY COUNT
        CLR      XMTDAT
        JSR      5, @MOVE          ; LOAD OUTPUT BUFFER WITH DATA
        MSG1     OUTBUF           ; TO BE TRANSMITTED
        MOV      @LBIT0, @LINBIT
        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      LINBIT          ; 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							
1444	005774	005726							
1445	005776	104006				85:	ERROR1		; INCORRECT CSR CONTENTS
1446							POPSP		; RESET THE STACK
1447							SCOPE		; SCOPE
1448	006000	011637	006050						
1449	006004	062716	000002						
1450	006010	017737	000034	006050					
1451	006016	001413							
1452	006020	012737	000050	006052					
1453	006026	162737	000001	006050		15:			
1454	006034	001404							
1455	006036	005337	006052						
1456	006042	001375							
1457	006044	000765							
1458	006046	000002				25:	RTI		; EXIT
1459	006050	000000				35:	OPEN		; CONTAINS DELAY COUNT
1460	006052	000000				45:	OPEN		; CONTAINS DELAY ROUTINE CONSTANT
1461									
1462									
1463	006054	012737	177777	006076					
1464	006062	004537	006224						
1465	006066	006076							
1466	006070	006077							
1467	006072	000005							
1468	006074	000207							
1469	006076	000000							
1470	006100	000000				RIND:	OPEN		
1471	006102	000000				PTO:	OPEN		
1472						PT1:	OPEN		
1473									
1474	006104	013737	006100	006102					
1475	006112	005137	006102						
1476	006116	005137	006076						
1477	006122	001002							
1478	006124	005237	006102						
1479	006130	042737	177400	006102					
1480	006136	013737	006102	006100					
1481	006144	013701	006102						
1482	006150	000240							
1483	006152	000207							
1484									
1485									
1486	006154	104007							
1487	006156	013504							
1488	006160	012501							
1489	006162	012502							
1490	006164	060201							
1491	006166	010403							
1492	006170	042703	177770						
1493	006174	062703	000060						
1494	006200	110341							
1495	006202	042704	000007						
1496	006206	006004							
1497	006210	006004							
1498	006212	006004							

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,%GTLINE	; FORM LINE BIT (FOR BAR)
1521	006266	012760	017102	001400			MOV	%CAT,%BASREG	; 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			DAT1AA:	MOV	%100,%COUNT	; GET CHARACTER COUNT
1533	006330	004737	006104				MOV	%OUTBUF,%2	; GET ADDRESS OF OUTPUT BUFFER
1534	006334	110122			15:	JSR	7,%GTBIN		; 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	%TUNTAB,%1	; LOAD TUMBLE TABLE POINTER
1539	006354	004537	006224				CLR	TUNTAB	
1540	006360	001600					JSR	5,BMOVE	; CLEAR
1541	006362	001601					TUNTAB		; TUMBLE
1542	006364	000177					TUNTAB+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 TST @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 ;ROUTINE TO STORE RECEIVED PARITY BIT IN PARITY BIT BUFFER
1566 006476 012705 000001 MOV #1, X5 ;GET ROTATE COUNT
1567 006502 000261 SEC ;SET THE CARRY BIT
1568 006504 032711 020000 BIT #BIT13, (1) ;TEST RECEIVED PARITY BIT
1569 006510 001001 BNE 65 ;BRANCH IF RECEIVED PARITY WAS ODD
1570 006512 000241 CLC ;CLEAR CARRY BIT
1571 006514 004537 006764 65: JSR 5, RORPARBUF ;ROTATE RECEIVED PARITY INTO PARITY BUFFER
1572
1573 ;ROUTINE TO TEST THAT ENTRY IS FOR THE CORRECT LINE.
1574 006520 011137 001562 MOV (1), @TTDAT ;GET TABLE ENTRY
1575 006524 042737 160777 001562 BIC #160777, TTDAT ;CLEAR ALL BUT LINE NUMBER
1576 006532 123737 016730 001563 CMPB LINE, TTDAT+1 ;COMPARE LINE NUMBERS
1577 006540 001410 BEQ 75
1578 006542 013737 016730 001570 MOV LINE, XMTDAT ;GET CORRECT LINE # (X2)
1579 006550 013737 001562 001566 MOV TTDAT, RCVDAT ;GET LINE # (X2) THAT FALSE DATA CAME IN ON
1580 006556 104011 ERROR1 ;ERROR! DATA CAME IN ON A LINE THAT
1581 ;PROGRAM WAS NOT TRANSMITTING ON
1582 006560 000477 BR 155 ;EXIT TEST
1583 006562 020127 001776 75: CMP X1, @TUMTAB+176 ;IS POINTER AT THE END
1584 006566 001002 BNE 85 ;OF THE TABLE
1585 006570 012701 001576 MOV @TUMTAB-2, X1
1586 006574 005721 85: TST (1)+ ;INCREMENT POINTER
1587 006576 010046 MOV X0, -(6) ;SAVE REGISTER ZERO
1588 006600 013700 016730 MOV LINE, X0 ;FETCH LINE NUMBER
1589 006604 005760 001440 TST WCT(0) ;HAS THE LAST CHARACTER BEEN TRANSMITTED
1590 006610 001402 BEQ +6 ;LAST CHARACTER HAS BEEN TRANSMITTED
1591 006612 012600 MOV (6)+, X0 ;RESTORE REGISTER ZERO
1592 006614 000674 BR 25 ;GO WAIT FOR NEXT CHARACTER
1593 006616 012600 MOV (6)+, X0 ;RESTORE REGISTER ZERO
1594 006620 012701 017102 95: MOV @OUTBUF, X1
1595 006624 012702 017246 MOV @INBUF, X2
1596 006630 012705 000014 MOV #12, X5 ;ROTATE PARITY BUFFER
1597 006634 004537 006764 105: JSR 5, RORPARBUF ;12 PLACES RIGHT
1598 006640 005037 001566 CLR RCVDAT
1599 006644 005037 001570 CLR XMTDAT
1600 006650 020127 017245 CMP X1, @OUTBUF+99 ;HAVE ALL CHARS. BEEN COMPARED
1601 006654 001441 BEQ 155
1602 006656 112137 001570 MOVB (1)+, XMTDAT ;GET TRANSMITTED CHARACTER
1603 006662 043737 001572 001570 BIC CARMASK, XMTDAT ;CLEAR NON-TRANSMITTED BITS
1604 006670 111237 001566 MOVB (2), RCVDAT ;GET RECEIVED CHARACTER
1605 006674 104002 DATCHK ;COMPARE TRANS. & RCVD. CHARS.
1606
1607 ;ROUTINE TO COMPUTE AND CHECK PARITY ON RECEIVED DATA
1608 006676 012703 000010 115: MOV #8, X3 ;GET BIT COUNTER
1609 006702 005000 CLR X0 ;CLEAR COMPUTED PARITY INDICATOR
1610 006704 106037 001566 125: RORB RCVDAT ;LOOK AT RECEIVED BIT
```



```
1611 006710 103001
1612 006712 005100
1613 006714 005303
1614 006716 001372
1615 006720 000240
1616
1617 006722 112237 001566
1618 006726 012705 000001
1619 006732 004537 006764
1620 006736 103004
1621 006740 005700
1622 006742 001336
1623 006744 104001
1624 006746 000734
1625 006750 005700
1626 006752 001732
1627 006754 104001
1628 006756 000730
1629 006760 005726
1630 006762 104006
1631
1632
1633 006764 006037 007026
1634 006770 006037 007030
1635 006774 006037 007032
1636 007000 006037 007034
1637 007004 006037 007036
1638 007010 006037 007040
1639 007014 006037 007042
1640 007020 005316
1641 007022 001360
1642 007024 000205
1643
1644 007026 000000
1645 007030 000000
1646 007032 000000
1647 007034 000000
1648 007036 000000
1649 007040 000000
1650 007042 000000
1651
1652
1653 007044 010046
1654 007046 005037 001564
1655 007052 013700 016730
1656 007056 000261
1657 007060 006137 001564
1658 007064 162700 000002
1659 007070 100373
1660 007072 012600
1661 007074 000207
1662
```

```
135: BCC 135 ; BRANCH IF A 0
      COM %0 ; COMPLEMENT RO IF A 1
      DEC %3 ; DECREMENT BIT COUNTER
      BNE 125 ; LOOK AT NEXT BIT IF NOT DONE
      NOP ; IF COMPUTED PARITY WAS ODD RO WILL
      ; CONTAIN ALL 1'S, IF EVEN RO = 0
      MOVB (2)+,RCV DAT ; GET RECEIVED CHARACTER
      MOV #1,%5 ; ROTATE PARITY BUFFER 1 PLACE
      JSR 5,RORPARBUF ; RIGHT LEAVING RECEIVED PARITY BIT IN CARRY
      BCC 145 ; BRANCH IF RECEIVED PARITY WAS EVEN
      TST %0 ; TEST FOR COMPUTED ODD PARITY
      BNE 105 ; BRANCH IF COMPUTED & RECEIVED WAS ODD
      ERROR ; ERROR! COMPUTED =EVEN, RECEIVED = ODD
145: BR 105 ; CONTINUE TEST
      TST %0 ; TEST FOR EVEN COMPUTED PARITY
      BEQ 105 ; BRANCH IF COMPUTED PARITY WAS EVEN
      ERROR ; ERROR! COMPUTED =ODD, RECEIVED = EVEN
155: BR 105 ; CONTINUE TEST
      POPSP ; REPOSITION STACK POINTER
      SCOPE ; SCOPE

; ROUTINE TO ROTATE PARITY BUFFER.
RORPARBUF: ROR PAR0
           ROR PAR1
           ROR PAR2
           ROR PAR3
           ROR PAR4
           ROR PAR5
           ROR PAR6
           DEC (SP) ; DECREMENT ROTATE COUNT
           BNE RORPARBUF
           RTS 5

; PARITY BUFFER
PAR0: OPEN
PAR1: OPEN
PAR2: OPEN
PAR3: OPEN
PAR4: OPEN
PAR5: OPEN
PAR6: OPEN

; SUBROUTINE TO FORM LINE BIT POSITION WITH THE LINE # IN LINE
GTLINB: MOV %0,-(SP) ; SAVE RO ON THE STACK
        CLR @LINBIT ; CLEAR LINE BIT
        MOV @LINE,%0 ; GET LINE
        SEC ; SET CARRY
15: ROL LINBIT ; SHIFT LINE BIT
    SUB #2,%0 ; SUBTRACT 2 FROM LINE NUMBER
    BPL 15 ; BRANCH IF GREATER THAN 0
    MOV (SP)+,%0 ; RESTORE RO
    RTS 7 ; EXIT SUBROUTINE
```

```
1663
1664 007076 104000
1665 007100 020103
1666 007102 012737 007136 002000
1667 007110 005037 002004
1668 007114 000137 002546
1669 007120 012737 007136 002000
1670 007126 005037 002004
1671 007132 000137 002600
1672
1673 007136 000000
1674 007140 007200
1675 007142 000144
1676 007144 007146
1677 000000
1678
1679
1680
1681 007146 012737 007170 000004
1682 007154 005777 172362
1683 007160 012737 000006 000004
1684 007166 104006
1685 007170 162716 000004
1686
1687 007174 104001
1688 007176 000002
1689
1690 007200 000001
1691 007202 007252
1692 007204 000144
1693 007206 007210
1694 000001
1695
1696
1697
1698 007210 012777 000001 172324
1699 007216 022777 000001 172316
1700 007224 001402
1701 007226 104001
1702 007230 000407
1703 007232 042777 000001 172302
1704 007240 005777 172276
1705 007244 001401
1706 007246 104001
1707 007250 104006
1708
1709 007252 000002
1710 007254 007324
1711 007256 000144
1712 007260 007262
1713 000002
1714
1715
1716
1717 007262 012777 000002 172252
1718 007270 022777 000002 172244

PRGO: TYPE
PRGOM
MOV #RTO, KSTART ; GET ADDRESS OF FIRST TEST
CLR RTNNO ; CLEAR ROUTINE #
JMP SRSET
PRGOR: MOV #RTO, KSTART ; GET ADDRESS OF FIRST TEST
CLR RTNNO ; CLEAR ROUTINE NUMBER
JMP GETRDY ; GO AND START PROGRAM
; *****
RTO: 0 ; ROUTINE # 0
RT1 ; ADDR OF NEXT ROUTINE.
100. ; ITERATION COUNT
RTOA ; SCOPE ENTRY POINT.
X=X+1
; *****
; TEST ABILITY TO REFERENCE CSR WITHOUT TRAPPING
RTOA: MOV #15, @ERRVEC ; SET UP ERROR TRAP.
TST @CSR ; REFERENCE CSR
MOV #ERRVEC+2, @ERRVEC ; RESET TIME OUT TRAP
SCOPE
15: SUB #4, (6) ; RESTORE PC TO WHERE THE ILLEGAL
; REFERENCE OCCURED
ERROR ; ERROR! ILLEGAL REFERENCE OCCURED
RT1 ; LOOP ILLEGAL REFERENCE INSTRUCTION
; *****
RT1: 1 ; ROUTINE # 1
RT2 ; ADDR OF NEXT ROUTINE.
100. ; ITERATION COUNT
RT1A ; SCOPE ENTRY POINT.
X=X+1
; *****
; TEST THAT CSR BIT0 CAN BE SET AND CLEARED
RT1A: MOV #BIT0, @CSR ; SET BIT0.
CMP #BIT0, @CSR ; TEST THAT BIT0 IS SET
BEQ 15 ; BRANCH IF SET
ERROR ; CSR BIT0 FAILED TO SET
BR 25 ; OR AN ADDITIONAL BIT ALSO SET
15: BIC #BIT0, @CSR ; CLEAR BIT0
TST @CSR ; TEST THAT BIT0 IS CLEAR
BEQ 25 ; CSR BIT0 FAILED TO CLEAR
ERROR
25: SCOPE
; *****
RT2: 2 ; ROUTINE # 2
RT3 ; ADDR OF NEXT ROUTINE.
100. ; ITERATION COUNT
RT2A ; SCOPE ENTRY POINT.
X=X+1
; *****
; TEST THAT CSR BIT1 CAN BE SET AND CLEARED
RT2A: MOV #BIT1, @CSR ; SET BIT1.
CMP #BIT1, @CSR ; TEST THAT BIT1 IS SET
```



```
1719 007276 001402 BEQ 15 ; BRANCH IF SET
1720 007300 104001 ERROR ; CSR BIT1 FAILED TO SET
1721 007302 000407 BR 25 ; OR AN ADDITIONAL BIT ALSO SET
1722 007304 042777 000002 172230 15: BIC #BIT1, @CSR ; CLEAR BIT1
1723 007312 005777 172224 TST @CSR ; TEST THAT BIT1 IS CLEAR
1724 007316 001401 BEQ 25 ; CSR BIT1 FAILED TO CLEAR
1725 007320 104001 ERROR
1726 007322 104006 SCOPE
1727 ; *****
1728 007324 000003 RT3: 3 ; ROUTINE # 3
1729 007326 007376 RT4 ; ADDR OF NEXT ROUTINE.
1730 007330 000144 100. ; ITERATION COUNT
1731 007332 007334 RT3A ; SCOPE ENTRY POINT.
1732 000003 X=X+1
1733 ; *****
1734 ; TEST THAT CSR BIT2 CAN BE SET AND CLEARED
1735 RT3A: MOV #BIT2, @CSR ; SET BIT2.
1736 007334 012777 000004 172200 CMP #BIT2, @CSR ; TEST THAT BIT2 IS SET
1737 007342 022777 000004 172172 BEQ 15 ; BRANCH IF SET
1738 007350 001402 ERROR ; CSR BIT2 FAILED TO SET
1739 007352 104001 BR 25 ; OR AN ADDITIONAL BIT ALSO SET
1740 007354 000407 172156 15: BIC #BIT2, @CSR ; CLEAR BIT2
1741 007356 042777 000004 172152 TST @CSR ; TEST THAT BIT2 IS CLEAR
1742 007364 005777 BEQ 25 ; CSR BIT2 FAILED TO CLEAR
1743 007370 001401 ERROR
1744 007372 104001 SCOPE
1745 007374 104006 ; *****
1746 RT4: 4 ; ROUTINE # 4
1747 007376 000004 RT5 ; ADDR OF NEXT ROUTINE.
1748 007400 007450 100. ; ITERATION COUNT
1749 007402 000144 RT4A ; SCOPE ENTRY POINT.
1750 007404 007406 X=X+1
1751 000004 ; *****
1752 ; TEST THAT CSR BIT4 CAN BE SET AND CLEARED
1753 RT4A: MOV #BIT4, @CSR ; SET BIT4.
1754 007406 012777 000020 172126 CMP #BIT4, @CSR ; TEST THAT BIT4 IS SET
1755 007414 022777 000020 172120 BEQ 15 ; BRANCH IF SET
1756 007422 001402 ERROR ; CSR BIT4 FAILED TO SET
1757 007424 104001 BR 25 ; OR AN ADDITIONAL BIT ALSO SET
1758 007426 000407 172104 15: BIC #BIT4, @CSR ; CLEAR BIT4
1759 007430 042777 000020 172100 TST @CSR ; TEST THAT BIT4 IS CLEAR
1760 007436 005777 BEQ 25 ; CSR BIT4 FAILED TO CLEAR
1761 007438 104001 ERROR
1762 007442 001401 SCOPE
1763 007444 104001 ; *****
1764 007446 104006 RT5: 5 ; ROUTINE # 5
1765 007450 000005 RT6 ; ADDR OF NEXT ROUTINE.
1766 007452 007522 100. ; ITERATION COUNT
1767 007454 000144 RT5A ; SCOPE ENTRY POINT.
1768 007456 007460 X=X+1
1769 000005 ; *****
1770 ; TEST THAT CSR BITS CAN BE SET AND CLEARED
1771 RT5A: MOV #BIT5, @CSR ; SET BITS.
1772
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      ;ADDR OF NEXT ROUTINE.
1787 007526 000144                      100.      ;ITERATION COUNT
1788 007530 007532                      RT6A     ;SCOPE ENTRY POINT.
1789 000006                      X=X+1
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     ;ADDR OF NEXT ROUTINE.
1806 007600 000144                      100.      ;ITERATION COUNT
1807 007602 007604                      RT7A     ;SCOPE ENTRY POINT.
1808 000007                      X=X+1
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     ;ADDR OF NEXT ROUTINE.
1825 007652 000144                      100.      ;ITERATION COUNT
1826 007654 007656                      RT10A    ;SCOPE ENTRY POINT.
1827 000010                      X=X+1
1828                                     ;*****
1829
1830                                     ;TEST THAT CSR BIT13 CAN BE SET AND CLEARED
```



```

1831 007656 012777 020000 171656 RT10A: MOV #BIT13, &CSR ;SET BIT13.
1832 007664 022777 020000 171650 CMP #BIT13, &CSR ;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, &CSR ;CLEAR BIT13
1837 007706 005777 171630 TST &CSR ;TEST THAT BIT13 IS CLEAR
1838 007712 001401 BEQ 25 ;CSR BIT13 FAILED TO CLEAR
1839 007714 104001 ERROR
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, &CSR ;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 &CSR, RCVDAT ;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 030167 171554 15: MOV #30167, &CSR ;SET ALL R/W BITS IN CSR
1858 007760 012777 171550 CLR &CSR ;CLEAR THE CSR
1859 007766 005077 171544 001566 MOV &CSR, RCVDAT ;GET & TEST CSR
1860 007772 017737 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
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 010020 005037 001570 171514 15: MOV XMTDAT, &BKCSR ;LOAD BINARY COUNT INTO BKCSR
1875 010024 013777 001570 001566 MOV &BKCSR, RCVDAT ;GET BKCSR DATA
1876 010032 017737 171510 001566 CMP XMTDAT, RCVDAT ;COMPARE DATA LOADED & DATA READ BACK
1877 010040 023737 001570 001566 BEQ 25 ;BRANCH IF DATA COMPARES
1878 010046 001405 ERROR1 ;ERROR! DATA DID NOT COMPARE
1879 010050 104011 BIT #BIT14, &SHR ;SCOPE LOOP?
1880 010052 032777 040000 171122 BNE 15 ;BRANCH IF SCOPE LOOP
1881 010054 001361 25: MOV XMTDAT, X1 ;SAVE BINARY COUNT
1882 010060 001361 CLR XMTDAT
1883 010062 013701 001570 35: CLR &BKCSR ;CLEAR BKCSR AND TEST
1884 010066 005037 001570 171450 MOV &BKCSR, RCVDAT ;BKCSR CAN BE CLEARED
1885 010072 005077 171444 001566
1886 010076 017737 171444

```

```
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, 2SHR ;SCOPE LOOP?
1890 010116 001365 BNE 35 ;BRANCH IF SCOPE LOOP
1891 010120 010137 001570 45: MOV 21, XMTDAT ;GET BINARY COUNT
1892 010124 023727 001570 177777 CMP XMTDAT, 8-1 ;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
1905 ;TEST THAT RANDOM NUMBERS CAN BE LOADED INTO THE BKCSR
1906 010154 012702 010000 RT13A: MOV 810000, 22 ;GET RANDOM COUNTER
1907 010160 017737 171362 002104 15: MOV 2BKCSR, PRVCNT ;GET PREVIOUS CONTENTS
1908 010166 004737 003250 JSR 7, 2ARRGEN ;GO GET A RANDOM NUMBER
1909 010172 010037 001570 MOV 20, XMTDAT ;GET RANDOM NUMBER
1910 010176 013777 001570 171342 25: MOV XMTDAT, 2BKCSR ;LOAD RANDOM NUMBER INTO BKCSR
1911 010204 017737 171336 001566 MOV 2BKCSR, RCVDAT ;GET BKCSR DATA
1912 010212 023737 001570 001566 CMP XMTDAT, RCVDAT ;COMPARE DATA
1913 010220 001401 BEQ 35 ;BRANCH IF SAME
1914 010222 104011 ERROR1 ;ERROR! DATA NOT THE SAME
1915 010224 032777 040000 170750 35: BIT 8BIT14, 2SHR ;SCOPE LOOP?
1916 010232 001406 BEQ 45 ;BRANCH IF NO LOOP ON ERROR
1917 010234 005077 171306 CLR 2BKCSR ;
1918 010240 013777 002104 171300 MOV PRVCNT, 2BKCSR ;LOAD PREVIOUS CONTENTS
1919 010246 000753 BR 25 ;REPEAT TEST
1920 010250 005302 45: DEC 22
1921 010252 001342 BNE 15 ;BRANCH IF NOT
1922 010254 104006 55: SCOPE ;SCOPE
1923 ;*****
1924 010256 000014 RT14: 14 ;ROUTINE # 14
1925 010260 010316 RT15 ;ADDR OF NEXT ROUTINE.
1926 010262 000012 10. ;ITERATION COUNT
1927 010264 010266 RT14A ;SCOPE ENTRY POINT.
1928 000014 X=X+1
1929 ;*****
1930
1931 ;TEST THAT RESET CLEARS ALL BREAK STATUS REGISTER BITS
1932
1933 010266 012777 177777 171252 RT14A: MOV 8-1, 2BKCSR
1934 010274 005037 001570 CLR XMTDAT
1935 010300 104005 SRESET
1936 010302 017737 171240 001566 MOV 2BKCSR, RCVDAT
1937 010310 001401 BEQ 15
1938 010312 104011 ERROR1
1939 010314 104006 15: SCOPE
1940 ;*****
1941 010316 000015 RT15: 15 ;ROUTINE # 15
1942 010320 010452 RT16 ;ADDR OF NEXT ROUTINE.
```


1943	010322	000012				10.					
1944	010324	010326				RT15A				; ITERATION COUNT	*
1945		000015				X=X+1				; SCOPE ENTRY POINT.	*
1946											
1947											
1948											
1949											
1950	010326	005037	001570								
1951	010332	013777	001570	171210		RT15A:	CLR	XMTDAT			
1952	010340	017737	171204	001566		15:	MOV	XMTDAT, 2BASREG		; LOAD BINARY COUNT INTO BASREG	
1953	010346	023737	001570	001566			MOV	2BASREG, RCV DAT		; GET BASREG DATA	
1954	010354	001405					CMP	XMTDAT, RCV DAT		; COMPARE DATA	
1955	010356	104011					BEQ	25		; BRANCH IF DATA COMPARES	
1956	010360	032777	040000	170614			ERROR1			; ERROR! DATA DID NOT COMPARE	
1957	010366	001361					BIT	%BIT14, 2SWR		; SCOPE LOOP?	
1958	010370	013701	001570			25:	BNE	15		; BRANCH IF SCOPE LOOP	
1959	010374	005037	001570				MOV	XMTDAT, %1		; SAVE BINARY COUNT	
1960	010400	005077	171144				CLR	XMTDAT			
1961	010404	017737	171140	001566		35:	CLR	2BASREG			
1962	010412	001405					MOV	2BASREG, RCV DAT			
1963	010414	104011					BEQ	45		; BRANCH IF BKCSR CLEARED	
1964	010416	032777	040000	170556			ERROR1			; ERROR! BKCSR DID NOT CLEAR	
1965	010424	001365					BIT	%BIT14, 2SWR		; SCOPE LOOP?	
1966	010426	010137	001570			45:	BNE	35		; BRANCH IF SCOPE LOOP	
1967	010432	023727	001570	177000			MOV	%1, XMTDAT		; GET BINARY COUNT	
1968	010440	001403					CMP	XMTDAT, #177000		; ALL NUMBERS BEEN LOADED	
1969	010442	105237	001571				BEQ	55		; GO TO EXIT	
1970	010446	000731					INCB	XMTDAT+1		; INCREMENT BINARY COUNT	
1971	010450	104006					BR	15		; REPEAT TEST	
1972						55:	SCOPE			; SCOPE	
1973	010452	000016									
1974	010454	010570				RT16:	16			; ROUTINE # 16	*
1975	010456	000144					RT17			; ADDR OF NEXT ROUTINE.	*
1976	010460	010462					100.			; ITERATION COUNT	*
1977		000016					RT16A			; SCOPE ENTRY POINT.	*
1978							X=X+1				
1979											
1980											
1981	010462	012702	010000								
1982	010466	017737	171056	002104		RT16A:	MOV	#10000, %2		; GET RANDOM #COUNTER	
1983	010474	004737	003250			15:	MOV	2BASREG, PRVCNT		; GET PREVIOUS CONTENTS	
1984	010500	042700	000377				JSR	7, 2BRNGEN		; GO GET A RANDOM NUMBER	
1985	010504	010037	001570				BIC	#000377, %0		; CLEAR UNUSED BITS	
1986	010510	013777	001570	171032		25:	MOV	%0, XMTDAT		; GET RANDOM NUMBER	
1987	010516	017737	171026	001566			MOV	XMTDAT, 2BASREG		; LOAD RANDOM NUMBER INTO BASREG	
1988	010524	023737	001570	001566			MOV	2BASREG, RCV DAT		; GET BASREG DATA	
1989	010532	001401					CMP	XMTDAT, RCV DAT		; COMPARE DATA	
1990	010534	104011					BEQ	35		; BRANCH IF SAME	
1991	010536	032777	040000	170436		35:	ERROR1			; ERROR! DATA NOT THE SAME	
1992	010544	001406					BIT	%BIT14, 2SWR		; SCOPE LOOP?	
1993	010546	005077	170776				BEQ	45		; BRANCH IF NO LOOP ON ERROR	
1994	010552	013777	002104	170770			CLR	2BASREG			
1995	010560	000753					MOV	PRVCNT, 2BASREG		; LOAD PREVIOUS CONTENTS	
1996	010562	005302				45:	BR	25		; REPEAT TEST	
1997	010564	001340					DEC	%2		; 10000 NUMBERS BEEN TESTED	
1998	010566	104006				55:	BNE	15		; BRANCH IF NOT	
							SCOPE			; SCOPE	

```
1999
2000 010570 000017
2001 010572 010662
2002 010574 000144
2003 010576 010600
2004 000017
2005
2006
2007
2008 010600 013701 001544
2009 010604 012777 001400 170736
2010 010612 012700 000001
2011 010616 050011
2012 010620 020011
2013 010622 001006
2014 010624 040011
2015 010626 005711
2016 010630 001011
2017 010632 006300
2018 010634 103370
2019 010636 104006
2020 010640 010037 001570
2021 010644 011137 001566
2022 010650 104011
2023 010652 000771
2024 010654 005037 001570
2025 010660 000771
2026
2027 010662 000020
2028 010664 010750
2029 010666 000012
2030 010670 010672
2031 000020
2032
2033
2034
2035 010672 005037 001570
2036 010676 052777 177777 170640
2037 010704 104005
2038 010706 017737 170632 001566
2039 010714 001402
2040 010716 104011
2041 010720 000412
2042 010722 052777 177777 170614
2043 010730 005077 170610 001566
2044 010734 017737 170604
2045 010742 001401
2046 010744 104011
2047 010746 104006
2048
2049 010750 000021
2050 010752 011056
2051 010754 000144
2052 010756 010760
2053 000021
2054

;*****
RT17: 17 ;ROUTINE # 17
      RT20 ;ADDR OF NEXT ROUTINE.
      100. ;ITERATION COUNT
      RT17A ;SCOPE ENTRY POINT.
      X=X+1
;*****

;TEST THAT ALL BAR BITS CAN BE INDIVIDUALLY SET AND CLEARED
RT17A: MOV BAR,X1 ;GET BAR ADDRESS
      MOV #CAT,ABASREG ;INITIALIZE BASE REGISTER
      MOV #1,X0 ;GET BIT TESTER
15: BIS X0,(1) ;SET BAR BIT
     CMP X0,(1) ;TEST THAT ONLY THE PROPER BAR BIT SET
     BNE 35 ;BRANCH IF ERROR
     BIC X0,(1) ;CLEAR BAR BIT
     TST (1) ;TEST THAT BAR BIT CLEARED
     BNE 55 ;BRANCH IF BAR BIT FAILED TO CLEAR
     ASL X0 ;SHIFT BIT TESTER
     BCC 15
25: SCOPE ;SCOPE
35: MOV X0,XMTDAT ;GET WHAT DATA WAS SUPPOSED TO BE
45: MOV (1),RCVDAT ;GET WHAT DATA WAS
     ERROR1 ;ERROR! IMPROPER BIT OPERATION
55: BR 25 ;GO TO SCOPE
     CLR XMTDAT ;GET WHAT DATA WAS SUPPOSED TO BE
     BR 45
;*****
RT20: 20 ;ROUTINE # 20
      RT21 ;ADDR OF NEXT ROUTINE.
      10. ;ITERATION COUNT
      RT20A ;SCOPE ENTRY POINT.
      X=X+1
;*****

;TEST THAT RESET CLEARS ALL BAR BITS
RT20A: CLR XMTDAT
      BIS #1,ABAR ;SET ALL BAR BITS
      SRESET ;RESET
      MOV ABAR,RCVDAT ;GET BAR DATA
      BEQ 15 ;BRANCH IF ALL 0'S
      ERROR1 ;ERROR! RESET DID NOT CLEAR ALL BAR BITS
      BR 25 ;GO TO EXIT
15: BIS #1,ABAR ;SET ALL BIT IN THE BAR
     CLR ABAR ;CLEAR ALL BITS IN THE BAR
     MOV ABAR,RCVDAT ;GET & TEST RESULT OF CLEAR OPERATION
     BEQ 25 ;EXIT IF ALL BITS CLEARED
     ERROR1 ;ERROR! ALL BITS DID NOT CLEAR
25: SCOPE ;SCOPE
;*****
RT21: 21 ;ROUTINE # 21
      RT22 ;ADDR OF NEXT ROUTINE.
      100. ;ITERATION COUNT
      RT21A ;SCOPE ENTRY POINT.
      X=X+1
;*****
```



```

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
2112 011234 000415
2113 011236 012777 010100 170302 15:
2114 011244 105077 170642
2115 011250 001407
2116 011252 012737 000100 001570
2117 011260 017737 170256 001566
2118 011266 104011
2119 011270 104006
2120
2121 011272 000024
2122 011274 011336
2123 011276 000144
2124 011300 011302
2125 000024
2126
2127
2128
2129 011302 012777 011334 170246
2130 011310 012777 010000 170224
2131 011316 052777 020000 170216
2132 011324 005037 177776
2133 011330 000240
2134 011332 104001
2135
2136
2137 011334 104006
2138
2139 011336 000025
2140 011340 011444
2141 011342 000100
2142 011344 011346
2143 000025
2144
2145
2146
2147 011346 012737 000340 177776
2148 011354 012777 011404 170174
2149 011362 012777 030000 170152
2150 011370 012737 000200 177776
2151 011376 000240
2152 011400 104001
2153 011402 000417
2154 011404 022626
2155 011406 013737 001560 177776
2156 011414 012777 011440 170134
2157 011422 005077 170114
2158 011426 012777 030000 170106
2159 011434 000240
2160 011436 000401
2161 011440 104001
2162
2163 011442 104006
2164
2165 011444 000026
2166 011446 011462

ERROR1
BR 25
MOV #10100, @BKCSR
CLRB @BKCSR
BEQ 25
MOV #00100, @MTDAT
MOV @CSR, @RCDAT
ERROR1
SCOPE
; *****
RT24: 24
RT25
100
RT24A
X=X+1
; *****
; TEST THAT OVER RUN BIT (CSR BIT13) CAUSES AN INTERRUPT WHEN SET
RT24A: MOV #15, @XMTINT
MOV @BIT12, @CSR
BIS @BIT13, @CSR
CLR @PSW
NOP
ERROR
; ERROR! OVER RUN FAILED TO CAUSE AN
; INTERRUPT, OR INTERRUPTED TO INCOR-
; RECT ADDRESS
15: SCOPE
; *****
RT25: 25
RT26
100
RT25A
X=X+1
; *****
; TEST THAT THE DM11 INTERRUPTS AT THE CORRECT LEVEL
RT25A: MOV @PTY7, @PSW
MOV #15, @XMTINT
MOV #30000, @CSR
MOV @PTY4, @PSW
NOP
ERROR
BR 35
CMP (6)+, (6)+
MOV @MTLVL, @PSW
MOV #25, @XMTINT
CLR @CSR
MOV #30000, @CSR
NOP
BR 35
ERROR
35: SCOPE
; *****
RT26: 26
RT27
```

; ERROR! EVEN BYTE INSTRUCTION FAILED
; GO TO SCOPE
; LOAD TEST NUMBER IN BKCSR
; TEST THAT ONLY ODD BYTE CLEARED
; LOAD CORRECT RESULT
; LOAD ACTUAL RESULT
; ERROR! ODD BYTE INSTRUCTION FAILED
; SCOPE
; ROUTINE # 24
; ADDR OF NEXT ROUTINE.
; ITERATION COUNT
; SCOPE ENTRY POINT.
; LOAD TRANSMITTER INTERRUPT VECTOR
; SET TRANSMITTER IE BIT
; SET OVER RUN BIT
; ENABLE INTERRUPTS
; ERROR! DM11 FAILED TO INTERRUPT
; GO TO EXIT
; RESET STACK POINTER
; LOAD DM11 INTERRUPT LEVEL
; LOAD TRANSMITTER INTERRUPT VECTOR
; GO TO EXIT
; ERROR! DM11 INTERRUPTED ON HIGHER
; PRIORITY LEVEL THAN SET FOR
; ROUTINE # 26
; 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	000014		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					X=X+1
2281					*****
2282	011670	004537	004622		; TRANSMITTER LINE TEST LINE 12
2283	011674	000024			LTST12: JSR 5, XMTTST ; GO TEST TRANSMITTER LINE 12
2284		000013			LINE12
2285					Y=Y+1
2286	011676	000041			*****
2287	011700	011714			RT41: 41 ; ROUTINE # 41
2288	011702	000144			RT42 ; ADDRESS OF NEXT TEST.
2289	011704	011706			100. ; ITERATION COUNT
2290		000041			LTST13 ; SCOPE ENTRY POINT
2291					X=X+1
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			RT43 ; ADDRESS OF NEXT TEST.
2300	011722	011724			100. ; ITERATION COUNT
2301		000042			LTST14 ; 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			RT44 ; ADDRESS OF NEXT TEST.
2311	011740	011742			100. ; ITERATION COUNT
2312		000043			LTST15 ; 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			RT45 ; ADDRESS OF NEXT TEST.
2322	011756	011760			100. ; ITERATION COUNT
2323		000044			LTST16 ; 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			RT46 ; ADDRESS OF NEXT TEST.
2333	011774	011776			100. ; ITERATION COUNT
2334		000045			LTST17 ; SCOPE ENTRY POINT
					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
```


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				;RECEIVER LINE TEST LINE 11
2449	012212	004537	005034	RCV11: JSR 5,RCVTST ;GO TEST RECEIVER LINE 11
2450	012216	000022		LINE11
2451		000012		Y=Y+1
2452				*****
2453	012220	000060		RT60: 60 ;ROUTINE # 60
2454	012222	012236		RT61 ;ADDRESS OF NEXT TEST
2455	012224	000144		100. ;ITERATION COUNT
2456	012226	012230		RCV12 ;SCOPE ENTRY POINT
2457		000060		X=X+1
2458				*****
2459				;RECEIVER LINE TEST LINE 12
2460	012230	004537	005034	RCV12: JSR 5,RCVTST ;GO TEST RECEIVER LINE 12
2461	012234	000024		LINE12
2462		000013		Y=Y+1
2463				*****
2464	012236	000061		RT61: 61 ;ROUTINE # 61
2465	012240	012254		RT62 ;ADDRESS OF NEXT TEST
2466	012242	000144		100. ;ITERATION COUNT
2467	012244	012246		RCV13 ;SCOPE ENTRY POINT
2468		000061		X=X+1
2469				*****
2470				;RECEIVER LINE TEST LINE 13
2471	012246	004537	005034	RCV13: JSR 5,RCVTST ;GO TEST RECEIVER LINE 13
2472	012252	000026		LINE13
2473		000014		Y=Y+1
2474				*****
2475	012254	000062		RT62: 62 ;ROUTINE # 62
2476	012256	012272		RT63 ;ADDRESS OF NEXT TEST
2477	012260	000144		100. ;ITERATION COUNT
2478	012262	012264		RCV14 ;SCOPE ENTRY POINT
2479		000062		X=X+1
2480				*****
2481				;RECEIVER LINE TEST LINE 14
2482	012264	004537	005034	RCV14: JSR 5,RCVTST ;GO TEST RECEIVER LINE 14
2483	012270	000030		LINE14
2484		000015		Y=Y+1
2485				*****
2486	012272	000063		RT63: 63 ;ROUTINE # 63
2487	012274	012310		RT64 ;ADDRESS OF NEXT TEST
2488	012276	000144		100. ;ITERATION COUNT
2489	012300	012302		RCV15 ;SCOPE ENTRY POINT
2490		000063		X=X+1
2491				*****
2492				;RECEIVER LINE TEST LINE 15
2493	012302	004537	005034	RCV15: JSR 5,RCVTST ;GO TEST RECEIVER LINE 15
2494	012306	000032		LINE15
2495		000016		Y=Y+1
2496				*****
2497	012310	000064		RT64: 64 ;ROUTINE # 64
2498	012312	012326		RT65 ;ADDRESS OF NEXT TEST
2499	012314	000144		100. ;ITERATION COUNT
2500	012316	012320		RCV16 ;SCOPE ENTRY POINT
2501		000064		X=X+1
2502				*****

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					
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 @CAT,X1 ;GET CAT ADDRESS	
2535	012372	012702	160000			MOV @160000,X2 ;GET A NON-EXISTANT ADDRESS	
2536	012376	012703	000020			MOV @16.,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 @LBITO,X1 ;GET LINE BIT	
2541	012414	012777	012552	167134		MOV @65,@XMTINT ;LOAD TRANSMITTER INT. VECTOR	
2542	012422	052777	000060	167112		25: BIS @60,@CSR ;SET EXTENDED ADDRESS BITS	
2543	012430	050177	167110			BIS X1,@BAR ;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 @BAR,RCVDAT ;GET BAR DATA & TEST	
2548	012454	001406				BEQ 45 ;THAT IT IS CLEAR	
2549	012456	005037	001570			CLR XMTDAT	
2550	012462	104011				ERROR1 ;ERROR!BAR BIT DID NOT CLEAR	
2551	012464	005077	167054			CLR @BAR	
2552	012470	000440				BR 75 ;GO TO SCOPE	
2553	012472	032777	040000	167042		45: BIT @BIT14,@CSR ;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 @BIT15,@CSR ;CLEAR TRANSMITTER READY FLAG	
2558	012514	052777	010000	167020		BIS @BIT12,@CSR ;SET TRANSMITTER IE BIT	

```
2559 012522 005037 177776 CLR @PSW ;ALLOW INTERRUPTS
2560 012526 000240 NOP
2561 012530 012737 000340 177776 MOV #PTY7,@PSW ;LOCK OUT INTERRUPTS
2562 012536 010137 001570 MOV X1,XMTDAT ;LOAD LINE THAT FAILED
2563 012542 005037 001566 CLR RCV DAT
2564 012546 104011 ERROR1 ;ERROR! NEX FAILED TO CAUSE INTERRUPT
2565 ;TYPEOUT SHOWS LINE # THAT FAILED
2566 012550 000410 BR 75 ;GO TO SCOPE
2567 012552 005077 166764 65: CLR @CSR
2568 012556 012737 000340 177776 MOV #PTY7,@PSW ;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 ;TO COMPLETION BEFORE
2574 012602 000000 85: OPEN ;EXITING TEST
2575 012604 104006 SCOPE ;SCOPE
2576 ;*****
2577 012606 000067 RT67: 67 ;ROUTINE # 67
2578 012610 012716 RT70 ;ADDR OF NEXT ROUTINE.
2579 012612 000012 10. ;ITERATION COUNT
2580 012614 012616 RT67A ;SCOPE ENTRY POINT.
2581 000067 X=X+1
2582 ;*****
2583 ;TEST THAT NEX BIT SETS IF THE DM11 TABLES ARE IN NON-EXISTANT CORE
2584 RT67A: MOV @160000,@BASREG ;SET BASE REGISTER TO NON-EXISTANT ADDR.
2585 012616 012777 160000 166724 MOV @45,@ERRVEC ;SET TIME OUT TRAP VECTOR
2586 012624 012737 012706 000004 TST @160000 ;CHECK THAT ADDRESS TIMES OUT
2587 012632 005737 160000 MOV TIME14,15 ;GET TIME TO TRANSMIT 1/4 CHAR.
2588 012636 013737 004370 012654 BIS @LB10,@BAR ;START TO TRANSMIT ON LINE 0
2589 012644 052777 000001 166672 DELAY ;DELAY 1/4TH OF A CHARACTER
2590 012652 104400 15: OPEN ;TIME
2591 012654 000000 CLR @BAR ;STOP TRANSMITTER
2592 012656 005077 166662 166652 CMP @BIT14+60,@CSR ;TEST THAT ONLY NEX IS SET
2593 012662 022777 040060 BEQ 25
2594 012670 001401 ERROR
2595 012672 104001 ;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 @ERRVEC+2,@ERRVEC ;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 000070 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, @XMITD	; 10. CHARACTERS
2616	012736	177766			-10.		; ON EACH LINE
2617	012740	005777	166576		TST	@CSR	; WAIT FOR 10. CHARACTERS
2618	012744	100375			BPL	-4	; TO BE TRANSMITTED
2619	012746	042777	100000 166566		BIC	@100000, @CSR	
2620	012754	105777	166562		TSTB	@CSR	; 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, XMITDAT	; 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	; SET UP NEXT LINE NUMBER
2627	013010	006337	001564		ASL	LINBIT	; GET READY TO TRANSMIT ON NEXT LINE
2628	013014	103346			BCC	15	; GO TRANSMIT ON NEXT LINE
2629	013016	013737	004366 013026		MOV	TIME1, 35	
2630	013024	104400			DELAY		; DELAY 1 CHARACTER
2631	013026	000000			0		; TIME BEFORE ENTERING NEXT TEST
2632	013030	104006		35:	0		; SCOPE
2633				45:	SCOPE		; SCOPE
2634	013032	000071			*****		
2635	013034	013226		RT71:	71		; ROUTINE @ 71
2636	013036	000024			RT72		; ADDR OF NEXT ROUTINE.
2637	013040	013042			20.		; ITERATION COUNT
2638		000071			RT71A		; SCOPE ENTRY POINT.
2639					X=X+1		
2640					*****		
2641							
2642							
2643	013042	005000					; TEST THAT CURRENT ADDRESS INCREMENTS PROPERLY WHEN A CHAR-
2644	013044	010001					; ACTER IS TRANSMITTED. LINE 0 IS USED FOR THE TEST.
2645	013046	005201		RT71A:	CLR	X0	; R0=CURRENT ADDR AFTER TRANSMISSION
2646	013050	012737	013212 000004	15:	MOV	X0, X1	; R0=CURRENT ADDRESS BEFORE TRANSMISSION
2647	013056	012737	000340 000006		INC	X1	; AND R1=CURRENT ADDRESS AFTER TRANSMISSION
2648	013064	012777	001400 166456		MOV	@35, @ERRVEC	; SET UP PROCESSOR
2649	013072	010037	001400		MOV	@PTY7, @ERRVEC+2	; TIME OUT TRAP
2650	013076	105710			MOV	@CAT, @BASREG	; SET UP BASE REGISTER
2651	013100	012737	177776 001440		MOV	X0, CAT	; LOAD CURRENT ADDRESS TABLE (LINE 0)
2652	013106	012777	000005 166426		TSTB	(0)	; DOES MEMORY EXIST?
2653	013114	012777	000001 166422		MOV	@-2, WCT	; SET CHAR. COUNT TO TRANSMIT 1 CHAR.
2654	013122	105777	166414		MOV	@5, @CSR	; SET MAINT & GO BITS
2655	013126	100375			MOV	@LBIT0, @BAR	; TRANSMIT ON LINE 0
2656	013130	042777	000200 166404		TSTB	@CSR	; WAIT FOR THE RECEIVER
2657	013136	105777	166400		BPL	-4	; TO RECEIVE FIRST CHARACTER
2658	013142	100375			BIC	@200, @CSR	; CLEAR RECEIVER DONE FLAG
2659	013144	023701	001400		TSTB	@CSR	; WAIT FOR RECEIVER TO RECEIVE
2660					BPL	-4	; THE SECOND CHARACTER
2661	013150	001413			CMP	CAT, X1	; TEST THAT CURRENT ADDR
2662	013152	010137	001570 001566				; INCREMENTED PROPERLY
2663	013156	013737	001400		BEQ	25	
2664	013164	104011			MOV	X1, XMITDAT	; GET COMPUTED RESULT
2665	013166	032777	040000 166006		MOV	CAT, RCVDAT	; GET ACTUAL RESULT
2666	013174	001323			ERROR1		; ERROR! CURRENT ADDRESS DID NOT
2667	013176	000405			BIT	@BIT14, @SHR	; INCREMENT PROPERLY
2668	013200	005701			BNE	15	; BRANCH IF SCOPE SWITCH IS SET
2669	013202	001403		25:	BR	35	; GO TO EXIT
2670	013204	000261			TST	X1	
					BEQ	35	
					SEC		

```

2671 013206 006100
2672 013210 100715
2673 013212 012737 000006 000004 35:
2674 013220 005037 000006
2675 013224 104006
2676
2677 013226 000072
2678 013230 013572
2679 013232 000024
2680 013234 013236
2681 000072
2682
2683
2684
2685
2686
2687 013236 005000
2688 013240 000005
2689 013242 016037 013526 001400
2690 013250 012777 000005 166264
2691 013256 012737 177777 001440
2692 013264 012777 000001 166252
2693 013272 005777 166244
2694 013276 100375
2695 013300 105777 166236
2696 013304 100375
2697 013306 005077 166230
2698 013312 005037 001566
2699 013316 113737 001600 001566
2700 013324 117037 013526 001570
2701 013332 043737 001572 001570
2702 013340 123737 001566 001570
2703 013346 001402
2704 013350 104011
2705
2706 013352 000464
2707 013354 020027 000006
2708 013360 001402
2709 013362 005720
2710 013364 000726
2711
2712
2713 013366
2714 013366 012737 013514 000004
2715
2716 013374 005001
2717 013376 005201
2718 013400 005720
2719 013402 110170 013526
2720 013406 016037 013526 001400
2721 013414 012777 000005 166120
2722 013422 012737 177777 001440
2723 013430 012777 000001 166106
2724 013436 005777 166100
2725 013442 100375
2726 013444 105777 166072

ROL X0
BMI 15
MOV #ERRVEC+2, @ERRVEC ; RESTORE TIME OUT TRAP
CLR @ERRVEC+2
SCOPE ; SCOPE
; *****
RT72: 72 ; ROUTINE # 72
RT73 ; ADDR OF NEXT ROUTINE.
20. ; ITERATION COUNT
RT72A ; SCOPE ENTRY POINT.
X=X+1
; *****

; TEST THAT DATA CAN BE TRANSMITTED FROM ALL AVAILABLE CORE.
; LINE 0 IS USED FOR THE TEST AND ONLY ONE WORD IS TRANSMITTED
; AT A TIME.
RT72A: CLR X0 ; CLEAR INDEX REGISTER
RESET
15: MOV AREA(0), CAT ; LOAD CURRENT ADDRESS
MOV #5, @CSR ; SET MAINT & GO BITS
MOV #1, WCT ; SET UP CHAR COUNT TO TRANSMIT 1 CHAR.
MOV @LBTO, @BAR ; TRANSMIT CHAR ON LINE 0
TST @CSR ; WAIT FOR THE TRANSMITTER
BPL -4 ; TO TRANSMIT THE CHARACTER
TSTB @CSR ; TEST FOR DONE
BPL -4
CLR @CSR ; CLEAR ALL FLAGS
CLR RCV DAT
MOV @TUNTAB, RCV DAT ; GET RECEIVED CHARACTER
MOV @AREA(0), XMT DAT ; GET TRANSMITTED CHARACTER
BIC @CARMSK, XMT DAT ; CLEAR NON-TRANSMITTED BITS
CMPB RCV DAT, XMT DAT ; COMPARE CHARACTERS
BEQ 25 ; BRANCH IF VALID COMPARISON
ERROR1 ; ERROR! DATA COMPARISON ERROR
; ((CAT)-1 IS THE MEMORY LOCATION WHERE THE DATA WAS TRANSMITTED FROM
BR 65 ; GO TO EXIT
25: CMP X0, #6 ; HAS FIRST 4K BEEN TESTED
BEQ 35 ; BRANCH IF IT HAS
TST (0)+ ; INCREMENT INDEX
BR 15 ; GO REPEAT TEST

35: MOV #55, @ERRVEC ; BEGIN TESTING ABOVE 4K
; SET TIME OUT TRAP TO EXIT
; TEST IF MEMORY TIMES OUT
45: INC X1 ; SET UP DATA IDENTIFIER
TST (0)+ ; INCREMENT DATA IDENTIFIER
MOV X1, @AREA(0) ; INCREMENT INDEX
MOV AREA(0), CAT ; LOAD IDENTIFIER INTO MEMORY
MOV #5, @CSR ; LOAD CURRENT ADDRESS
MOV #1, WCT ; SET MAINT & GO BITS
MOV @LBTO, @BAR ; SET UP CHAR COUNT TO TRANSMIT 1 CHAR.
TST @CSR ; TRANSMIT ON LINE 0
BPL -4 ; WAIT FOR THE TRANSMITTER TO
TSTB @CSR ; TRANSMIT THE CHARACTER
; TEST FOR CHARACTER DONE

```


2727	013450	100375			BPL	-4	
2728	013452	005077	166064		CLR	ACSR	
2729	013456	113737	001600	001566	MOVB	TUMTAB, RCV DAT	; GET THE RECEIVED CHARACTER
2730	013464	117037	013526	001570	MOVB	AREA(0), XMT DAT	; GET THE TRANSMITTED CHARACTER
2731	013472	043737	001572	001570	BIC	CARMSK, XMT DAT	; CLEAR NON-TRANSMITTED BITS
2732	013500	123737	001566	001570	CMPS	RCV DAT, XMT DAT	; COMPARE CHARACTERS
2733	013506	001733			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, @ERRVEC	; RESTORE TIME OUT TRAP
2738	013524	104006			SCOPE		; EXIT TEST
2739							
2740	013526	000000					
2741	013530	005252					
2742	013532	012525					
2743	013534	017777					
2744	013536	020000					
2745	013540	026314					
2746	013542	031463					
2747							
2748	013544	037477					
2749	013546	040000					
2750	013550	057477					
2751	013552	060000					
2752	013554	077477					
2753	013556	100000					
2754	013560	117477					
2755	013562	120000					
2756	013564	137477					
2757	013566	140000					
2758	013570	173000					
2759							
2760	013572	000073					
2761	013574	014006					
2762	013576	000012					
2763	013600	013602					
2764		000073					
2765							
2766							
2767							
2768							
2769	013602	005037	016730				
2770	013606	012777	000001	165726			
2771	013614	005037	001566				
2772	013620	013737	016730	001570			
2773	013626	004537	006246				
2774	013632	177634					
2775	013634	106777	165702				
2776	013640	100375					
2777	013642	042777	000200	165672			
2778	013660	005237	001566				
2779	013664	023727	001566	000144			
2780	013662	001416					
2781	013664	005777	165652				
2782	013670	100002					

```

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: DELAY TIME14, 55 ;EXIT TEST
;LOCATION LINE IT HAS THE CORRECT BAR CONTENTS. ;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
2802 013750 001407 BEQ 75 ;ERROR! READY FLAG FAILED TO SET
;GO TO EXIT
2803 013752 017737 165566 001566 MOV @BAR, RCV DAT ;TEST THAT BAR BIT IS CLEAR
;GO TO 75 IF CLEAR
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 052777 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
BNE 15
MOV @TUMTAB, X1
MOV @BIT2, @CSR ;SET MAINT BIT & CLEAR GO BIT
CLR XMTDAT
CLR RCV DAT
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 RCV DAT ;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

Line	Address	Instruction	Comment
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	

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				*****

```
3007 ;BREAK TEST ON LINE 15.
3008 014472 004537 005456 BRK15: JSR 5, BRKTST ;GO DO BREAK TEST
3009 014476 020000 ;ON LINE 15
3010 000016
3011 ;*****
3012 014500 000113 RT113: 113 ;ROUTINE # 113
3013 014502 014516 ;ADDRESS OF NEXT TEST
3014 014504 000144 100. ;ITERATION COUNT
3015 014506 014510 BRK16 ;SCOPE ENTRY POINT
3016 000113
3017 X=X+1
3018 ;*****
3019 014510 004537 005456 ;BREAK TEST ON LINE 16.
3020 014514 040000 BRK16: JSR 5, BRKTST ;GO DO BREAK TEST
3021 000017 ;ON LINE 16
3022 Y=Y+1
3023 014516 000114 ;*****
3024 014520 014534 RT114: 114 ;ROUTINE # 114
3025 014522 000144 ;ADDRESS OF NEXT TEST
3026 014524 014526 100. ;ITERATION COUNT
3027 000114 BRK17 ;SCOPE ENTRY POINT
3028 X=X+1
3029 ;*****
3030 014526 004537 005456 ;BREAK TEST ON LINE 17.
3031 014532 100000 BRK17: JSR 5, BRKTST ;GO DO BREAK TEST
3032 000020 ;ON LINE 17
3033 000000
3034 000000
3035 Y=0
3036 014534 000115 ;*****
3037 014536 014552 RT115: 115 ;ROUTINE #115
3038 014540 000144 ;ADDRESS OF NEXT TEST
3039 014542 014544 100. ;ITERATION COUNT
3040 000115 DAT0 ;SCOPE ENTRY POINT
3041 X=X+1
3042 ;*****
3043 014544 004537 006312 ;DATA TEST 100 CHARACTERS LINE0
3044 014550 000000 DAT0: JSR 5, DATTST ;GO RUN DATA TEST
3045 000001 ;ON LINE0
3046 Y=Y+1
3047 014552 000116 ;*****
3048 014554 014570 RT116: 116 ;ROUTINE #116
3049 014556 000144 ;ADDRESS OF NEXT TEST
3050 014560 014562 100. ;ITERATION COUNT
3051 000116 DAT1 ;SCOPE ENTRY POINT
3052 X=X+1
3053 ;*****
3054 014562 004537 006312 ;DATA TEST 100 CHARACTERS LINE1
3055 014566 000002 DAT1: JSR 5, DATTST ;GO RUN DATA TEST
3056 000002 ;ON LINE1
3057 Y=Y+1
3058 014570 000117 ;*****
3059 014572 014606 RT117: 117 ;ROUTINE #117
3060 014574 000144 ;ADDRESS OF NEXT TEST
3061 014576 014600 100. ;ITERATION COUNT
3062 000117 DAT2 ;SCOPE ENTRY POINT
X=X+1
```


3063			
3064			
3065	014600	004537	006312
3066	014604	000004	
3067		000003	
3068			
3069	014606	000120	
3070	014610	014624	
3071	014612	000144	
3072	014614	014616	
3073		000120	
3074			
3075			
3076	014616	004537	006312
3077	014622	000006	
3078		000004	
3079			
3080	014624	000121	
3081	014626	014642	
3082	014630	000144	
3083	014632	014634	
3084		000121	
3085			
3086			
3087	014634	004537	006312
3088	014640	000010	
3089		000005	
3090			
3091	014642	000122	
3092	014644	014660	
3093	014646	000144	
3094	014650	014652	
3095		000122	
3096			
3097			
3098	014652	004537	006312
3099	014656	000012	
3100		000006	
3101			
3102	014660	000123	
3103	014662	014676	
3104	014664	000144	
3105	014666	014670	
3106		000123	
3107			
3108			
3109	014670	004537	006312
3110	014674	000014	
3111		000007	
3112			
3113	014676	000124	
3114	014700	014714	
3115	014702	000144	
3116	014704	014706	
3117		000124	
3118			

```

;*****
;DATA TEST 100 CHARACTERS LINE2
DAT2:  JSR      5,DATTST      ;GO RUN DATA TEST
        LINE2      ;ON LINE2
        Y=Y+1
;*****
RT120:  120      ;ROUTINE #120
        RT121      ;ADDRESS OF NEXT TEST
        100.      ;ITERATION COUNT
        DAT3      ;SCOPE ENTRY POINT
        X=X+1
;*****
;DATA TEST 100 CHARACTERS LINE3
DAT3:  JSR      5,DATTST      ;GO RUN DATA TEST
        LINE3      ;ON LINE3
        Y=Y+1
;*****
RT121:  121      ;ROUTINE #121
        RT122      ;ADDRESS OF NEXT TEST
        100.      ;ITERATION COUNT
        DAT4      ;SCOPE ENTRY POINT
        X=X+1
;*****
;DATA TEST 100 CHARACTERS LINE4
DAT4:  JSR      5,DATTST      ;GO RUN DATA TEST
        LINE4      ;ON LINE4
        Y=Y+1
;*****
RT122:  122      ;ROUTINE #122
        RT123      ;ADDRESS OF NEXT TEST
        100.      ;ITERATION COUNT
        DAT5      ;SCOPE ENTRY POINT
        X=X+1
;*****
;DATA TEST 100 CHARACTERS LINE5
DAT5:  JSR      5,DATTST      ;GO RUN DATA TEST
        LINE5      ;ON LINE5
        Y=Y+1
;*****
RT123:  123      ;ROUTINE #123
        RT124      ;ADDRESS OF NEXT TEST
        100.      ;ITERATION COUNT
        DAT6      ;SCOPE ENTRY POINT
        X=X+1
;*****
;DATA TEST 100 CHARACTERS LINE6
DAT6:  JSR      5,DATTST      ;GO RUN DATA TEST
        LINE6      ;ON LINE6
        Y=Y+1
;*****
RT124:  124      ;ROUTINE #124
        RT125      ;ADDRESS OF NEXT TEST
        100.      ;ITERATION COUNT
        DAT7      ;SCOPE ENTRY POINT
        X=X+1
;*****

```

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				
3219				TEST THAT DATA (ALL 1'S) CAN BE TRANSMITTED ON LINES SIMULTANEOUSLY.
3220				THE FOLLOWING TESTS ARE PERFORMED:
3221				THERE ARE 16. DATA ENTRIES
3222				THERE ISN'T A 17TH ENTRY
3223				DATA RECEIVED IS CORRECT
3224				ONE DATA ENTRY PER LINE
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	#LBIT0,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
3259	015256	012702	000020		MOV	#16.,X2	;CLEAR THE CSR
3260	015262	012701	001600		MOV	#TUNTAB,X1	;GET TT SCAN COUNT
3261	015266	005302			55:	DEC	X2
3262	015270	100404			BMI	65	;GET FIRST TT ADDRESS
3263	015272	005721			TST	(1)+	;DECREMENT SCAN COUNTER
3264	015274	100774			BMI	55	;BRANCH IF 16. ENTRIES SCANNED
3265	015276	104001			ERROR		;TEST FOR VALID DATA ENTRY
3266	015300	000474			BR	165	;BRANCH IF FOUND
3267	015302	005721			65:	TST	(1)+
3268	015304	001402			BEQ	75	;ERROR! MISSING DATA ENTRY
3269	015306	104001			ERROR		;GO TO EXIT
3270	015310	000470			BR	165	;TEST 17TH ENTRY (SHOULD BE = TO 0)
3271							;BRANCH IF 0
3272							;ERROR! EXTRA DATA ENTRY
3273	015312	012701	001600		75:	MOV	#TUNTAB,X1
3274	015316	012702	000020		MOV	#16.,X2	;GET FIST TT ADDRESS
3275	015322	005302			85:	DEC	X2
3276	015324	100421			BMI	105	;GET SCAN COUNT
3277	015326	013737	017102	001570	MOV	OUTBUF,XMTDAT	;DECREMENT SCAN COUNT
3278	015334	043737	001572	001570	BIC	CARMSK,XMTDAT	;BRANCH IF 16. ENTRIES SCANNED
3279	015342	113737	001600	001566	MOVB	TUNTAB,RCVDAT	;GET TRANSMITTED DATA
3280	015360	123737	001570	001566	CMPB	XMTDAT,RCVDAT	;CLEAR NON-TRANSMITTED BITS
3281	015364	001402			BEQ	95	;GET RECEIVED DATA
3282	015360	104011			ERROR1		;COMPARE DATA
3283	015362	000443			BR	165	;ERROR INCORRECT DATA
3284	015364	005721			95:	TST	(1)+
3285	015366	000755			BR	85	;GO TO EXIT
3286							;INCREMENT TT ADDRESS
							;TEST NEXT ENTRY

3287					;CLEAR ALL BUT LINE NUMBER IN TUMBLE TABLE ENTRY
3288	015370	012701	001600		105: MOV #TUMTAB,X1 ;GET FIRST TT ADDRESS
3289	015374	012702	000020		MOV #16.,X2 ;GET SCAN COUNT
3290	015400	005302			115: DEC X2 ;DECREMENT SCAN COUNT
3291	015402	100403			BMI 125 ;BRANCH IF ALL LINES TESTED
3292	015404	042721	160777		BIC #160777,(1)+ ;CLEAR ALL BUT LINE NUMBER IN TT
3293	015410	000773			BR 115 ;DO NEXT TT ADDRESS
3294					
3295					
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.,X1
3299	015426	012700	001600		135: MOV #16.,X2
3300	015432	023720	001570		MOV #TUMTAB,X0
3301	015436	001406			145: CMP XMTDAT,(0)+ ;TEST FOR LINE ENTRY
3302	015440	005302			BEQ 155 ;BRANCH IF FOUND
3303	015442	001373			DEC X2 ;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 X1 ;GO TO EXIT
3309	015460	001404			TST X1 ;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					165: SCOPE ;SCOPE
3314	015474	000136			;*****
3315	015476	016010			RT136: 136 ;ROUTINE # 136
3316	015500	000144			RT137 ;ADDR OF NEXT ROUTINE.
3317	015502	015504			100. ;ITERATION COUNT
3318		000136			RT136A ;SCOPE ENTRY POINT.
3319					X=X+1
3320					;*****
3321					
3322	015504	013737	004366	015560	;TEST THAT THE DM11 CAN TRANSMIT A BREAK ON ALL LINES SIMULTANEOUSLY
3323	015512	005037	001600		RT136A: MOV #TIME1,IS ;GET TIME TO TRANSMIT ONE CHARACTER
3324	015516	004537	006224		CLR TUMTAB ;CLEAR
3325	015522	001600			JSR 5,BMOVE ;THE
3326	015524	001601			TUMTAB ;TUMBLE
3327	015526	000177			TUMTAB+1 ;TABLE
3328	015530	012777	000001	164004	177
3329	015536	012777	177777	164002	MOV #BITO,@CSR ;SET GO
3330	015544	105777	163772		MOV @-1,@KCSR ;SET BREAK BIT FOR ALL LINES
3331	015550	100375			TSTB @CSR ;WAIT FOR THE RECEIVER
3332	015552	005077	163770		BPL -4 ;TO RECEIVE A BREAK
3333	015556	104400			CLR @KCSR ;CLEAR ALL BREAK BITS
3334	015560	000000			DELAY ;WAIT ONE CHARACTER
3335	015562	022777	000201	163752	15: OPEN ;TIME
3336	015570	001410			CMP #201,@CSR ;TEST THAT ONLY GO AND DONE ARE SET
3337	015572	017737	163744	001566	BEQ 25
3338	015600	012737	000201	001570	MOV @CSR,RCVDAT ;GET CSR ENTRY
3339	015606	104011			MOV #201,XMTDAT ;GET CORRECT RESULT
3340	015610	000476			ERROR1 ;ERROR! INCORRECT CSR DATA
3341					BR 135 ;EXIT
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							
3353	015636	012701	001600				; TEST THAT THE BREAK BIT IS SET IN 16. TUMBLE TABLE ENTRIES
3354	015642	012702	000020		MOV	#TUMTAB,X1	
3355	015646	032721	040000	55:	MOV	#16,X2	
3356	015652	001002			BIT	#BIT14,(1)+	; BREAK BIT SET?
3357	015654	104001			BNE	65	; BRANCH IF SET
3358	015656	000453			ERROR		; ERROR! MISSING BREAK BIT
3359	015660	005302		65:	BR	135	; EXIT
3360	015662	001371			DEC	X2	; DECREMENT SCAN COUNT
3361					BNE	55	
3362							
3363	015664	012701	001600				; TEST THAT THE TUMBLE TABLE DATA BYTE IS ALL 0'S
3364	015670	012702	000020		MOV	#TUMTAB,X1	
3365	015674	105721		75:	MOV	#16,X2	
3366	015676	001402			TSTB	(1)+	; TEST DATA BYTE
3367	015700	104001			BEQ	85	; BRANCH IF 0'S
3368	015702	000441			ERROR		; ERROR! INCORRECT DATA
3369	015704	105721		85:	BR	135	; EXIT
3370	015706	005302			TSTB	(1)+	; STEP TABLE POINTER TO NEXT DATA BYTE
3371	015710	001371			DEC	X2	
3372					BNE	75	
3373							
3374	015712	012701	001600				; CLEAR ALL BUT LINE NUMBER IN TUMBLE TABLE ENTRY
3375	015716	012702	000020		MOV	#TUMTAB,X1	
3376	015722	042721	160777	95:	MOV	#16,X2	
3377	015726	005302			BIC	#160777,(1)+	; CLEAR ALL BUT LINE NUMBER
3378	015730	001374			DEC	X2	
3379					BNE	95	
3380							
3381	015732	005004					; TEST THAT THERE IS A TUMBLE TABLE ENTRY FOR EACH LINE
3382	015734	012703	000020		CLR	X4	; CLEAR LINE NUMBER
3383	015740	012702	000020	105:	MOV	#16,X3	
3384	015744	012701	001600		MOV	#16,X2	
3385	015750	020421			MOV	#TUMTAB,X1	
3386	015752	001410		115:	CMP	X4,(1)+	; TEST FOR LINE ENTRY FOR THIS LINE
3387	015754	005302			BEQ	125	; BRANCH IF FOUND
3388	015756	001374			DEC	X2	
3389	015760	010437	001570		BNE	115	
3390	015764	010437	001566		MOV	X4,XMTDAT	
3391	015770	104011			MOV	X4,RCVDAT	
3392	015772	000405			ERROR1		; ERROR! NO LINE ENTRY FOUND FOR THIS LINE
3393	015774	005303		125:	BR	135	; EXIT
3394	015776	001403			DEC	X3	; ALL LINES BEEN FOUND
3395	016000	062704	001000		BEQ	135	; EXIT IF YES
3396	016004	000755			ADD	#1000,X4	; SEARCH FOR
3397	016006	104006		135:	BR	105	; NEXT LINE
3398					SCOPE		; SCOPE

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 ;***** ;THIS MUCH BETWEEN LINES
3459 016116 000144 ;*****
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
3513          X=X+1
3514          ;*****
3515          ;TEST THAT THE DM11 RESPONDS CORRECTLY TO A RESET
3516          RT146A: MOV      #OUTBUF,CAT      ;SET UP TO TRANSMIT 10
3517                   MOV      @-10,WCT      ;CHARACTERS ON LINE 0
3518                   MOV      TIME1,25      ;GET TIME TO TRANSMIT 2 CHARACTERS
3519                   MOV      TIME1,65
3520                   CLR      XMTDAT
3521                   MOV      @7,@CSR      ;SET MAINT., HALF DUPLEX & GO BITS
3522                   MOV      @LBIT0,@BAR   ;START TO TRANSMIT ON LINE 0
3523                   MOV      @LAIT1,@BKCSR ;BREAK ON LINE 1
3524                   MOV      @2,%4
3525                   15:    DELAY              ;WAIT 2 CHARACTER
3526                   25:    OPEN              ;TIMES
3527                   DEC      %4
3528                   BNE      15
3529                   SRESET
3530                   MOV      @CSR,RCVDAT    ;RESET
3531                   BEQ      35              ;GET CSR CONTENTS
3532                   ERROR1
3533                   MOV      @BAR,RCVDAT    ;BRANCH IF 0
3534                   BEQ      45              ;ERROR! CSR DID NOT CLEAR
3535                   ERROR1
3536                   BR       55              ;GET BAR CONTENTS
3537                   MOV      @8,%4          ;BRANCH IF 0
3538                   45:    DELAY              ;ERROR! BAR IS NOT CLEAR
3539                   55:    OPEN              ;EXIT
3540                   65:    DEC      %4
3541                   BNE      55
3542                   MOV      @CSR,RCVDAT    ;WAIT 8 MORE CHARACTER TIMES
3543                   BEQ      75
3544                   ERROR1
3545                   BR       95
3546                   MOV      @BAR,RCVDAT    ;TEST THAT CSR IS CLEAR
3547                   BEQ      85
3548                   ERROR1
3549                   BR       95
3550                   MOV      @BKCSR,RCVDAT  ;ERROR! CSR WAS NOT CLEAR
3551                   BEQ      95              ;GO TO EXIT
3552                   ERROR1                  ;TES THAT BAR IS CLEAR
3553                   SCOPE
3554                   75:
3555                   85:
3556                   95:
3557                   95:
3558                   95:
3559                   95:
3560                   95:
3561                   95:
3562                   95:
3563                   95:
3564                   95:
3565                   95:
3566                   95:
3567                   95:
3568                   95:
3569                   95:
3570                   95:
3571                   95:
3572                   95:
3573                   95:
3574                   95:
3575                   95:
3576                   95:
3577                   95:
3578                   95:
3579                   95:
3580                   95:
3581                   95:
3582                   95:
3583                   95:
3584                   95:
3585                   95:
3586                   95:
3587                   95:
3588                   95:
3589                   95:
3590                   95:
3591                   95:
3592                   95:
3593                   95:
3594                   95:
3595                   95:
3596                   95:
3597                   95:
3598                   95:
3599                   95:
3600                   95:
3601                   95:
3602                   95:
3603                   95:
3604                   95:
3605                   95:
3606                   95:
3607                   95:
3608                   95:
3609                   95:
3610                   95:
3611                   95:
3612                   95:
3613                   95:
3614                   95:
3615                   95:
3616                   95:
3617                   95:
3618                   95:
3619                   95:
3620                   95:
3621                   95:
3622                   95:
3623                   95:
3624                   95:
3625                   95:
3626                   95:
3627                   95:
3628                   95:
3629                   95:
3630                   95:
3631                   95:
3632                   95:
3633                   95:
3634                   95:
3635                   95:
3636                   95:
3637                   95:
3638                   95:
3639                   95:
3640                   95:
3641                   95:
3642                   95:
3643                   95:
3644                   95:
3645                   95:
3646                   95:
3647                   95:
3648                   95:
3649                   95:
3650                   95:
3651                   95:
3652                   95:
3653                   95:
3654                   95:
3655                   95:
3656                   95:
3657                   95:
3658                   95:
3659                   95:
3660                   95:
3661                   95:
3662                   95:
3663                   95:
3664                   95:
3665                   95:
3666                   95:
3667                   95:
3668                   95:
3669                   95:
3670                   95:
3671                   95:
3672                   95:
3673                   95:
3674                   95:
3675                   95:
3676                   95:
3677                   95:
3678                   95:
3679                   95:
3680                   95:
3681                   95:
3682                   95:
3683                   95:
3684                   95:
3685                   95:
3686                   95:
3687                   95:
3688                   95:
3689                   95:
3690                   95:
3691                   95:
3692                   95:
3693                   95:
3694                   95:
3695                   95:
3696                   95:
3697                   95:
3698                   95:
3699                   95:
3700                   95:
3701                   95:
3702                   95:
3703                   95:
3704                   95:
3705                   95:
3706                   95:
3707                   95:
3708                   95:
3709                   95:
3710                   95:
3711                   95:
3712                   95:
3713                   95:
3714                   95:
3715                   95:
3716                   95:
3717                   95:
3718                   95:
3719                   95:
3720                   95:
3721                   95:
3722                   95:
3723                   95:
3724                   95:
3725                   95:
3726                   95:
3727                   95:
3728                   95:
3729                   95:
3730                   95:
3731                   95:
3732                   95:
3733                   95:
3734                   95:
3735                   95:
3736                   95:
3737                   95:
3738                   95:
3739                   95:
3740                   95:
3741                   95:
3742                   95:
3743                   95:
3744                   95:
3745                   95:
3746                   95:
3747                   95:
3748                   95:
3749                   95:
3750                   95:
3751                   95:
3752                   95:
3753                   95:
3754                   95:
3755                   95:
3756                   95:
3757                   95:
3758                   95:
3759                   95:
3760                   95:
3761                   95:
3762                   95:
3763                   95:
3764                   95:
3765                   95:
3766                   95:
3767                   95:
3768                   95:
3769                   95:
3770                   95:
3771                   95:
3772                   95:
3773                   95:
3774                   95:
3775                   95:
3776                   95:
3777                   95:
3778                   95:
3779                   95:
3780                   95:
3781                   95:
3782                   95:
3783                   95:
3784                   95:
3785                   95:
3786                   95:
3787                   95:
3788                   95:
3789                   95:
3790                   95:
3791                   95:
3792                   95:
3793                   95:
3794                   95:
3795                   95:
3796                   95:
3797                   95:
3798                   95:
3799                   95:
3800                   95:
3801                   95:
3802                   95:
3803                   95:
3804                   95:
3805                   95:
3806                   95:
3807                   95:
3808                   95:
3809                   95:
3810                   95:
3811                   95:
3812                   95:
3813                   95:
3814                   95:
3815                   95:
3816                   95:
3817                   95:
3818                   95:
3819                   95:
3820                   95:
3821                   95:
3822                   95:
3823                   95:
3824                   95:
3825                   95:
3826                   95:
3827                   95:
3828                   95:
3829                   95:
3830                   95:
3831                   95:
3832                   95:
3833                   95:
3834                   95:
3835                   95:
3836                   95:
3837                   95:
3838                   95:
3839                   95:
3840                   95:
3841                   95:
3842                   95:
3843                   95:
3844                   95:
3845                   95:
3846                   95:
3847                   95:
3848                   95:
3849                   95:
3850                   95:
3851                   95:
3852                   95:
3853                   95:
3854                   95:
3855                   95:
3856                   95:
3857                   95:
3858                   95:
3859                   95:
3860                   95:
3861                   95:
3862                   95:
3863                   95:
3864                   95:
3865                   95:
3866                   95:
3867                   95:
3868                   95:
3869                   95:
3870                   95:
3871                   95:
3872                   95:
3873                   95:
3874                   95:
3875                   95:
3876                   95:
3877                   95:
3878                   95:
3879                   95:
3880                   95:
3881                   95:
3882                   95:
3883                   95:
3884                   95:
3885                   95:
3886                   95:
3887                   95:
3888                   95:
3889                   95:
3890                   95:
3891                   95:
3892                   95:
3893                   95:
3894                   95:
3895                   95:
3896                   95:
3897                   95:
3898                   95:
3899                   95:
3900                   95:
3901                   95:
3902                   95:
3903                   95:
3904                   95:
3905                   95:
3906                   95:
3907                   95:
3908                   95:
3909                   95:
3910                   95:
3911                   95:
3912                   95:
3913                   95:
3914                   95:
3915                   95:
3916                   95:
3917                   95:
3918                   95:
3919                   95:
3920                   95:
3921                   95:
3922                   95:
3923                   95:
3924                   95:
3925                   95:
3926                   95:
3927                   95:
3928                   95:
3929                   95:
3930                   95:
3931                   95:
3932                   95:
3933                   95:
3934                   95:
3935                   95:
3936                   95:
3937                   95:
3938                   95:
3939                   95:
3940                   95:
3941                   95:
3942                   95:
3943                   95:
3944                   95:
3945                   95:
3946                   95:
3947                   95:
3948                   95:
3949                   95:
3950                   95:
3951                   95:
3952                   95:
3953                   95:
3954                   95:
3955                   95:
3956                   95:
3957                   95:
3958                   95:
3959                   95:
3960                   95:
3961                   95:
3962                   95:
3963                   95:
3964                   95:
3965                   95:
3966                   95:
3967                   95:
3968                   95:
3969                   95:
3970                   95:
3971                   95:
3972                   95:
3973                   95:
3974                   95:
3975                   95:
3976                   95:
3977                   95:
3978                   95:
3979                   95:
3980                   95:
3981                   95:
3982                   95:
3983                   95:
3984                   95:
3985                   95:
3986                   95:
3987                   95:
3988                   95:
3989                   95:
3990                   95:
3991                   95:
3992                   95:
3993                   95:
3994                   95:
3995                   95:
3996                   95:
3997                   95:
3998                   95:
3999                   95:
4000                   95:

```

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 PRG2B: BR PRG2A
3590 016662 005077 162656 CLR @CSR
3591 016666 005037 016714 CLR @BAR
3592 016672 017737 162304 016714 CLR PRG2C
3593 016700 042737 000377 016714 MOV @SHR, PRG2C
3594 016706 000337 016714 BIC #377, PRG2C
3595 016712 104400 SWAB PRG2C
3596 016714 000000 PRG2C: DELAY
3597 016716 000732 OPEN
3598 BR PRG2R
3599
3600 ;SUBROUTINE TO GET USER PARAMETERS (FOR PRG1 & 2)
3601 016720 104000 PARAM: TYPE
3602 016722 020154 LINPAR
3603 016724 004537 004416 JSR 5, RECD
3604 016730 000000 LINE: 0
3605 016732 104000 PARAM: TYPE
3606 016734 020177 HOWMAN
3607 016736 004537 004416 JSR 5, RECD
3608 016742 000000 CHARS: 0
3609 016744 023727 016742 000310 CMP CHARS, #200.
3610 016752 101403 BLOS PARAMB
3611 016754 104000 TYPE
3612 016756 020017 M1
3613 016760 000764 BR PARAMA
3614 016762 104000 PARAMB: TYPE
3615 016764 020043 M4
3616 016766 104015 CNTLU
3617 016770 000207 RTS 7
3618
3619
3620 ;SUBROUTINE TO TRANSMIT DATA FROM THE SR
3621 016772 117737 162204 017102 LOOP: MOV @SHR, OUTBUF
3622 017000 004537 006224 JSR 5, BMOVE
3623 017004 017102 OUTBUF
3624 017006 017103 OUTBUF+1
3625 017010 000307 199.
3626 017012 012777 001400 162530 MOV #CAT, @BASREG
  
```

;;++D
 ;;++D
 ;;++D

;DELAY AS SPECIFIED
 ;BY USER
 ;REPEAT LOOP

;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      ;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      ;NOP
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

```


3683	017560	017641			\$SWREG		
3684	017562	004537	006154		JSR	R5,OACNV	; CONVERT TO ASCII
3685	017566	000176			SWREG		
3686	017570	017647			SVALUE		
3687	017572	000006			B		
3688	017574	104000			TYPE		
3689	017576	017647			SVALUE		
3690	017600	004537	004416		JSR	5.RECD	; GET TMP1 AND PUT IT
3691	017604	000000		TMP1:	O		; HERE
3692	017606	022737	000007	004574	CMP	#7,CNT	
3693	017614	001403			BEQ	FJX	
3694	017616	013777	017604	161356	MOV	TMP1,\$SWR	; CHANGE CONTENTS OF SWREG
3695	017624	000002		FJX:	RTI		
3696							

Address	Hex 1	Hex 2	Hex 3	Hex 4	Hex 5	Hex 6	Hex 7	Hex 8	Label	Comment
3697										
3698	017626	053045	041505	020124					MESSAGES	
3699	017634	042101	037522	100					WHERE: . ASCII	'%VECT ADR?'
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	' ?'
3720	020010	100								
3721	020011	045	051120	021507					MO: . ASCII	'%PRG#?'
3722	020016	100								
3723	020017	045	040077						M1: . ASCII	'%?=?'
3724	020022	042445	042116	040040					M2: . ASCII	'%END ?'
3725	020030	051445	036522	037460					M3: . ASCII	'%SR=0? GO. ?'
3726	020036	043440	027117	100						
3727	020043	045	042114	041440					M4: . ASCII	'%LD CHAR IN SR0-7; DLY IN SR8-15?'
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 TSTS?'
3734	020110	020103	051524	051524						
3735	020116	100								
3736	020117	045	046530	052111					PRG1M: . ASCII	'%XMITTER LOOP?'
3737	020124	042524	020122	047514						
3738	020132	050117	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]

BRK5	014312	2916	2920#											
BRK6	014330	2927	2931#											
BRK7	014346	2938	2942#											
CARMSK	001572	699#	1114#	1334	1364	1603	2701	2731	3278					
CAT	001400	680#	681	1009	1030#	1128#	1256	1520	1521#	2009	2534	2648	2649#	2659
		2663	2689#	2720#	3483#	3515#	3626	3627#	3629	3630				
CC	= 177776	432#												
CHALT	= 104003	509#												
CHARS	016742	3608#	3609	3632										
CLKINT	001552	691#	1061#	1129#	1317#									
CLKLVL	001554	692#	1130#											
CNT	004574	1175#	1192#	1211#	3692									
CNTLU =	104015	519#	860	3616	3674									
CNTLUU	017546	729	3680#											
CNV DAT	002332	753	765	800#										
COUNT	002076	734#	1034#	1035#	1134#	1141#	1409#	1421	1531#	1535#				
CSR	001542	687#	883#	1031#	1081	1088	1132#	1137#	1144#	1248	1261#	1269#	1270	1305#
		1306	1308#	1310	1318#	1326#	1327	1351	1353	1355#	1380#	1382	1384#	1385
		1387#	1417#	1439	1442	1544#	1547	1549	1552	1556#	1557	1559#	1682	1698#
		1699	1703#	1704	1717#	1718	1722#	1723	1736#	1737	1741#	1742	1755#	1756
		1760#	1761	1774#	1775	1779#	1780	1793#	1794	1798#	1799	1812#	1813	1817#
		1818	1831#	1832	1836#	1837	1851#	1854	1858#	1859#	1860	2057#	2058#	2059
		2062	2065#	2069	2093	2117	2130#	2131#	2149#	2157#	2158#	2542#	2553	2557#
		2558#	2567#	2593	2617	2619#	2620	2652#	2654	2656#	2657	2690#	2693	2695
		2697#	2721#	2724	2726	2728#	2770#	2775	2777#	2781	2797	2828#	2832#	2834
		2836#	2847	2849	3234#	3249#	3251	3254#	3258#	3328#	3330	3335	3337	3485#
		3488	3503	3520#	3529	3541	3561#	3578#	3581	3583#	3589#			
		704#	867	873	917#									
CURTST	002002	508#	1605											
DATCHK=	104002	1530#	3043	3054	3065	3076	3087	3098	3109	3120	3131	3142	3153	3164
DATTST	006312	3175	3186	3197	3208									
		3039	3043#											
DATO	014544	3050	3054#											
DAT1	014562	1531#												
DAT1AA	006316	3127	3131#											
DAT10	014724	3138	3142#											
DAT11	014742	3149	3153#											
DAT12	014760	3160	3164#											
DAT13	014776	3171	317											

[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											
PSM = 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								
RCVDAT 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#						
	1298#	2350	2361	2372	2383	2394	2405	2416	2427	2438	2449	2460	2471		
	2482	2493	2504	2515											
RCVTST 005034	2346	2350#													
RCV0 012014	2357	2361#													
RCV1 012032	2434	2438#													
RCV10 012174	2445	2449#													
RCV11 012212	2456	2460#													
RCV12 012230	2467	2471#													
RCV13 012246	2478	2482#													
RCV14 012264	2489	2493#													
RCV15 012302	2500	2504#													
RCV16 012320	2511	2515#													
RCV17 012336	2368	2372#													
RCV2 012050	2379	2383#													
RCV3 012066	2390	2394#													
RCV4 012104	2401	2405#													
RCV5 012122	2412	2416#													
RCV6 012140	2423	2427#													

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#	1857
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#	

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					
		889#	897																
SCOPEA	002736	888	890#																
SCOPEB	002742	868	891	895	898#														
SCOPED	002776	708#	886	916*															
SCOPTH	002012	576#	818	825	832	848	1011*	1012	1219										
SPBOT	001200	511#	1853	1935	2037	3528													
SRESET=	104005	852#	1668																
SRSET	002546	721	961#	963*															
SRSETT	003232	669	818#																
START	002360	517#	819																
SUSMR =	104013	727	3649#																
SUSMRR	017412	932*	940	942#															
SVRPC	003166	933*	939	943#															
SVRPSH	003170	677#	767	782	791	852	865	869	887	890	898	1881	1889	1915					
SHR	001202	1956	1964	1991	2665	3563	365	3621	3652	3656*	3665	3680	3694*						
SHREG	000176	666#	852	3656	3665	3680	3685												
TIMEA	004272	1134#	1143																
TIMEB	004326	1136	1141#																
TIMEC	004336	1129	1144#																
TIMER	004216	1126#	1140	2532															
TIME1	004366	1126*	1135*	1145	1152#	1420	1430	2572	2597	2629	3241	3322	3517	3518					
TIME14	004370	1145*	1146*	1148*	1153#	2544	2588	2794											
TKCSR	002066	730#	1176																
TKDSR	002070	731#	1178	3670															
TMP1	017604	3669*	3670*	3671*	3672	3691#	3694												
TPCSR	002072	732#	784	787	996	1195	1198	1201											
TPDSR	002074	733#	786*	995*	1180*	1197*	1200*												
TTDAT	001562	695#	1574*	1575*	1576	1579													
TUMTAG	001600	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	1069	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	2413*
		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*					
		693*	1063*	1260*	2129*	2148*	2156*	2541*						
		694*	2155											
XMTINT	001556	1234*	2172	2183	2194	2205	2216	2227	2238	2249	2260	2271	2282	2293
XMTLVL	001560	2304	2315	2326	2337									
XMTTST	004622	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													
SSWREG	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