

NOV 1976
digital
MADE IN USA

.REM :

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDMB-B-D

PRODUCT NAME: DM11 DATA TESTS

DATE RELEASED: MAY, 1976

MAINTAINER: DIAGNOSTIC GROUP

COPYRIGHT 1972, 1976 BY DIGITAL EQUIPMENT CORPORATION
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.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A
LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH
THE TERMS OF SUCH LICENSE.
DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY
FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT
THAT IS NOT SUPPLIED BY DIGITAL.

1. ABSTRACT

TWO SEPARATE DIAGNOSTIC PROGRAMS ARE PROVIDED FOR TESTING THE DM11 (ASYNCHRONOUS DATA MULTIPLEXER), MAINDEC-11-DZDMA (DM11 LOGIC TESTS), AND MAINDEC-11-DZDMB (DM11 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 MULTIPLE LINE DATA TESTS. THE AVAILABLE TESTS ARE:

PRG0 - DATA TESTS
PRG1 - DATA TEST (ALL LINES SIMULTANEOUSLY)
PRG2 - TRANSMIT TO TERMINALS
PRG3 - ECHO RECEIVED DATA

2. REQUIREMENTS

2.1 EQUIPMENT

- A. PDP 11 FAMILY PROCESSOR
- B. DM11
- C. JUMPERS CONNECTING 16 TRANSMITTERS TO THEIR RESPECTIVE RECEIVERS.
- D. TERMINALS (IF AVAILABLE)
- E. DM11 DISTRIBUTION PANEL

2.2 STORAGE

THIS PROGRAM USES ALL OF CORE (4K) EXCEPT THAT AREA RESERVED FOR THE LOADERS.

2.3 PREREQUISITE PROGRAMS
MAINDEC-11-DZDMA (DM11 LOGIC TESTS)

3. LOADING PROCEDURE

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.
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 #(9)?

RESPONSE: THE DM11 UNIT NUMBER CORRESPONDS TO THE
ADDRESS TO WHICH THE CLOCK 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. PRG #

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

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.

SWITCH SETTINGS

THE FOLLOWING SWITCH SETTINGS APPLY TO PROGRAM #0.

```

SR  8-6  ROUTINE TO BE RUN (IF ENABLED BY SR-9)
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.

169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202

5.0 PROGRAM DESCRIPTIONS

5.1 PRG0 - LOGIC TESTS

PRG0 CONSISTS OF 34(8) INDEPENDENT ROUTINES WHICH TRANSMIT VARIOUS DATA PATTERNS ON ALL LINES WITH A DECREASING DELAY BEFORE STARTING SUCCESSIVE LINES. THE DATA IS CHECKED WHEN ALL TRANSMITTERS HAVE COMPLETED TRANSMITTING. IF A DATA ERROR OCCURS THE ERROR TYPEOUT WILL SHOW THE DATA FAILURE AND THE LINE NUMBER.

5.2 PRG1 - DATA TEST (ALL LINES SIMULTANEOUSLY)

PROGRAM 1 TRANSMITS ' A QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 1234567890' ON ALL LINES SIMULTANEOUSLY. WHEN ALL LINES HAVE FINISHED RECEIVED DATA IS VERIFIED. AN ERROR TYPEOUT IS AS IN PRG0.

5.3 PRG2 - TRANSMIT TO TERMINALS

PROGRAM 2 IS THE SAME AS PROGRAM 1 EXCEPT THAT THE RECEIVED DATA IS NOT CHECKED.

5.4 PRG3 - ECHO RECEIVED DATA

NOTE: THIS PROGRAM MAY ONLY BE RUN IF USING AN ASR 33 NOT MODIFIED BY DEC.

PROGRAM 3 ECHOES BACK DATA RECEIVED FROM A TERMINAL.

NOTE: PROGRAM 3 SHOULD BE RUN AND DATA TYPED AT ALL AVAILABLE DM11 TERMINALS. IT IS THE ONLY TEST THAT INSURE CORRECT OPERATION OF THE DM11 DISTRIBUTION PANEL LOGIC. IF THE TERMINALS ARE ASR-33 WITH A PAPER TAPE READER/PUNCH I SUGGEST THAT INDIVIDUAL TAPES BE MADE UP FOR EACH LINE. THIS CAN BE DONE BY RUNNING PROGRAM 2 WITH THE PUNCH TURNED ON. PROGRAM 2 WILL THEN PUNCH A TAPE ON EACH TERMINAL WITH THE LINE NUMBER IDENTIFIER AT THE BEGINNING OF EACH TAPE. PROGRAM 3 CAN BE RUN WITH THESE TAPES IN THE PAPER TAPE READERS.

%

```
.TITLE MAINDEC-11-DZDMB DM11 DATA TESTS
.NLIST MC,MD
.LIST ME
.ENABLE ABS,AMA
```

```
:DM11 DATA TESTS DIAGNOSTIC MAINDEC-11-DZDMB-B (FORMERLY D9GB)
:COPYRIGHT 1971, 1976 DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
:PRG0- DATA TESTS
:PRG1- DATA TEST (ALL LINES SIMULTANEOUSLY)
:PRG2- TRANSMIT TO TERMINALS
:PRG3- ECHO RECEIVED DATA
```

```
:STANDARD SR SWITCH OPTIONS [SWITCH SET TO A 1 (UP)]
:SR15- HALT ON ERROR
:SR14- SCOPE.
:SR13- INHIBIT PRINTOUT
:SR12- INHIBIT TRACE
:SR11- INHIBIT ITERATION.
:SR9- LOOP ROUTINE.
:SR6 THROUGH SR0 - NUMBER OF ROUTINE TO BE LOOPED.
```

```
:EQUATE STATEMENTS
CC=177776
PSW=177776
ERRVEC=4
NOP=240
OPEN=0
MANUAL=BIT15
BIT15=100000
BIT14=40000
BIT13=20000
BIT12=10000
BIT11=4000
BIT10=2000
BIT9=1000
BIT8=400
BIT7=200
BIT6=100
BIT5=40
BIT4=20
BIT3=10
BIT2=4
BIT1=2
BIT0=1
POPSP=5726
POPSP2=022626
PRTY7=340
PRTY6=300
PRTY5=240
PRTY4=200
PRTY3=140
PRTY2=100
PRTY1=40
```

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

```
177776
177776
000004
000240
000000
100000
!00000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001
005726
022626
000340
000300
000240
000200
000140
000100
000040
```


259	000000	PRTY0=0	
260	000000	R0=%0	
261	000001	R1=%1	
262	000002	R2=%2	
263	000003	R3=%3	
264	000004	R4=%4	
265	000005	R5=%5	
266	000006	SP=%6	
267	000007	PC=%7	
268		;EMT CALLS	
269	104000	TYPE=EMT+0	
270	104001	TYPES=EMT+1	
271	104002	STALL=EMT+2	
272	104003	ERROR=EMT+3	
273	104004	DATCHK=EMT+4	
274	104006	STRXV=EMT+6	
275	104007	STTXV=EMT+7	
276	104010	EHALT=EMT+10	
277	104012	SCOPE=EMT+12	
278	104013	SAVREG=EMT+13	
279	104014	RSTREG=EMT+14	
280	104015	ERROR1=EMT+15	
281	104016	SUSWR=EMT+16	
282	104017	KBDIN=EMT+17	
283	104020	CNTLU=EMT+20	
284			
285	000007	BELL=007	
286	177777	ATLAST=-1	
287	125252	ALTO=125252	;ALTERNATING 0'S PATTERN
288	052525	ALT1=052525	;ALTERNATING 1'S PATTERN
289	000000	Y=0	
290	177777	X=-1	
291	000000	A=0	
292	000000	. = 0	
293	000000	HALT	
294	000002	HALT	
295	000004	.+2	;SP OVERFLOW, BUS ERROR TRAP
296	000006	HALT	
297	000010	.+2	;RESERVED INSTRUCTION TRAP
298	000012	HALT	
299	000014	.+2	;TRACE TRAP
300	000016	HALT	
301	000020	.+2	;TRAP TO CALL IOX
302	000022	HALT	
303	000024	.+2	;POWER FAIL TRAP
304	000026	HALT	
305	000030	EMTINT	
306	000032	PRTY7	
307	000034	.+2	
308	000036	HALT	
309	000040	.+2	
310	000042	HALT	;TRAPPED TO PREVIOUS ADDRESS.
311	000044	.+2	
312	000046	HALT	;TRAPPED TO PREVIOUS ADDRESS.
313	000050	.+2	
314	000052	HALT	;TRAPPED TO PREVIOUS ADDRESS.

315	000054	000056	.+2	
316	000056	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
317	000060	000062	.+2	
318	000062	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
319	000064	000056	.+2	
320	000066	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
321	000070	000072	.+2	
322	000072	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
323	000074	000076	.+2	
324	000076	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
325	000100	000102	.+2	
326	000102	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
327	000104	000106	.+2	
328	000106	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
329	000110	000112	.+2	
330	000112	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
331	000114	000116	.+2	
332	000116	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
333	000120	000122	.+2	
334	000122	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
335	000124	000126	.+2	
336	000126	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
337	000130	000132	.+2	
338	000132	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
339	000134	000136	.+2	
340	000136	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
341	000140	000142	.+2	
342	000142	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
343	000144	000146	.+2	
344	000146	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
345	000150	000152	.+2	
346	000152	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
347	000154	000156	.+2	
348	000156	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
349	000160	000162	.+2	
350	000162	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
351	000164	000166	.+2	
352	000166	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
353	000170	000172	.+2	
354	000172	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
355	000174	000176	.+2	
356	000176	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
357	000200	000202	.+2	
358	000202	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
359	000204	000206	.+2	
360	000206	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
361	000210	000212	.+2	
362	000212	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
363	000214	000216	.+2	
364	000216	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
365	000220	000222	.+2	
366	000222	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
367	000224	000226	.+2	
368	000226	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.
369	000230	000232	.+2	
370	000232	000050	HALT	; TRAPPED TO PREVIOUS ADDRESS.

371	000234	000236	.+2	
372	000236	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
373	000240	000242	.+2	
374	000242	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
375	000244	000246	.+2	
376	000246	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
377	000250	000252	.+2	
378	000252	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
379	000254	000256	.+2	
380	000256	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
381	000260	000262	.+2	
382	000262	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
383	000264	000266	.+2	
384	000266	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
385	000270	000272	.+2	
386	000272	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
387	000274	000276	.+2	
388	000276	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
389	000300	000302	.+2	
390	000302	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
391	000304	000306	.+2	
392	000306	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
393	000310	000312	.+2	
394	000312	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
395	000314	000316	.+2	
396	000316	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
397	000320	000322	.+2	
398	000322	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
399	000324	000326	.+2	
400	000326	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
401	000330	000332	.+2	
402	000332	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
403	000334	000336	.+2	
404	000336	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
405	000340	000342	.+2	
406	000342	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
407	000344	000346	.+2	
408	000346	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
409	000350	000352	.+2	
410	000352	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
411	000354	000356	.+2	
412	000356	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
413	000360	000362	.+2	
414	000362	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
415	000364	000366	.+2	
416	000366	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
417	000370	000372	.+2	
418	000372	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
419	000374	000376	.+2	
420	000376	000000	HALT	; TRAPPED TO PREVIOUS ADDRESS.
421				

K01

```

422      000046 000046      =46      ;ACT11 HOOKS
423      000046 002452      $ENDAD
424      000052 000052      =52
425      000052 020000      020000
426
427
428      000174 000174      =174
429      000174 000000      DISPREG:0
430      000176 000000      SWREG: 0
431
432      000200 000200      =200
433      000200 000137 002076      JMP      @#START      ;GO TO START OF DIAGNOSTIC.
434      000204 000137 002114      JMP      @#RSTAT1      ;GO GET PROGRAM # & RESTART PROGRAM
435      000210 000137 002154      JMP      @#RSTAT2      ;USING PREVIOUS DM11 PARAMETERS
436      000210 000137 002154      JMP      @#RSTAT2      ;RESTART PREVIOUS PROGRAM USING
437      000210 000137 002154      JMP      @#RSTAT2      ;PREVIOUS DM11 PARAMETERS
438
439      001100 001100      =1100
440
441      001100 000000      SPBOT: 0
442      001102 177570      SWR: 177570
443      001104 177570      DISPLAY:177570
444      001106 000000      CAT: OPEN
445      001146 000000      =CAT+32.
446      001146 000000      WCT: OPEN
447      001206 000000      =WCT+32.
448      001206 000000      BAT: OPEN
449      001246 000000      =BAT+32.
450      001246 000000      VAC: OPEN
451      001250 175000      CSR: 175000
452      001252 175002      BAR: 175002
453      001254 175004      BKCSR: 175004
454      001256 175006      BASREG: 175006
455      001260 000000      CLKINT: OPEN
456      001262 000240      CLKLVL: PRTY5
457      001264 000000      XMTINT: OPEN
458      001266 000240      XMTLVL: PRTY5
459      001270 000000      BARIM: OPEN
460      001272 000000      TTDAT: OPEN
461      001274 000000      LINBIT: OPEN
462      001276 000000      BARDAT: OPEN
463      001300 000000      TTPTR: OPEN
464      001306 001306      =VAC+32.
465      001306 000000      TUMTAB: OPEN
466      001506 001506      =TUMTAB+128.
467      001506 000060      TKVTR: 60
468      001510 000200      TKLVL: PRTY4
469      001512 000064      TPVTR: 64
470      001514 000200      TPLVL: PRTY4
471      001516 000000      KSTART: OPEN
472      001520 000000      CURTST: OPEN
473      001522 000000      RTNNO: OPEN
474      001524 000000      NXTST: OPEN
475      001526 000000      ICTR: OPEN
476      001530 000000      SCOPTR: OPEN
477      001532 177774      PRGLIM: -4

```

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

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

;STARTING ADDRESS OF
;TUMBLE TABLE
;LSR INTERRUPT VECTOR
;LSR PRIORITY LEVEL
;LSP INTERRUPT VECTOR
;LSP PRIORITY LEVEL
;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.

478 001534 005120
479 001536 006100
480 001540 006264
481 001542 006274
482 001544 005142
483 001546 006116
484 001550 006270
485 001552 006312
486 001554 002650
487 001556 000000
488 001560 000000
489 001562 001726
490 001564 001660
491 001566 000000
492 001570 000000
493 001572 000000
494 001574 000000
495 001576 000000
496 001600 002322
497 001602 002546
498 001604 002606
499 001606 001744
500 001610 003652
501 001612 003526
502 001614 003572
503
504 001616 000000
505 001620 177560
506 001622 177562
507 001624 177564
508 001626 177566
509 001630 000000
510 001632 000000
511 001634 000000
512 001636 000000
513 001640 000000
514 001642 000000
515 001644 000000
516 001646 000000
517 001650 000000

PRGTAB: PRG0
PRG1
PRG2
PRG3
RSTART: PRGOR
PRG1R
PRG2R
PRG3R
EMTTAB: TYP
OPEN
OPEN
ERR
DTCHK
OPEN
OPEN
OPEN
OPEN
OPEN
OPEN
ESCOPE
SAVRG
RSTRG
ERR1
SUSWRR
KBDINTT
CNTLUU

SRT: OPEN
TKCSR: 177560
TKDBR: 177562
TPCSR: 177564
TPDBR: 177566
RCVDAT: OPEN
XMTDAT: OPEN
CARMSK: OPEN
TEMP: OPEN
PCADD: OPEN
APCADD: OPEN
PRVCNT: OPEN
LINE: OPEN
LINBUF: OPEN

;PRG0 START ADDRESS
;PRG1 START ADDRESS
;PRG2 START ADDRESS
;PRG3 START ADDRESS
;PRG0 RESTART ADDRESS
;PRG1 " "
;PRG2 " "
;PRG3 " "
;POINTER TO TYPEOUT ROUTINE
;POINTER TO CHAINED MESSAGES ROUTINE
;POINTER TO RANDOM STALL ROUTINE
;POINTER TO ERROR ROUTINE

Address	Instruction	Comment
001652	INCRTN: TYPE	
001654	M1	
001656	RTS %7	;TYPE INCORRECT ROUTINE SELECTED. ;EXIT.
		;DATA CHECK ROUTINE.
001660	DTCHK: CMPB RCVDAT,XMTDAT	;COMPARE EXPECTED AND RECEIVED
001666	BEQ 1\$	;CHARS. BRANCH IF SAME.
001670	JSR 7,CNVDAT	;CONVERT RCVDAT & XMTDAT TO ASCII
001674	BIT #BIT13,JSWR	;ERROR TYPEOUT DESIRED?
001702	BNE 1\$	;BRANCH IF NO TYPEOUT DESIRED
001704	JSR 5,2#OACNV	;CONVERT LINE
001710	LINE	;NUMBER
001712	ALINE	;TO ASCII
001714	2	
001716	ERROR1	
001720	TYPE	;TYPE LINE # AS PART
001722	LINEM	;OF ERROR MESSAGE
001724	1\$: RTI	;EXIT.
		;ERROR SERVICE ROUTINE CALLED BY TRAP (HLT)
001726	ERR: MOV #402,ERRB	;MOV BR .+6 TO ERRB
001734	MOV 2#PCADD,2#APCADD	;GET PC WHERE ERROR OCCURRED
001742	BR ERRA	
001744	ERR1: MOV #240,ERRB	;MOVE NOP TO ERRB
001752	MOV 2#PCADD,2#APCADD	;GET PC WHERE ERROR OCCURRED
001760	JSR 7,2#CNVDAT	;CONVERT RCVDAT & XMTDAT TO ASCII
001764	ERRA: KBDIN	;CHECK FOR IG
001766	BIT #BIT13,JSWR	;ERROR PRINTOUT DESIRED
001774	BNE ERRC	;BRANCH IF NO PRINTOUT
001776	JSR 5,2#OACNV	;CONVERT
002002	APCADD	;DATA
002004	APC	;TO
002006	6	;ASCII
002010	JSR 5,2#OACNV	;FOR
002014	RTNNO	;PRINTOUT
002016	ATNUMB	
002020	3	
002022	TYPE	;TYPE ERROR
002024	EMO	;MESSAGE
002026	ERRB: OPEN	;NOP IF ERROR1, BR .+6 IF ERROR
002030	TYPE	;TYPE ANOTHER MESSAGE
002032	ERDAT	;IF ERROR 1
002034	ERRC: TST JSWR	;HALT ON ERROR
002040	BPL 1\$	;GO TO EXIT IF NO HALT ON ERROR
002042	HALT	;HALT
002044	1\$: KBDIN	;CHECK FOR IG
002046	RTI	;RETURN
		;SUBROUTINE TO CONVERT RCVDAT AND XMTDAT TO ASCII AND PLACE
		;IN MESSAGE.
002050	CNVDAT: JSR 5,OACNV	
002054	XMTDAT	
002056	AASB	
002060	6	
002062	JSR 5,OACNV	

NO1

MAINDEC-11-DZOMB DM11 DATA TESTS
DZOMBB.SRC

MACY11 27(732) 21-SEP-76 15:45 PAGE 13

574 002066 001630
575 002070 012475
576 002072 000006
577 002074 000207
578
579

RCVDAT
AWAS
6
RTS

7

;EXIT

```

00000000 002076 012706 001100 START: MOV #SPBOT,%6 ;INITIALIZE STACK
00000001 002102 104016 SUSWR ;CHECK FOR HARDWARE SWITCH REGISTER
00000002 002104 004737 002764 JSR 7,%DMPAR ;GET DM11 PARAMETERS
00000003 002110 004737 003312 JSR 7,%OVLAY ;PUT HALT...+2 IN VECTOR AREA
00000004 002114 012706 001100 RSTAT1: MOV #SPBOT,%6 ;INITIALIZE STACK
00000005 002120 104000 TYPE
00000006 002122 012520 MO
00000007 002124 004537 003346 JSR 5,RECD ;GET THE PRGNUM &
00000008 002130 000000 PRGNUM: 0 ;PUT IT HERE
00000009 002132 043737 001532 002130 BIC PRGLIM,PRGNUM ;MASK OFF UNUSED BITS
00000010 002140 006337 002130 ASL PRGNUM ;SHIFT PROGRAM #
00000011 002144 012700 002130 MOV PRGNUM,%0 ;GET PROGRAM #
00000012 002150 000170 001534 JMP @PRGTAB(0) ;GO START PROGRAM
00000013 002154 012706 001100 RSTATE2: MOV #SPBOT,%6 ;INITIALIZE STACK
00000014 002160 013700 002130 MOV PRGNUM,%0 ;GET PROGRAM #
00000015 002164 000170 001544 JMP @RSTART(0) ;GO RESTART PROGRAM
00000016 002170 022737 000176 001102 SRSET: CMP #SWREG,SWR
00000017 002176 001404 BEQ 1$
00000018 002200 104000 TYPE ;TYPE OPTIONS MESSAGE
00000019 002202 012570 M3
00000020 002204 000000 HALT
00000021 002206 000401 BR ;WAIT FOR USER TO SET OPTIONS
00000022 002210 104020 1$: CNTRL GETRDY
00000023 002212 013737 001516 001524 GETRDY: MOV KSTART,NXTST ;ADDR OF 1ST ROUTINE TO NXTST
00000024 002220 012737 000006 000004 GTRDYX: MOV #6,%ERRVEC ;RESET ERROR TRAP.
00000025 002226 005037 177776 CLR PSW
00000026 002232 012706 001100 MOV #SPBOT,%6 ;SET BOTTOM OF STACK.
00000027 002236 000005 RESET ;ISSUE RESET.
00000028 002240 004737 002462 GTRDYA: JSR %7,FORMD ;ROLL FORWARD TO "NEXT" ROUTINE.
00000029 002244 032777 001000 176630 BIT #BIT9,SSWR ;CHECK SELECT ROUTINE SWITCH
00000030 002252 001003 BNE GTRDYC ;BRANCH IF SELECT ROUTINE SWITCH IS SET.
00000031 002254 000177 177240 JMP @COURTST ;GO RUN CURRENT ROUTINE.
00000032 002260 000452 BR SCOPED ;NO GO. MANUAL RTN BYPASSED.
00000033 002262 017700 176614 GTRDYC: MOV %SWR,%0 ;(SR) TO R0
00000034 002266 042700 177600 BIC #177600,%0 ;MASK UNDESIED BITS
00000035 002272 123700 001522 CMPB RTNNO,%0 ;COMPARE RTNNO TO (R0)
00000036 002276 001002 BNE GTRDYD ;BRANCH IF ROUTINE NOT FOUND YET.
00000037 002300 000177 177214 JMP @COURTST ;GO RUN ROUTINE.
00000038 002304 022737 177777 001524 GTRDYD: CMP #-1,NXTST ;NO. CHECK FOR LAST ROUTINE.
00000039 002312 001352 BNE GTRDYA ;BRANCH IF NOT LAST ROUTINE.
00000040 002314 004737 001652 JSR %7,INCRTN ;YES. INCORRECT ROUTINE SELECTED.
00000041 002320 000734 BR GETRDY ;START OVER.

;SCOPE ROUTINE (CALLED BY EMT INST.)
00000042 002322 000240 ESCOPE: NOP
00000043 002324 005077 176720 CLR %CSR ;INITIALIZE
00000044 002330 005077 176720 CLR %BKCSR ;THE
00000045 002334 005077 176712 CLR %BAR ;DM11
00000046 002340 104017 KBDIN
00000047 002342 012777 001106 176706 MOV #CAT,%BASREG
00000048 002350 032777 040000 176524 BIT #BIT14,%SWR ;CHECK FOR SCOPE OPTION.
00000049 002356 001403 BEQ SCOPEB ;BRANCH IF SCOPE SW NOT SET.
00000050 002360 013716 001530 SCOPEA: MOV SCOPEB ;SET UP TO RETURN TO ROUTINE.
00000051 002364 000002 RTI ;RETURN TO ROUTINE.
00000052 002366 032777 004000 176506 SCOPEB: BIT #BIT11,%SWR ;TEST INHIBIT ITERATION SWITCH

```


636	002374	001003		BNE	SCOPEC	:BRANCH IF INHIBIT ITERATION SW SET.
637	002376	005337	001526	DEC	ICTR	:DECREMENT ITERATION COUNT.
638	002402	001366		BNE	SCOPEA	:BRANCH IF COUNT NOT 0.
639	002404	022626		SCOPEC: POPSP2		:POP STACK TWICE
640	002406	032777	001000 176466	SCOPEC: BIT	%BIT9,2SWR	:CHECK SELECT ROUTINE SWITCH
641	002414	001276		BNE	GETRDY	:BRANCH IF SELECT RTN SW SET
642	002416	022737	177777 001524	CMP	%-1,NXTST	:LAST TEST
643	002424	001275		BNE	GTCDYX	:BRANCH IF NOT LAST TEST.
644	002426	104000		TYPE		:TYPE
645	002430	012543		M2		:PRGEND
646	002432	013702	000042	MOV	%42,%2	:CHECK DDP/ACT11 MONITOR HOOK
647	002436	001655		BEQ	GETRDY	
648	002440	000005		RESET		
649	002442	000240		NOP		
650	002444	000240		NOP		
651	002446	000240		NOP		
652	002450	000240		NOP		
653	002452	004712		SENDAD: JSR	7.(2)	:RETURN TO DDP/ACT11 MONITOR
654	002454	000240		NOP		
655	002456	000240		NOP		
656	002460	000240		NOP		
657	002462	013705	001524	FORWD: MOV	NXTST,%5	:ADDR OF NEXT ROUTINE TO RS.
658	002466	012537	001522	MOV	(5)+,RTNNO	:GET NEXT ROUTINE NUMBER.
659	002472	012537	001524	MOV	(5)+,NXTST	:GET ADDR OF NEXT "NEXT" ROUTINE.
660	002476	012537	001526	MOV	(5)+,ICTR	:GET ITERATION COUNT.
661	002502	012537	001530	MOV	(5)+,SCOCTR	:GET SCOPE LOOP ENTRY POINTER.
662	002506	010537	001520	MOV	%5,CURTST	:ADDR OF NOW CURRENT TEST TO CURTST.
663	002512	000207		RTS	%7	:EXIT FORWD SUBROUTINE.
664						
665						
666						
667	002514	011646		:EMT TRAP INTERPRETER		
668	002516	162716	000002	EMTINT: MOV	(6)+,(6)	:GET PC OF NEXT INSTRUCTION
669	002522	011637	001640	SUB	%2,(6)	:POINT SP TO PC OF EMT
670	002526	017616	000000	MOV	(6),PCADD	:GET PC OF EMT CALL
671	002532	105066	000001	MOV	%6,(6)	:GET EMT CALL
672	002536	006316		CLRB	1(6)	:STRIP EMT & SAVE IDENTIFIER
673	002540	062716	001554	ASL	(6)	:SHIFT IDENTIFIER LEFT
674	002544	013607		ADD	%EMTAB,(6)	
675				MOV	%6)+,%7	:GO TO PROPER EMT
676						
677	002546	012637	002602	:SAVE REGS 0 TO 4 SUBROUTINE.		
678	002552	012637	002604	SAVRG: MOV	(6)+,%1	:SAVE PC AND PSW.
679	002556	010446		MOV	(6)+,%2	
680	002560	010346		MOV	%4,-(6)	:SAVE REGS 0 - 4
681	002562	010246		MOV	%3,-(6)	:IN STACK.
682	002564	010146		MOV	%2,-(6)	
683	002566	010046		MOV	%1,-(6)	
684	002570	013746	002604	MOV	%0,-(6)	
685	002574	013746	002602	MOV	%5,-(6)	:RESTORE PC AND PSW.
686	002600	000002		MOV	%5,-(6)	
687	002602	000000		RTI		:EXIT.
688	002604	000000		1\$: OPEN		:CONTAINS SAVED PC
689				2\$: OPEN		:CONTAINS SAVED PSW
690						
691	002606	000240		:RESTORE REGS 0 TO 4 SUBROUTINE.		
				RSTRG: NOP		

002

```

692 002610 012637 002644      MOV      (6)+,1$      ;SAVE PC AND PSW.
693 002614 012637 002646      MOV      (6)+,2$
694 002620 012600              MOV      (6)+,%0      ;RESTORE REGS 0 - 4
695 002622 012601              MOV      (6)+,%1      ;FROM STACK.
696 002624 012602              MOV      (6)+,%2
697 002626 012603              MOV      (6)+,%3
698 002630 012604              MOV      (6)+,%4
699 002632 013746 002646      MOV      2$,-(6)      ;RESTORE PC AND PSW.
700 002636 013746 002644      MOV      1$,-(6)
701 002642 000002              RTI
702 002644 000000              1$: OPEN      ;CONTAINS SAVED PC
703 002646 000000              2$: OPEN      ;CONTAINS SAVED PSW
704
705      ;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER.
706      TYP: NOP
707 002652 011600              MOV      (SP),%0      ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS.
708 002654 062716 000002      ADD      #2,(SP)      ;SET UP EXIT.
709 002660 011000              MOV      3%0,%0      ;ADDRESS OF MESSAGE TO RD.
710 002662 112037 002762      1$: MOV      (0)+,5$      ;GET CHARACTER
711 002666 122737 000100 002762 CMP      #100,5$      ;CHECK FOR "3" CHARACTER
712 002674 001001              BNE      2$      ;BRANCH IF NOT "3".
713 002676 000002              RTI      ;TERMINATOR CHAR. DONE. EXIT.
714 002700 122737 000045 002762 2$: CMP      #45,5$      ;CHECK FOR "%".
715 002706 001412              BEQ      4$      ;BRANCH IF "%".
716 002710 004737 002716      JSR      %7,3$      ;TYPE CHAR IN 5$
717 002714 000762              BR      1$
718 002716 113777 002762 176702 3$: MOV      5$,JTPDBR      ;OUTPUT CHARACTER TO PRINTER
719 002724 105777 176674      TSTB      JTPCSR      ;WAIT FOR DONE FLAG.
720 002730 100375              BPL      -4
721 002732 000207              RTS      %7
722 002734 112737 000015 002762 4$: MOV      #15,5$      ;EXIT
723 002742 004737 002716      JSR      %7,3$      ;MOVE CARRIAGE RETURN CODE TO 5$
724 002746 112737 000012 002762 MOV      #12,5$      ;GO TYPE CHAR.
725 002754 004737 002716      JSR      %7,3$      ;MOVE LF CODE TO 5$.
726 002760 000740              BR      1$      ;GO TYPE CHAR.
727 002762 000000              5$: OPEN
728
729      ;SUBROUTINE TO GET DM11 PARAMETERS
730      ;VECTOR ADDRESS
731 002764 000240              DMPAR: NOP      ;BEGIN
732 002766 004737 003312      JSR      7,OVRLAY      ;PUT HALT...+2 IN VECTOR AREA
733 002772 104000              TYPE      ;ASK USER FOR RECEIVER INT. VECTOR
734 002774 012310              WHERE      ;OF UNIT UNDER TEST
735 002776 004537 003346      JSR      5,RECD      ;GET THE VECTOR &
736 003002 000000              VECTOR: 0      ;PUT IT HERE
737 003004 005737 003002      TST      VECTOR
738 003010 001003              BNE      1$
739 003012 012737 000300 003002 1$: MOV      #300,VECTOR      ;SET VECTOR = TO 0300
740 003020 023727 003002 000300 CMP      VECTOR,#300      ;IS VECTOR HIGHER OR
741 003026 103003              BHS      3$      ;EQUAL TO 0300
742 003030 104000              2$: TYPE      ;TYPE ...
743 003032 012540              MI
744 003034 000753              BR      DMPAR      ;ASK FOR ANOTHER VECTOR
745 003036 023727 003002 000770 3$: CMP      VECTOR,#770      ;IS VECTOR = TO OR
746 003044 101371              BHI      2$      ;LESS THAN 770
747 003046 032737 000007 003002 BIT      #7,VECTOR      ;LSB OF VECTOR MUST BE ALL 0'S

```



```

748 003054 001365      BNE 2$
749 003056 013737 003002 001260      MOV VECTOR,3*CLKINT
750 003064 062737 000004 003002      ADD #4,VECTOR
751 003072 013737 003002 001254      MOV VECTOR,3*XMTINT
752
753      ;UNIT NUMBER
754 003100 104000      DMPARB: TYPE
755 003102 012403      WHICH
756 003104 004537 003346      JSR 5,RECD      ;GET THE UNIT 3
757 003110 000000      UNIT: 0      ;PUT IT HERE
758 003112 023727 003110 000017      CMP UNIT,#17
759 003120 101403      BLOS 1$
760 003122 104000      TYPE
761 003124 012540      M1
762 003126 000764      BR DMPARB
763 003130 006337 003110      1$: ASL UNIT
764 003134 006337 003110      ASL UNIT
765 003140 006337 003110      ASL UNIT
766 003144 012702 000004      MOV #4,%2
767 003150 012701 001250      MOV #CSR,%1
768 003154 042711 000370      2$: BIC #370,(1)
769 003160 063721 003110      ADD UNIT,(1)+
770 003164 005302      DEC %2
771 003166 001372      BNE 2$
772
773      ;CALCULATE CHARACTER LENGTH
774 003170 012777 001106 176060      MOV #CAT,3BASREG
775 003176 005077 176046      CLR 3CSR
776 003202 012737 177777 006622      MOV #-1,OUTBUF      ;LOAD OUTBUF WITH CHAR TO BE TRANSMITTED
777 003210 012737 177777 001146      MOV #-1,WCT      ;SET UP TO TRANSMIT 1 CHAR
778 003216 012737 177777 001634      MOV #-1,3CARMSK      ;PRE SET THE CHARACTER MASK
779 003224 012737 006622 001106      MOV #OUTBUF,CAT      ;1 CHARACTER ON LINE 0
780 003232 012777 003270 176020      MOV #3$,3CLKINT      ;LOAD RECEIVER INTERRUPT
781 003240 012777 000340 176014      MOV #340,3CLKLVL      ;AND PRIORITY LEVEL
782 003246 005037 001306      CLR TUMTAB
783 003252 012777 000001 175772      MOV #1,3BAR      ;START TRANSMITTING
784 003260 012777 000105 175762      MOV #BIT6+BIT2+BIT0,3CSR      ;SET IE,MAINT AND GO BITS
785 003266 000001      WAIT      ;WAIT FOR RECEIVER INTERRUPT
786 003270 005077 175754      3$: CLR 3CSR
787 003274 143737 001306 001634      BICB TUMTAB,CARMSK      ;LOAD CHARACTER LENGTH MASK
788 003302 005037 177776      CLR PSW      ;RESTORE PROCEESSOR TO PRIORITY 0
789 003306 022626      POPSP2      ;RESTORE THE STACK POINTER
790 003310 000207      RTS 7      ;EXIT PARAMETERS ROUTINE
791
792      ;ROUTINE TO LOAD TRAP/INTERRUPT VECTOR AREA WITH HALT,.,+2. HALTS PROGRAM
793      ;AT ADDRESS OF TRAP/INTERRUPT VECTOR +2.
794 003312 012701 000300      OVRLAY: MOV #300,%1
795 003316 012702 000302      MOV #302,%2
796 003322 010221      1$: MOV %2,(1)+
797 003324 005021      CLR (1)+
798 003326 020227 000776      CMP %2,#776
799 003332 001403      BEQ 2$
800 003334 062702 000004      ADD #4,%2
801 003340 000770      BR 1$
802 003342 000240      2$: NOP
803 003344 000207      RTS 7      ;EXIT

```

304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359

003346 010046
003350 005015
003352 012737 000007 003524
003353 105777 176234
003364 100375
003366 117700 176230
003372 142700 000200
003376 110077 176224
003402 122700 000025
003406 001443
003410 122700 000015
003414 001415
003416 142700 000060
003422 132700 000110
003426 001031
003430 006315
003432 006315
003434 006315
003436 150015
003440 005337 003524
003444 001422
003446 000744
003450 105777 176150
003454 100375
003456 012777 000012 176142
003464 105777 176134
003470 100375
003472 005077 176130
003476 105777 176122
003502 100375
003504 005725
003506 012600
003510 000205
003512 104000
003514 012540
003516 104000
003520 012400
003522 000712
003524 000000

; 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 R0, -(SP)
1\$: CLR (5) ; CLEAR OLD DATA
MOV #7, CNT ; SET CHAR COUNT
2\$: TSTB @TKCSR ; WAIT FOR CHAR
BPL 2\$
MOVB @TKDBR, R0
BICB #20C, R0 ; STRIP OFF PARITY
MOVB R0, @TPDBR ; ECHO CHARACTER
CMPB #25, R0 ; IS IT A 'U'
BEQ 5\$; BRANCH IF YES
CMPB #15, R0 ; IS IT A '<CR>'
BEQ 6\$; BRANCH IF YES
BICB #60, R0
BITB #110, R0 ; CHECK FOR 0-7 (8)
BNE 7\$; BRANCH IF NOT
ASL (5)
ASL (5)
ASL (5) ; SHIFT DATA
BISB R0, (5) ; INSET NEW CHAR
DEC CNT
BEQ 7\$; ONLY 6 CHAR'S PLEASE
BR 2\$; NEXT CHARACTER
6\$: TSTB @TPCSR
BPL 6\$; WAIT FOR READY
MOV #12, @TPDBR ; TYPE '<LF>'
8\$: TSTB @TPCSR
BPL 8\$; WAIT FOR READY
CLR @TPDBR ; NEXT CHARACTER
9\$: TSTB @TPCSR
BPL 9\$; WAIT FOR READY
TST (R5)+ ; ADJUST R5
MOV (SP)+, R0 ; RESTORE R0
RTS R5
7\$: TYPE M1
5\$: TYPE \$CTLU
BR 1\$; START OVER
CNT: 0

; ROUTINE TO CHECK FOR 'G' BEING TYPED

KBDINTT: CMP #SWREG, SWR
BNE 1\$
CLR TMP1 ; CLEAR TEMP AREA
MOVB @TKDBR, TMP1 ; FETCH THE BUFFER


```

860 003550 142737 000200 003630      BICB      #200,TMP1      :STRIP OFF PARITY
861 003556 122737 000007 003630      CMPB      #7,TMP1      :WAS IT 1G
862 003564 001001      BNE          1$          :NOP
863 003566 104020      CNTLU      :GO CHANGE IT
864 003570 000002      1$: RTI          :EXIT
865
866
867      ;ROUTINE TO CHANGE CONTENTS OF SWREG(LOC 176)
868
869 003572 022737 000176 001102 CNTLUU: CMP      #SWREG,SWR
870 003600 001023      BNE      FAJAG
871 003602 104000      TYPE
872 003604 012351      $SWREG
873 003606 004537 004352 JSR      R5,OACNV      :CONVERT TO ASCII
874 003612 000176      SWREG
875 003614 012360      $VALUE
876 003616 000006      6
877 003620 104000      TYPE
878 003622 012360      $VALUE
879 003624 004537 003346 JSR      5,RECD      :GET THE TMP1 &
880 003630 000000      TMP1: 0          :PUT IT HERE
881 003632 022737 000007 003524      CMP      #7,CNT
882 003640 001403      BNE      FAJAG
883 003642 013777 003630 175232      MOV      TMP1,$SWR      :CHANGE CONTENTS OF SWREG
884 003650 000002      FAJAG: RTI
885
886
887 003652 013746 000006      SUSWRR: MOV      @#6,-(SP)      :SAVE VECTORS
888 003656 013746 000004      MOV      @#4,-(SP)
889 003662 012737 003702 000004      MOV      #1$,@#4      :SET UP FOR TIMEOUT
890 003670 022777 177777 175204      CMP      #-1,$SWR      :REFERENCE HARDWARE SWITCH REGISTER
891 003676 001402      BEQ      2$
892 003700 000407      BR      3$
893 003702 022626      1$: CMP      (SP)+,(SP)+      :ADJUST STACK
894 003704 012737 000176 001102 2$: MOV      #SWREG,SWR      :POINT TO SOFTWARE SWITCH REG
895 003712 012737 000174 001104      MOV      #DISPREG,DISPLAY      :POINT TO SOFT DISPLAY REG
896 003720 012637 000004      3$: MOV      (SP)+,@#4      :RESTORE VECTORS
897 003724 012637 000006      MOV      (SP)+,@#6
898 003730 000002      RTI
899

```

```

900      :SUBROUTINE TO TRANSMIT ON ALL LINES WITH A DELAY BETWEEN TRANSMITTING
901      :ON SUCCESSIVE LINES. THE DELAY FOR THE TEST IS SUPPLIED BY THE
902      :CALLING JSR INSTRUCTION. DATA IS CHECKED AFTER ALL
903      :LINES HAVE FINISHED TRANSMITTING.
904
905      003732 000240      DLYXMT: NOP      ;BEGIN TEST
906      003734 012777 001106 175314      MOV      #CAT, @BASREG      ;SET UP BASE REGISTER
907      003742 004737 004264      JSR      7, @IDENT      ;TRANSMIT LINE # ON EACH LINE
908      003746 000240      NOP      ;NOP
909      003750 005077 175274      CLR      @CSR      ;GET MESSAGE ADDRESS
910      003754 012537 003764      MOV      (5)+, 10$      ;LOAD OUTPUT BUFFER
911      003760 004537 004440      JSR      5, @BMOVE      ;WITH DATA TO
912      003764 000000      10$: OPEN      ;BE TRANSMITTED
913      003766 006622      OUTBUF
914      003770 000100      64.
915      003772 005037 001306      CLR      @TUMTAB      ;CLEAR TUMBLE
916      003776 004537 004440      JSR      5, @BMOVE      ;TABLE (200
917      004002 001306      TUMTAB      ;BYTES)
918      004004 001307      TUMTAB+1
919      004006 000177      177
920      004010 004537 004440      JSR      5, @BMOVE      ;CLEAR CHARACTER COUNT TABLE
921      004014 001306      TUMTAB
922      004016 012126      CNTTAB
923      004020 000020      16.
924      004022 005037 006766      CLR      @LNOBUF      ;CLEAR ALL
925      004026 004537 004440      JSR      5, @BMOVE      ;LINE'S INPUT
926      004032 006766      LNOBUF      ;BUFFERS
927      004034 006767      LNOBUF+1      ; (16. BUFFERS OF 100. CHARS. EACH)
928      004036 003077      1599.
929      004040 022737 000006 002130      CMP      #6, PRGNUM
930      004046 001002      BNE      .+6
931      004050 000137 006322      JMP      PRG3A
932      004054 012504      MOV      (5)+, %4      ;GET # OF CHARACTERS TO TRANSMIT BEFORE
933      :TRANSMITTING ON NEXT LINE
934      004056 012737 001306 001300      MOV      @TUMTAB, @TTPTR      ;INITIALIZE TUMBLE TABLE POINTER
935      004064 013701 001260      MOV      @CLKINT, %1      ;GET RECEIVER VECTOR ADDRESS
936      004070 012721 004666      MOV      @RINT, (1)+      ;LOAD RECEIVER VECTOR
937      004074 013721 001262      MOV      @CLKLVL, (1)+      ;AND PRIORITY LEVEL
938      004100 012721 005060      MOV      @TINT, (1)+      ;LOAD TRANSMITTER VECTOR
939      004104 013721 001266      MOV      @XMTLVL, (1)+      ;AND PRIORITY LEVEL
940      004110 005737 002130      TST      PRGNUM      ;RUNNING PROGRAM 0?
941      004114 001402      BEQ      .+6
942      004116 000137 006126      JMP      PRG1A      ;RETURN TO PROGRAM 1 CODE
943      004122 012777 010101 175120      MOV      @BIT12+@BIT6+@BIT0, @CSR      ;SET IE & GO BITS
944      004130 012737 000001 001274      MOV      #1, @LINBIT
945      004136 005037 001646      CLR      @LINE
946      004142 013700 001646      1$: MOV      LINE, %0      ;LINE # X2 TO R0
947      004146 000240      NOP      ;NOP
948      004150 004537 004462      JSR      5, @XMITD      ;TRANSMIT 64 CHARACTERS
949      004154 177700      -64.      ;ON LINE # AS SPECIFIED IN ADDRESS LINE
950      004156 020460 001146      2$: CMP      %4, WCT(0)      ;WAIT FOR THE WORD COUNT TO DEC TO THE
951      004162 001375      BNE      2$      ;CORRECT VALUE BEFORE STARTING NEXT LINE
952      004164 062737 000002 001646      ADD      #2, LINE      ;FORM NEXT LINE NUMBER
953      004172 006337 001274      ASL      LINBIT      ;SHIFT LINE BIT
954      004176 103361      BCC      1$      ;START NEXT LINE
955      004200 005760 001146      3$: TST      WCT(0)      ;WAIT FOR LAST LINE TO FINISH

```



```

956 004204 001375      BNE      3$
957 004206 042777 177400 175034      BIC      #177400, @CSR      ;CLEAR ODD BYTE OF CSR
958 004214 062700 000001      ADD      #1, R0      ;WAIT FOR RECEIVER TO RECEIVE
959 004220 001375      BNE      31$      ;ALL TRANSMITTED DATA
960 004222 017737 175024 001630      MOV      @BAR, RCV DAT      ;GET AND TEST BAR CONTENTS
961 004230 001410      BEQ      4$      ;BRANCH IF IS CLEAR
962
963 004232 005037 001632      CLR      XMT DAT
964 004236 005077 175006      CLR      @CSR
965 004242 005077 175004      CLR      @BAR
966 004246 104015      ERROR1      ;ERROR! BAR DID NOT CLEAR IN SUFFICIENT TIME
967 004250 000403      BR      5$      ;EXIT TEST
968 004252 000240      4$:      NOP
969 004254 004737 004560      JSR      7, @CHK DAT      ;GO TEST DATA
970 004260 022626      5$:      CMP      (6)+, (6)+      ;RESET THE STACK
971 004262 104012      SCOPE      ;SCOPE
972
973
974      ;SUBROUTINE TO TRANSMIT ON EACH LINE ITS LINE NUMBER (CRLF YX CRLF).
975 004264 005037 001646      IDENT:  CLR      @LINE      ;GET LINE NUMBER 0
976 004270 012737 000001 001274      MOV      #1, @LIN BIT      ;GET LINE BIT
977 004276 013702 001646      1$:      MOV      LINE, %2
978 004302 016262 012146 001106      MOV      ID(2), CAT(2)      ;LOAD CAT
979 004310 012762 177772 001146      MOV      #-6, WCT(2)      ;LOAD WORD COUNT
980 004316 053777 001274 174726      BIS      LIN BIT, @BAR      ;SET BAR BIT
981 004324 062737 000002 001646      ADD      #2, LINE      ;FORM NEXT LINE NUMBER
982 004332 006337 001274      ASL      LIN BIT      ;FORM NEXT LINE BIT
983 004336 103357      BCC      1$      ;BRANCH IF NOT DONE
984 004340 005777 174706      2$:      TST      @BAR      ;WAIT FOR BAR TO CLEAR
985 004344 001375      BNE      2$
986 004346 000240      NOP
987 004350 000207      RTS      7      ;EXIT SUBROUTINE
988
989      ;OCTAL TO ASCII CONVERT ROUTINE
990 004352 104013      OACNV:  SAVREG
991 004354 013537 004436      MOV      @5)+, 2$      ;SAVE REGISTERS ON THE STACK
992 004360 012501      MOV      (5)+, %1      ;GET OCTAL VALUE.
993 004362 012502      MOV      (5)+, %2      ;GET DESTINATION ADDR.
994 004364 060201      ADD      %2, %1      ;GET CONVERT COUNT.
995 004366 013703 004436      1$:      MOV      2$, %3      ;DEVELOP ADDR TO STORE 1ST CHAR.
996 004372 042703 177770      BIC      #177770, %3      ;ISOLATE LEAST SIGNIFICANT DIGIT.
997 004376 062703 000060      ADD      #60, %3      ;CONVERT DIGIT TO ASCII.
998 004402 110341      MOVB      %3, -(1)      ;STORE ASCII CHARACTER.
999 004404 042737 000007 004436      BIC      #7, 2$
1000 004412 006037 004436      ROR      2$
1001 004416 006037 004436      ROR      2$
1002 004422 006037 004436      ROR      2$
1003 004426 005302      DEC      %2
1004 004430 001356      BNE      1$      ;DONE ALL DIGITS?
1005 004432 104014      RSTREG      ;BRANCH IF NOT DONE.
1006 004434 000205      RTS      %5      ;RESTORE THE REGISTERS
1007 004436 000000      2$:      OPEN
1008
1009
1010
1011      ;SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES.

```

MAINDEC-11-DZDMB DM11 DATA TESTS
DZDMBB.SRC

MACY11 27(732) 21-SEP-76 15:45 PAGE 22

1012	004440	104013
1013	004442	012501
1014	004444	012502
1015	004446	012503
1016	004450	112122
1017	004452	005303
1018	004454	001375
1019	004456	104014
1020	004460	000205

SMOVE:	SAVREG
	MOV (5)+,%1
	MOV (5)+,%2
	MOV (5)+,%3
1\$:	MOVB (1)+,(2)+
	DEC %3
	BNE 1\$
	RSTREG
	RTS %5

;SAVE REGS.
;GET"FROM"ADDRESS
;GET"TO"ADDRESS
;GET COUNT
;MOVE BYTE
;DECREMENT COUNT
;BRANCH IF NOT DONE.
;RESTORE REGS.
;DONE EXIT

K02

```

1021
1022
1023      ;SUBROUTINE TO TRANSMIT DATA.  SUBROUTINE CALLED BY
1024      ;JSR 5,XMITD
1025      XMITD:  NOP
1026             MOV      %0,-(SP)      ;SAVE RO ON THE STACK
1027             MOV      @#LINE,%0    ;GET LINE
1028             MOV      #OUTBUF,CAT(0) ;LOAD FIRST CHAR ADDRESS IN CAT
1029             MOV      (5)+,WCT(0)   ;LOAD WORD COUNT INTO LINE'S TABLE ADDRESS
1030             BIS      @#LINBIT,@#BARIM ;LOAD LINE POSITION INTO BAR IMAGE
1031             BIS      LINBIT,@BAR    ;START TRANSMITTING ON LINE SPECIFIED
1032             ;IN LINBIT
1033             MOV      (SP)+,%0      ;RESTORE RO
1034             NOP
1035             RTS      5              ;EXIT
1036
1037      ;SUBROUTINE TO FORM LINE BIT POSITION WITH THE LINE # IN LINE
1038      GTLINB: MOV      %0,-(SP)      ;SAVE RO ON THE STACK
1039             CLR      @#LINBIT
1040             MOV      @#LINE,%0      ;GET LINE
1041             SEC
1042             ;SET CARRY
1043             1$: ROL      LINBIT      ;SHIFT LINE BIT
1044             SUB      #2,%0          ;SUBTRACT 2 FROM LINE NUMBER
1045             BPL      1$
1046             MOV      (SP)+,%0      ;RESTORE RO
1047             RTS      7              ;EXIT
1048
1049      ;SUBROUTINE TO CHECK TRANSMITTED DATA
1050      CHKDAT: SAVREG
1051             ;SAVE THE REGISTERS ON THE STACK
1052             NOP
1053             CLR      %1              ;CLEAR CHARACTER COUNT
1054             MOV      #INTAB,%2      ;GET ADDRESS OF LINE'S INPUT BUFFER
1055             CLR      %3
1056             1$: MOV      %3,@#LINE  ;MOVE LINE # TO LINE
1057             MOV      (2)+,@#LINBUF  ;GET LINE'S INPUT BUFFER ADDRESS
1058             DEC      LINBUF          ;SUBTRACT 1 FROM LINE'S INPUT BUFFER ADDRESS
1059             2$: INC      LINBUF      ;INCREMENT LINE'S INPUT BUFFER ADDRESS
1060             MOVB     @LINBUF,@#RCV DAT ;GET RECEIVED CHARACTER
1061             MOVB     OUTBUF(1),XMTDAT ;GET TRANSMITTED CHARACTER
1062             BIC      @#CARMSK,XMTDAT ;CLEAR UNTRANSMITTED BITS
1063             DATCHK
1064             INC      %1              ;COMPARE CHARACTERS
1065             CMP      %1,%64         ;INCREMENT CHARACTER COUNT
1066             BNE      2$             ;ALL CHARACTERS BEEN COMPARED
1067             CLR      %1              ;GO CHECK NEXT CHAR. IF NOT
1068             INC      %3              ;CLEAR CHARACTER COUNT
1069             CMP      %3,%16         ;INCREMENT LINE COUNT
1070             BLT      1$             ;ALL LINES CHECKED?
1071             RSTREG
1072             RTS      7              ;BRANCH IF ALL LINES NOT CHECKED
1073                                     ;RESTORE REGISTERS
1074                                     ;EXIT SUBROUTINE
1075
1076      ;RECEIVER INTERRUPT SERVICE ROUTINE
1077      RINT:  NOP
1078             SAVREG
1079             ;BEGIN
1080             ;SAVE THE REGISTERS ON THE STACK

```

L02

```

1077 004672 013701 001300      MOV      3#TTPTR,%1      ;GET TUMBLE TABLE POINTER
1078 004676 011137 001272      MOV      (1),TTDAT      ;GET TUMBLE TABLE ENTRY
1079 004702 100410              BMI      2$      ;BRANCH IF VALID DATA ENTRY
1080 004704 104003              ERROR      ;ERROR! FALSE INTERRUPT
1081 004706 000454              BR      6$      ;EXIT
1082 004710 011137 001272      1$: MOV      (1),3#TTDAT      ;GET TUMBLE TABLE ENTRY
1083 004714 001451              BEQ      6$      ;GO TO EXIT IF NO DATA ENTRY
1084 004716 100402              BMI      2$      ;BRANCH IF VALID DATA ENTRY
1085 004720 104003              ERROR      ;ERROR! NO VALID DATA ENTRY INDICATOR
1086 004722 000425              BR      3$
1087 004724 005011              CLR      (1)      ;CLEAR TUMBLE TABLE ENTRY
1088 004726 042737 160400 001272 2$: BIC      #160400,3#TTDAT ;CLEAR ALL BUT CHAR. & LINE #
1089 004734 113702 001273      MOV      TTDAT+1,%2      ;PUT LINE # IN R2 (LINE WILL BE IN LSH)
1090 004740 010204              MOV      %2,%4
1091 004742 016237 012066 001650 MOV      INTAB(2),3#LINBUF ;GET LINE'S INPUT BUFFER ADDRESS
1092 004750 006202              ASR      %2      ;SHIFT LINE #
1093 004752 005003              CLR      %3
1094 004754 116203 012126      MOV      CNTTAB(2),%3      ;GET LINE'S RECEIVED CHAR. COUNT
1095 004760 105262 012126      INCB     CNTTAB(2)      ;INCREMENT CHARACTER COUNT
1096 004764 060337 001650      ADD      %3,LINBUF      ;FORM ADDRESS WHERE CHAR. IS TO BE STORED
1097 004770 113777 001272 174652 MOV      TTDAT,3#LINBUF ;STORE CHAR. IN LINE'S INPUT BUFFER
1098 004776 000240              NOP
1099 005000 016437 001146 001630 3$: MOV      WCT(4),RCVDAT ;GET TRANSMITTERS WORD COUNT
1100 005006 003405              BLE      4$      ;BRANCH IF WORD COUNT IS 0 OR NEGATIVE
1101 005010 010437 001632      MOV      %4,XMTDAT      ;GET LINE # OF FAILING LINE
1102 005014 104015              ERROR1 ;ERROR! INCORRECT WORD COUNT IN
1103              ;TYPE OUT SHOWS FAILING LINE #, AND FAILING LINE'S WORD COUNT
1104 005016 000005              RESET
1105 005020 104012              SCOPE
1106              ;STOP THE DM11
1107 005022 022701 001504      4$: CMP      #TUMTAB+176,%1 ;IS THE TUMBLE TABLE POINTER AT THE
1108 005026 001002              BNE      5$      ;THE END OF THE TABLE
1109 005030 012701 001304      MOV      #TUMTAB-2,%1 ;RESET POINTER
1110 005034 005721              5$: TST      (1)+      ;INCREMENT POINTER
1111 005036 000724              BR      1$      ;GO CHECK NEXT ENTRY
1112 005040 042777 000200 174202 6$: BIC      #BIT7,3CSR ;CLEAR RECEIVER DONE FLAG
1113 005046 010137 001300      MOV      %1,TTPTR      ;SAVE POINTER
1114 005052 104014              RSTREG ;RESTORE THE REGISTERS
1115 005054 000240              NOP
1116 005056 000002              RTI      ;EXIT SERVICE ROUTINE
1117
1118      ;TRANSMITTER INTERRUPT SERVICE ROUTINE
1119 005060 000240      TINT: NOP      ;BEGIN
1120 005062 032777 060000 174160 BIT      #BIT14+BIT13,3CSR ;TEST ERROR FLAGS
1121 005070 001404              BEQ      1$      ;BRANCH IF NO ERROR FLAGS
1122 005072 104003              ERROR ;ERROR! ERROR FLAG IS SET
1123 005074 042777 060000 174146 BIC      #BIT14+BIT13,3CSR ;CLEAR ERROR FLAGS
1124 005102 005777 174142      1$: TST      3CSR      ;TEST READY FLAG
1125 005106 100003              BPL      2$      ;BRANCH IF READY IS CLEAR
1126 005110 042777 100000 174132 BIC      #BIT15,3CSR ;CLEAR READY FLAG
1127 005116 000002      2$: RTI
1128
1129

```



```

1130
1131 005120 104000
1132 005122 012661
1133 005124 012737 005160 001516 PRGO: TYPE
1134 005132 005037 001522 PRGOM
1135 005136 000137 002170 PRGOA: MOV #RTO,KSTART ;GET ADDRESS OF FIRST TEST
1136 005142 012737 005160 001516 PRGOR: CLR RTNNO ;CLEAR ROUTINE #
1137 005150 005037 001522 PRGOR: JMP SRSET ;GET ADDRESS OF FIRST TEST
1138 005154 000137 002212 PRGOR: MOV #RTO,KSTART ;GET ADDRESS OF FIRST TEST
1139 *****
1140 005160 000000 RT0: 0 ;ROUTINE # 0
1141 005162 005200 RT1: RT1 ;ADDR OF NEXT ROUTINE.
1142 005164 000002 2 ;ITERATION COUNT
1143 005166 005170 RT0A ;SCOPE ENTRY POINT.
1144 000000 X=X+1
1145 *****
1146 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1147 ;NEXT LINE.
1148 005170 004537 003732 RT0A: JSR 5,DLYXMT ;GO DO TEST.
1149 005174 013203 MSG1 ;TRANSMIT THIS MESSAGE &
1150 005176 000000 0 ;DELAY THIS MUCH BETWEEN LINES
1151 *****
1152 005200 000001 RT1: 1 ;ROUTINE # 1
1153 005202 005220 RT2: RT2 ;ADDR OF NEXT ROUTINE.
1154 005204 000002 2 ;ITERATION COUNT
1155 005206 005210 RT1A ;SCOPE ENTRY POINT.
1156 000001 X=X+1
1157 *****
1158 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1159 ;NEXT LINE.
1160 005210 004537 003732 RT1A: JSR 5,DLYXMT ;GO DO TEST.
1161 005214 013203 MSG1 ;TRANSMIT THIS MESSAGE &
1162 005216 177740 -32. ;DELAY THIS MUCH BETWEEN LINES
1163 *****
1164 005220 000002 RT2: 2 ;ROUTINE # 2
1165 005222 005240 RT3: RT3 ;ADDR OF NEXT ROUTINE.
1166 005224 000002 2 ;ITERATION COUNT
1167 005226 005230 RT2A ;SCOPE ENTRY POINT.
1168 000002 X=X+1
1169 *****
1170 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1171 ;NEXT LINE.
1172 005230 004537 003732 RT2A: JSR 5,DLYXMT ;GO DO TEST.
1173 005234 013203 MSG1 ;TRANSMIT THIS MESSAGE &
1174 005236 177720 -48. ;DELAY THIS MUCH BETWEEN LINES
1175 *****
1176 005240 000003 RT3: 3 ;ROUTINE # 3
1177 005242 005260 RT4: RT4 ;ADDR OF NEXT ROUTINE.
1178 005244 000002 2 ;ITERATION COUNT
1179 005246 005250 RT3A ;SCOPE ENTRY POINT.
1180 000003 X=X+1
1181 *****
1182 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1183 ;NEXT LINE.
1184 005250 004537 003732 RT3A: JSR 5,DLYXMT ;GO DO TEST.
1185 005254 013203 MSG1 ;TRANSMIT THIS MESSAGE &

```

```

1186 005256 177710
1187
1188 005260 000004
1189 005262 005300
1190 005264 000002
1191 005266 005270
1192 000004
1193
1194
1195
1196 005270 004537 003732
1197 005274 013203
1198 005276 177704
1199
1200 005300 000005
1201 005302 005320
1202 005304 000002
1203 005306 005310
1204 000005
1205
1206
1207
1208 005310 004537 003732
1209 005314 013203
1210 005316 177702
1211
1212 005320 000006
1213 005322 005340
1214 005324 000002
1215 005326 005330
1216 000006
1217
1218
1219
1220 005330 004537 003732
1221 005334 013203
1222 005336 177701
1223
1224 005340 000007
1225 005342 005360
1226 005344 000002
1227 005346 005350
1228 000007
1229
1230
1231
1232 005350 004537 003732
1233 005354 013203
1234 005356 177700
1235
1236 005360 000010
1237 005362 005400
1238 005364 000002
1239 005366 005370
1240 000010

```

```

-56. ;DELAY THIS MUCH BETWEEN LINES
*****
RT4: 4 ;ROUTINE # 4 *
RT5 ;ADDR OF NEXT ROUTINE. *
2 ;ITERATION COUNT *
RT4A ;SCOPE ENTRY POINT. *
X=X+1
*****
;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
;NEXT LINE.
RT4A: JSR 5,DLYXMT ;GO DO TEST.
MSG1 ;TRANSMIT THIS MESSAGE &
-60. ;DELAY THIS MUCH BETWEEN LINES
*****
RT5: 5 ;ROUTINE # 5 *
RT6 ;ADDR OF NEXT ROUTINE. *
2 ;ITERATION COUNT *
RT5A ;SCOPE ENTRY POINT. *
X=X+1
*****
;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
;NEXT LINE.
RT5A: JSR 5,DLYXMT ;GO DO TEST.
MSG1 ;TRANSMIT THIS MESSAGE &
-62. ;DELAY THIS MUCH BETWEEN LINES
*****
RT6: 6 ;ROUTINE # 6 *
RT7 ;ADDR OF NEXT ROUTINE. *
2 ;ITERATION COUNT *
RT6A ;SCOPE ENTRY POINT. *
X=X+1
*****
;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
;NEXT LINE.
RT6A: JSR 5,DLYXMT ;GO DO TEST.
MSG1 ;TRANSMIT THIS MESSAGE &
-63. ;DELAY THIS MUCH BETWEEN LINES
*****
RT7: 7 ;ROUTINE # 7 *
RT10 ;ADDR OF NEXT ROUTINE. *
2 ;ITERATION COUNT *
RT7A ;SCOPE ENTRY POINT. *
X=X+1
*****
;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
;NEXT LINE.
RT7A: JSR 5,DLYXMT ;GO DO TEST.
MSG1 ;TRANSMIT THIS MESSAGE &
-64. ;DELAY THIS MUCH BETWEEN LINES
*****
RT10: 10 ;ROUTINE # 10 *
RT11 ;ADDR OF NEXT ROUTINE. *
2 ;ITERATION COUNT *
RT10A ;SCOPE ENTRY POINT. *
X=X+1
*****

```


B03

```

:TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
:NEXT LINE.

```

ATTN: JSA 5, DLYXMT :GO DO TEST.
MSG2 :TRANSMIT THIS MESSAGE &
-32. :DELAY THIS MUCH BETWEEN LINES

```

*****
ROUTINE # 11
ADDR OF NEXT ROUTINE.
ITERATION COUNT
SCOPE ENTRY POINT.
*****

```

```

*****
:TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE

```

NEXT LINE
ATTN: JGR 5, DLYXMT :GO DO TEST.
MSG2 :TRANSMIT THIS MESSAGE 3
-40- :DELAY THIS MUCH BETWEEN LINES

```

*****
ROUTINE # 12
ADDR OF NEXT ROUTINE.
ITERATION COUNT
SCOPE ENTRY POINT.
*****

```

:TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE

NEXT LINE
RT12A: :COR :GO DO TEST.
:MOG2 :TRANSMIT THIS MESSAGE 3
:JUGG :DELAY THIS MUCH BETWEEN LINES

```

*****
RT13:      13          :ROUTINE # 13
           14          :ADDR OF NEXT ROUTINE.
           15          :ITERATION COUNT
           16          :SCOPE ENTRY POINT.
           17          :
           18          :
           19          :
           20          :
           21          :
           22          :
           23          :
           24          :
           25          :
           26          :
           27          :
           28          :
           29          :
           30          :
           31          :
           32          :
           33          :
           34          :
           35          :
           36          :
           37          :
           38          :
           39          :
           40          :
           41          :
           42          :
           43          :
           44          :
           45          :
           46          :
           47          :
           48          :
           49          :
           50          :
           51          :
           52          :
           53          :
           54          :
           55          :
           56          :
           57          :
           58          :
           59          :
           60          :
           61          :
           62          :
           63          :
           64          :
           65          :
           66          :
           67          :
           68          :
           69          :
           70          :
           71          :
           72          :
           73          :
           74          :
           75          :
           76          :
           77          :
           78          :
           79          :
           80          :
           81          :
           82          :
           83          :
           84          :
           85          :
           86          :
           87          :
           88          :
           89          :
           90          :
           91          :
           92          :
           93          :
           94          :
           95          :
           96          :
           97          :
           98          :
           99          :
          100          :
          101          :
          102          :
          103          :
          104          :
          105          :
          106          :
          107          :
          108          :
          109          :
          110          :
          111          :
          112          :
          113          :
          114          :
          115          :
          116          :
          117          :
          118          :
          119          :
          120          :
          121          :
          122          :
          123          :
          124          :
          125          :
          126          :
          127          :
          128          :
          129          :
          130          :
          131          :
          132          :
          133          :
          134          :
          135          :
          136          :
          137          :
          138          :
          139          :
          140          :
          141          :
          142          :
          143          :
          144          :
          145          :
          146          :
          147          :
          148          :
          149          :
          150          :
          151          :
          152          :
          153          :
          154          :
          155          :
          156          :
          157          :
          158          :
          159          :
          160          :
          161          :
          162          :
          163          :
          164          :
          165          :
          166          :
          167          :
          168          :
          169          :
          170          :
          171          :
          172          :
          173          :
          174          :
          175          :
          176          :
          177          :
          178          :
          179          :
          180          :
          181          :
          182          :
          183          :
          184          :
          185          :
          186          :
          187          :
          188          :
          189          :
          190          :
          191          :
          192          :
          193          :
          194          :
          195          :
          196          :
          197          :
          198          :
          199          :
          200          :
          201          :
          202          :
          203          :
          204          :
          205          :
          206          :
          207          :
          208          :
          209          :
          210          :
          211          :
          212          :
          213          :
          214          :
          215          :
          216          :
          217          :
          218          :
          219          :
          220          :
          221          :
          222          :
          223          :
          224          :
          225          :
          226          :
          227          :
          228          :
          229          :
          230          :
          231          :
          232          :
          233          :
          234          :
          235          :
          236          :
          237          :
          238          :
          239          :
          240          :
          241          :
          242          :
          243          :
          244          :
          245          :
          246          :
          247          :
          248          :
          249          :
          250          :
          251          :
          252          :
          253          :
          254          :
          255          :
          256          :
          257          :
          258          :
          259          :
          260          :
          261          :
          262          :
          263          :
          264          :
          265          :
          266          :
          267          :
          268          :
          269          :
          270          :
          271          :
          272          :
          273          :
          274          :
          275          :
          276          :
          277          :
          278          :
          279          :
          280          :
          281          :
          282          :
          283          :
          284          :
          285          :
          286          :
          287          :
          288          :
          289          :
          290          :
          291          :
          292          :
          293          :
          294          :
          295          :
          296          :
          297          :
          298          :
          299          :
          300          :
          301          :
          302          :
          303          :
          304          :
          305          :
          306          :
          307          :
          308          :
          309          :
          310          :
          311          :
          312          :
          313          :
          314          :
          315          :
          316          :
          317          :
          318          :
          319          :
          320          :
          321          :
          322          :
          323          :
          324          :
          325          :
          326          :
          327          :
          328          :
          329          :
          330          :
          331          :
          332          :
          333          :
          334          :
          335          :
          336          :
          337          :
          338          :
          339          :
          340          :
          341          :
          342          :
          343          :
          344          :
          345          :
          346          :
          347          :
          348          :
          349          :
          350          :
          351          :
          352          :
          353          :
          354          :
          355          :
          356          :
          357          :
          358          :
          359          :
          360          :
          361          :
          362          :
          363          :
          364          :
          365          :
          366          :
          367          :
          368          :
          369          :
          370          :
          371          :
          372          :
          373          :
          374          :
          375          :
          376          :
          377          :
          378          :
          379          :
          380          :
          381          :
          382          :
          383          :
          384          :
          385          :
          386          :
          387          :
          388          :
          389          :
          390          :
          391          :
          392          :
          393          :
          394          :
          395          :
          396          :
          397          :
          398          :
          399          :
          400          :
          401          :
          402          :
          403          :
          404          :
          405          :
          406          :
          407          :
          408          :
          409          :
          410          :
          411          :
          412          :
          413          :
          414          :
          415          :
          416          :
          417          :
          418          :
          419          :
          420          :
          421          :
          422          :
          423          :
          424          :
          425          :
          426          :
          427          :
          428          :
          429          :
          430          :
          431          :
          432          :
          433          :
          434          :
          435          :
          436          :
          437          :
          438          :
          439          :
          440          :
          441          :
          442          :
          443          :
          444          :
          445          :
          446          :
          447          :
          448          :
          449          :
          450          :
          451          :
          452          :
          453          :
          454          :
          455          :
          456          :
          457          :
          458          :
          459          :
          460          :
          461          :
          462          :
          463          :
          464          :
          465          :
          466          :
          467          :
          468          :
          469          :
          470          :
          471          :
          472          :
          473          :
          474          :
          475          :
          476          :
          477          :
          478          :
          479          :
          480          :
          481          :
          482          :
          483          :
          484          :
          485          :
          486          :
          487          :
          488          :
          489          :
          490          :
          491          :
          492          :
          493          :
          494          :
          495          :
          496          :
          497          :
          498          :
          499          :
          500          :
          501          :
          502          :
          503          :
          504          :
          505          :
          506          :
          507          :
          508          :
          509          :
          510          :
          511          :
          512          :
          513          :
          514          :
          515          :
          516          :
          517          :
          518          :
          519          :
          520          :
          521          :
          522          :
          523          :
          524          :
          525          :
          526          :
          527          :
          528          :
          529          :
          530          :
          531          :
          
```

:*****
:TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE

```

NEXT LINE.
RT13A: JSR      5,DLYXMT      :GO DO TEST.
      MSG2      :TRANSMIT THIS MESSAGE &
      -GO.      :DELAY THIS MUCH BETWEEN LINES

```

```
*****
RT14:      14          :ROUTINE # 14
           RT15        :ADDR OF NEXT ROUTINE.
           2           :ITERATION COUNT
           RT14A       :SCOPE ENTRY POINT.
           X=X+1
*****
```

;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE

RT14A: JSR 5,DLYXMT :GO DO TEST.
MSG2 :TRANSMIT THIS MESSAGE 3
-62. :DELAY THIS MUCH BETWEEN LINES

```
*****
RT15:      15                      :ROUTINE # 15                *
          RT16                     :ADDR OF NEXT ROUTINE.        *
```

```

1298 005504 000002          2          : ITERATION COUNT          *
1299 005506 005510          RT15A        : SCOPE ENTRY POINT.      *
1300 000015          X=X+1
1301 *****
1302 : TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1303 : NEXT LINE.
1304 005510 004537 003732 RT15A: JSR      5,DLYXMT      : GO DO TEST.
1305 005514 013304          MSG2          : TRANSMIT THIS MESSAGE 3
1306 005516 177701          -63.          : DELAY THIS MUCH BETWEEN LINES
1307 *****
1308 005520 000016          RT16: 16          : ROUTINE # 16
1309 005522 005540          RT17          : ADDR OF NEXT ROUTINE.
1310 005524 000002          2            : ITERATION COUNT
1311 005526 005530          RT16A         : SCOPE ENTRY POINT.
1312 000016          X=X+1
1313 *****
1314 : TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1315 : NEXT LINE.
1316 005530 004537 003732 RT16A: JSR      5,DLYXMT      : GO DO TEST.
1317 005534 013304          MSG2          : TRANSMIT THIS MESSAGE 3
1318 005536 177700          -64.          : DELAY THIS MUCH BETWEEN LINES
1319 *****
1320 005540 000017          RT17: 17          : ROUTINE # 17
1321 005542 005560          RT20          : ADDR OF NEXT ROUTINE.
1322 005544 000002          2            : ITERATION COUNT
1323 005546 005550          RT17A         : SCOPE ENTRY POINT.
1324 000017          X=X+1
1325 *****
1326 : TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1327 : NEXT LINE.
1328 005550 004537 003732 RT17A: JSR      5,DLYXMT      : GO DO TEST.
1329 005554 013404          MSG3          : TRANSMIT THIS MESSAGE 3
1330 005556 177720          -49.          : DELAY THIS MUCH BETWEEN LINES
1331 *****
1332 005560 000020          RT20: 20          : ROUTINE # 20
1333 005562 005600          RT21          : ADDR OF NEXT ROUTINE.
1334 005564 000002          2            : ITERATION COUNT
1335 005566 005570          RT20A         : SCOPE ENTRY POINT.
1336 000020          X=X+1
1337 *****
1338 : TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1339 : NEXT LINE.
1340 005570 004537 003732 RT20A: JSR      5,DLYXMT      : GO DO TEST.
1341 005574 013404          MSG3          : TRANSMIT THIS MESSAGE 3
1342 005576 177704          -60.          : DELAY THIS MUCH BETWEEN LINES
1343 *****
1344 005600 000021          RT21: 21          : ROUTINE # 21
1345 005602 005620          RT22          : ADDR OF NEXT ROUTINE.
1346 005604 000002          2            : ITERATION COUNT
1347 005606 005610          RT21A         : SCOPE ENTRY POINT.
1348 000021          X=X+1
1349 *****
1350 : TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1351 : NEXT LINE.
1352 005610 004537 003732 RT21A: JSR      5,DLYXMT      : GO DO TEST.
1353 005614 013404          MSG3          : TRANSMIT THIS MESSAGE 3

```



```

1354 005616 177701          -53.          ;DELAY THIS MUCH BETWEEN LINES
1355 *****
1356 005620 000022      RT22: 22          ;ROUTINE # 22
1357 005622 005640          RT23          ;ADDR OF NEXT ROUTINE.
1358 005624 000002          2            ;ITERATION COUNT
1359 005626 005630          RT22A        ;SCOPE ENTRY POINT.
1360          000022          X=X+1
1361 *****
1362 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1363 ;NEXT LINE.
1364 005630 004537 003732      RT22A: JSR      5,DLYXMT      ;GO DO TEST.
1365 005634 013404          MSG3          ;TRANSMIT THIS MESSAGE &
1366 005636 177700          -64.          ;DELAY THIS MUCH BETWEEN LINES
1367 *****
1368 005640 000023      RT23: 23          ;ROUTINE # 23
1369 005642 005660          RT24          ;ADDR OF NEXT ROUTINE.
1370 005644 000002          2            ;ITERATION COUNT
1371 005646 005650          RT23A        ;SCOPE ENTRY POINT.
1372          000023          X=X+1
1373 *****
1374 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1375 ;NEXT LINE.
1376 005650 004537 003732      RT23A: JSR      5,DLYXMT      ;GO DO TEST.
1377 005654 013504          MSG4          ;TRANSMIT THIS MESSAGE &
1378 005656 177740          -32.          ;DELAY THIS MUCH BETWEEN LINES
1379 *****
1380 005660 000024      RT24: 24          ;ROUTINE # 24
1381 005662 005700          RT25          ;ADDR OF NEXT ROUTINE.
1382 005664 000002          2            ;ITERATION COUNT
1383 005666 005670          RT24A        ;SCOPE ENTRY POINT.
1384          000024          X=X+1
1385 *****
1386 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1387 ;NEXT LINE.
1388 005670 004537 003732      RT24A: JSR      5,DLYXMT      ;GO DO TEST.
1389 005674 013504          MSG4          ;TRANSMIT THIS MESSAGE &
1390 005676 177710          -56.          ;DELAY THIS MUCH BETWEEN LINES
1391 *****
1392 005700 000025      RT25: 25          ;ROUTINE # 25
1393 005702 005720          RT26          ;ADDR OF NEXT ROUTINE.
1394 005704 000002          2            ;ITERATION COUNT
1395 005706 005710          RT25A        ;SCOPE ENTRY POINT.
1396          000025          X=X+1
1397 *****
1398 ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1399 ;NEXT LINE.
1400 005710 004537 003732      RT25A: JSR      5,DLYXMT      ;GO DO TEST.
1401 005714 013504          MSG4          ;TRANSMIT THIS MESSAGE &
1402 005716 177702          -62.          ;DELAY THIS MUCH BETWEEN LINES
1403 *****
1404 005720 000026      RT26: 26          ;ROUTINE # 26
1405 005722 005740          RT27          ;ADDR OF NEXT ROUTINE.
1406 005724 000002          2            ;ITERATION COUNT
1407 005726 005730          RT26A        ;SCOPE ENTRY POINT.
1408          000026          X=X+1
1409 *****

```

E03

MAINDEC-11-DZDMB DM11 DATA TESTS
DZDMBB.SRC

MACY11 27(732) 21-SEP-76 15:45 PAGE 30

```

1410
1411
1412 005730 004537 003732
1413 005734 013504
1414 005736 177700
1415
1416 005740 000027
1417 005742 005760
1418 005744 000002
1419 005746 005750
1420 000027
1421
1422
1423
1424 005750 004537 003732
1425 005754 013604
1426 005756 177720
1427
1428 005760 000030
1429 005762 006000
1430 005764 000002
1431 005766 005770
1432 000030
1433
1434
1435
1436 005770 004537 003732
1437 005774 013604
1438 005776 177710
1439
1440 006000 000031
1441 006002 006020
1442 006004 000002
1443 006006 006010
1444 000031
1445
1446
1447
1448 006010 004537 003732
1449 006014 013604
1450 006016 177704
1451
1452 006020 000032
1453 006022 006040
1454 006024 000002
1455 006026 006030
1456 000032
1457
1458
1459
1460 006030 004537 003732
1461 006034 013604
1462 006036 177702
1463
1464 006040 000033
1465 006042 006060

```

```

:TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
:NEXT LINE.
RT26A: JSR      5,DLYXMT      :GO DO TEST.
      MSG4      :TRANSMIT THIS MESSAGE &
      -64.      :DELAY THIS MUCH BETWEEN LINES
:*****
RT27:  27      :ROUTINE # 27
      RT30      :ADDR OF NEXT ROUTINE.
      2         :ITERATION COUNT
      RT27A     :SCOPE ENTRY POINT.
      X=X+1
:*****
:TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
:NEXT LINE.
RT27A: JSR      5,DLYXMT      :GO DO TEST.
      MSG5      :TRANSMIT THIS MESSAGE &
      -48.      :DELAY THIS MUCH BETWEEN LINES
:*****
RT30:  30      :ROUTINE # 30
      RT31      :ADDR OF NEXT ROUTINE.
      2         :ITERATION COUNT
      RT30A     :SCOPE ENTRY POINT.
      X=X+1
:*****
:TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
:NEXT LINE.
RT30A: JSR      5,DLYXMT      :GO DO TEST.
      MSG5      :TRANSMIT THIS MESSAGE &
      -56.      :DELAY THIS MUCH BETWEEN LINES
:*****
RT31:  31      :ROUTINE # 31
      RT32      :ADDR OF NEXT ROUTINE.
      2         :ITERATION COUNT
      RT31A     :SCOPE ENTRY POINT.
      X=X+1
:*****
:TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
:NEXT LINE.
RT31A: JSR      5,DLYXMT      :GO DO TEST.
      MSG5      :TRANSMIT THIS MESSAGE &
      -60.      :DELAY THIS MUCH BETWEEN LINES
:*****
RT32:  32      :ROUTINE # 32
      RT33      :ADDR OF NEXT ROUTINE.
      2         :ITERATION COUNT
      RT32A     :SCOPE ENTRY POINT.
      X=X+1
:*****
:TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
:NEXT LINE.
RT32A: JSR      5,DLYXMT      :GO DO TEST.
      MSG5      :TRANSMIT THIS MESSAGE &
      -62.      :DELAY THIS MUCH BETWEEN LINES
:*****
RT33:  33      :ROUTINE # 33
      RT34      :ADDR OF NEXT ROUTINE.

```



```

1466 006044 000002          2          : ITERATION COUNT          *
1467 006046 006050          RT33A       : SCOPE ENTRY POINT.      *
1468 000033          X=X+1
1469 *****
1470 : TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1471 : NEXT LINE.
1472 006050 004537 003732 RT33A: JSR      5,DLYXMT      : GO DO TEST.
1473 006054 013604          MSG5         : TRANSMIT THIS MESSAGE 3
1474 006056 177701          -63.         : DELAY THIS MUCH BETWEEN LINES
1475 *****
1476 006060 000034          RT34: 34         : ROUTINE # 34          *
1477 006062 177777          RT35         : ADDR OF NEXT ROUTINE. *
1478 006064 000002          2           : ITERATION COUNT      *
1479 006066 006070          RT34A        : SCOPE ENTRY POINT.   *
1480 000034          X=X+1
1481 *****
1482 : TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
1483 : NEXT LINE.
1484 006070 004537 003732 RT34A: JSR      5,DLYXMT      : GO DO TEST.
1485 006074 013604          MSG5         : TRANSMIT THIS MESSAGE 3
1486 006076 177700          -64.         : DELAY THIS MUCH BETWEEN LINES
1487 177777
RT35=-1

```

```

1488
1489
1490
1491 006100 104000
1492 006102 012707
1493 006104 022737 000176 001102 PRGX: CMP #SWREG,SWR
1494 006112 001001 PRG1: TYPE
1495 006114 104020 PRG1: PRGIM
1496 006116 004537 003732 PRGX: BNE PRG1R
1497 006122 013203 PRG1R: CNTLU
1498 006124 177700 PRG1R: JSR 5,DLYXMT
1499 006126 012737 006622 001106 PRG1A: MSG1
1500 006134 004537 004440 PRG1A: MOV #OUTBUF,CAT
1501 006140 001106 PRG1A: JSR 5,BMOVE
1502 006142 001110 PRG1A: CAT
1503 006144 000040 PRG1A: CAT+2
1504 006146 012737 177700 001146 PRG1A: MOV #-64.,WCT
1505 006154 004537 004440 PRG1A: JSR 5,BMOVE
1506 006160 001146 PRG1A: WCT
1507 006162 001150 PRG1A: WCT+2
1508 006164 000040 PRG1A: 32.
1509 006166 012777 010100 173054 PRG1A: MOV #BIT12+BIT6,DCSR
1510 006174 023727 002130 000004 PRG1A: CMP PRGNUM,#4
1511 006202 001403 PRG1A: BEQ +10
1512 006204 052777 000001 173036 PRG1A: BIS #BIT0,DCSR
1513 006212 012777 177777 173032 PRG1A: MOV #-1,ABAR
1514 006220 005777 173026 PRG1A: TST ABAR
1515 006224 001375 PRG1A: BNE -4
1516 006226 005205 PRG1A: INC %5
1517 006230 001376 PRG1A: BNE -2
1518 006232 005077 173012 PRG1A: CLR DCSR
1519 006236 023727 002130 000004 PRG1C: CMP PRGNUM,#4
1520 006244 001402 PRG1C: BEQ PRG1D
1521 006246 004737 004560 PRG1C: JSR 7,CHKDAT
1522 006252 104000 PRG1D: TYPE
1523 006254 012543 PRG1D: M2
1524 006256 012706 001100 PRG1EX: MOV #SPBOT,SP
1525 006262 000715 PRG1EX: BR PRG1R
1526
1527
1528
1529
1530
1531
1532 006264 104000
1533 006266 012767
1534 006270 000137 006104 PRG2: TYPE
PRG2: PRG2M
PRG2R: JMP PRGX

:PRG1- DATA TESTS ALL LINES SIMULTANEOUSLY. DATA TRANSMITTED IS 'THE
:QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 1234567890'
:TYPE
:PROGRAM TITLE
:SEE IF SWITCH-LESS
:BRANCH IF NOT
:GET SWREG SETTINGS
:GO TO DLYXMT TO SET UP DM11
:MSG1 WILL BE THE DATA TRANSMITTED
:DO NOT DELAY
:LOAD CURRENT
:ADDRESS TABLE
:TO POINT TO
:OUTBUF
:LOAD WORD COUNT
:TO -64.
:SET TRANSMITTER & RECIEVER TO BITS
:RUNNING PROGRAM #2?
:SET THE GO BIT
:START TRANSMITTING ON ALL LINES
:WAIT FOR ALL LINES TO COMPLETE
:DO NOT CHECK DATA IF RUNNING
:PROGRAM # 2
:GO CHECK RECEIVED DATA
:TYPE
:'PRGEND'
:RESET THE STACK POINTER
:GO RESTART TEST

:PRG2-PROGRAM 2 RUNS PROGRAM 1 EXCEPT FOR THE DATA CHECKING
:WHEN ALL LINES ARE FINISHED TRANSMITTING. THIS ALLOWS THE DATA
:TRANSMITTED TO BE SENT TO TERMINALS. BEFORE STARTING THIS PROGRAM
:REMOVE THE JUMPERS CONNECTING THE TRANSMITTERS TO THE RECEIVERS.
:TYPE PROGRAM TITLE
:AND INSTRUCTIONS
:GO RUN PRG1

```


H03

```

1535
1536
1537
1538
1539
1540
1541 006274 104000
1542 006276 013016
1543 006300 022737 000176 001102
1544 006306 001001
1545 006310 104020
1546 006312 004537 003732
1547 006316 013203
1548 006320 000240
1549 006322 012737 001306 001300
1550 006330 013701 001260
1551 006334 012721 006372
1552 006340 013721 001262
1553 006344 012721 006576
1554 006350 013721 001266
1555 006354 012777 010101 172666
1556 006362 012700 000001
1557 006366 005200
1558 006370 000776
1559
1560
1561 006372 000240
1562 006374 000240
1563 006376 013701 001300
1564 006402 011137 001272
1565 006406 001463
1566 006410 005011
1567 006412 032737 040000 001272
1568 006420 001047
1569 006422 042737 160400 001272
1570 006430 113702 001273
1571 006434 010237 001646
1572 006440 004737 004526
1573 006444 033777 001274 172600
1574 006452 001414
1575 006454 033777 001274 172570
1576 006462 001374
1577 006464 032777 060000 172556
1578 006472 001401
1579 006474 104003
1580 006476 042777 100000 172544
1581 006504 113762 001272 006622
1582 006512 012762 177777 001146
1583 006520 010203
1584 006522 062703 006622
1585 006526 010362 001106
1586 006532 053777 001274 172512
1587 006540 022701 001504
1588 006544 001002
1589 006546 012701 001304
1590 006552 005721

;PRG3-ECHO TEST THIS PROGRAM ECHOS BACK DATA RECEIVED FROM ANY DM11
;TERMINAL(S)
;NOTE: THIS TEST IS THE ONLY TEST THAT INSURES PROPER OPERATION
;OF THE DM11 DISTRIBUTION PANEL LOGIC.

PRG3: TYPE
PRG3M
CMP #SWREG,SWR
BNE PRG3R
CNTLU
PRG3R: JSR 5,DLYXMT
MSG1
NOP
PRG3A: MOV #TUMTAB,TTPTR
MOV CLKINT,%1
MOV #RINT3,(1)+
MOV CLKLVL,(1)+
MOV #TINT3,(1)+
MOV XMTLVL,(1)+
MOV #BIT12+BIT6+BIT0,ACSR ;SET IE AND GO BITS
MOV #1,%0
INC %0
BR -2

;TYPE PROGRAM
;TITLE
;SEE IF SWITCH-LESS
;BRANCH IF NOT
;GET SWREG SETTINGS
;USE PART OF THE
;DLYXMT ROUTINE TO
;SET UP DM11
;INITIALIZE SOFTWARE POINTER
;LOAD RECEIVER
;AND TRANSMITTER
;VECTORS AND PRIORITY
;LEVELS

RINT3: NOP
RINT3A: MOV TTPTR,%1
MOV (1),TTDAT ;GET SOFTWARE POINTER
BEQ RINT3X ;GET TUMBLE TABLE ENTRY
CLR (1) ;EXIT IF NO ENTRY
CLR #BIT14,TTDAT ;CLEAR ENTRY
BNE RINT3B ;WAS BREAK RECEIVED
BIC #160400,TTDAT ;DO NOTHING ABOUT IT
MOV TTDAT+1,%2 ;CLEAR ALL BUT LINE # AND DATA
JSR 7,GTLINE ;GET LINE NUMBER
BIT LINBIT,ABAR ;FETCH LINE NUMBER
BEQ NONACT ;FORM LINE BIT FOR BAR
BIT LINBIT,ABAR ;IS THIS LINE ACTIVE
BNE -6 ;LINE NOT ACTIVE
BIT #BIT14+BIT13,ACSR ;WAIT FOR LINE
BEQ -4 ;BRANCH IF NO ERRORS
BIC #BIT15,ACSR ;CLEAR TRANSMIT DONE
MOV TTDAT,OUTBUF(2) ;STORE RECEIVED CHARACTER
MOV #-1,WCT(2) ;LOAD LINE'S WORD COUNT
MOV %2,%3
ADD #OUTBUF,%3
MOV %3,CAT(2) ;AND CURRENT ADDRESS
BIS LINBIT,ABAR ;ECHO RECEIVED CHARACTER
RINT3B: CMP #TUMTAB+176,%1 ;CHECK TUMBLE
BNE -6 ;TABLE POINTER
MOV #TUMTAB-2,%1
TST (1)+

```

MAINDEC-11-DZDMB DM11 DATA TESTS
DZDMBB.SRC

MACY11 27(732) 21-SEP-76 15:45 PAGE 34

1591	006554	000712				BR	RINT3A	
1592	006556	042777	000200	172464	RINT3X:	BIC	#BIT7,@CSR	;CLEAR CHARACTER DONE FLAG
1593	006564	010137	001300			MOV	%1,TTPT	;RESTORE POINTER
1594	006570	000240				NOP		
1595	006572	000240				NOP		
1596	006574	000002				RTI		;EXIT
1597								
1598	006576	000240			TINT3:	NOP		
1599	006600	032777	060000	172442		BIT	#BIT14+BIT13,@CSR	;ANY ERROR FLAGS SET
1600	006606	001401				BEQ	.+4	
1601	006610	104003				ERROR		
1602	006612	042777	160000	172430		BIC	#BIT15+BIT14+BIT13,@CSR	;CLEAR ALL FLAGS
1603	006620	000002				RTI		;EXIT


```

1604 006622 000000
1605      006766
1606 006766 000000
1607      007132
1609 007132 000000
1609      007276
1610 007276 000000
1611      007442
1612 007442 000000
1613      007606
1614 007606 000000
1615      007752
1616 007752 000000
1617      010116
1618 010116 000000
1619      010262
1620 010262 000000
1621      010426
1622 010426 000000
1623      010572
1624 010572 000000
1625      010736
1626 010736 000000
1627      011102
1628 011102 000000
1629      011246
1630 011246 000000
1631      011412
1632 011412 000000
1633      011556
1634 011556 000000
1635      011722
1636 011722 000000
1637      012066
1638 012066 006766
1639 012070 007132
1640 012072 007276
1641 012074 007442
1642 012076 007606
1643 012100 007752
1644 012102 010116
1645 012104 010262
1646 012106 010426
1647 012110 010572
1648 012112 010736
1649 012114 011102
1650 012116 011246
1651 012120 011412
1652 012122 011556
1653 012124 011722
1654 012126 000000
1655      012146
1656
1657 012146 012206
1658 012150 012212
1659 012152 012216

```

```

OUTBUF: 0
        . = OUTBUF + 100.
LN0BUF: 0
        . = LN0BUF + 100.
LN1BUF: 0
        . = LN1BUF + 100.
LN2BUF: 0
        . = LN2BUF + 100.
LN3BUF: 0
        . = LN3BUF + 100.
LN4BUF: 0
        . = LN4BUF + 100.
LN5BUF: 0
        . = LN5BUF + 100.
LN6BUF: 0
        . = LN6BUF + 100.
LN7BUF: 0
        . = LN7BUF + 100.
LN10BF: 0
        . = LN10BF + 100.
LN11BF: 0
        . = LN11BF + 100.
LN12BF: 0
        . = LN12BF + 100.
LN13BF: 0
        . = LN13BF + 100.
LN14BF: 0
        . = LN14BF + 100.
LN15BF: 0
        . = LN15BF + 100.
LN16BF: 0
        . = LN16BF + 100.
LN17BF: 0
        . = LN17BF + 100.
INTAB:  LN0BUF
        LN1BUF
        LN2BUF
        LN3BUF
        LN4BUF
        LN5BUF
        LN6BUF
        LN7BUF
        LN10BF
        LN11BF
        LN12BF
        LN13BF
        LN14BF
        LN15BF
        LN16BF
        LN17BF
CNTTAB: 0
        . = CNTTAB + 16.
ID:      IDENT0
        IDENT1
        IDENT2

```

1660 012154 012222
1661 012156 012226
1662 012160 012232
1663 012162 012236
1664 012164 012242
1665 012166 012246
1666 012170 012252
1667 012172 012256
1668 012174 012262
1669 012176 012266
1670 012200 012272
1671 012202 012276
1672 012204 012302
1673 012206 105215
1674 012210 030060
1675 012212 105215
1676 012214 030460
1677 012216 105215
1678 012220 031060
1679 012222 105215
1680 012224 031460
1681 012226 105215
1682 012230 032060
1683 012232 105215
1684 012234 032460
1685 012236 105215
1686 012240 033060
1687 012242 105215
1688 012244 033460
1689 012246 105215
1690 012250 030061
1691 012252 105215
1692 012254 030461
1693 012256 105215
1694 012260 031061
1695 012262 105215
1696 012264 031461
1697 012266 105215
1698 012270 032061
1699 012272 105215
1700 012274 032461
1701 012276 105215
1702 012300 033061
1703 012302 105215
1704 012304 033461
1705 012306 105215
1706 105215
1707

IDENT3
IDENT4
IDENT5
IDENT6
IDENT7
IDNT10
IDNT11
IDNT12
IDNT13
IDNT14
IDNT15
IDNT16
IDNT17
IDENT0: CRLF
"00
IDENT1: CRLF
"01
IDENT2: CRLF
"02
IDENT3: CRLF
"03
IDENT4: CRLF
"04
IDENT5: CRLF
"05
IDENT6: CRLF
"06
IDENT7: CRLF
"07
IDNT10: CRLF
"10
IDNT11: CRLF
"11
IDNT12: CRLF
"12
IDNT13: CRLF
"13
IDNT14: CRLF
"14
IDNT15: CRLF
"15
IDNT16: CRLF
"16
IDNT17: CRLF
"17
CRLF
CRLF=105215

1708						
1709						:MESSAGES
1710	012310	042045	0305'5	020061		WHERE: .ASCII '%DM11 RECEIVER VECTOR ADDRESS = 3'
1711	012316	042522	042503	053111		
1712	012324	051105	053040	041505		
1713	012332	047524	020122	042101		
1714	012340	051104	051505	020123		
1715	012346	020075	100			
1716	012351	045	053523	036522	\$SWREG: .ASCII	'%SWR= 3'
1717	012356	040040				
1718	012360	020040	020040	020040	\$VALUE: .ASCII	' NEW= 3'
1719	012366	020040	020040	042516		
1720	012374	036527	040040			
1721	012400	036445	040		\$CTLU: .ASCII	'%= '
1722	012403	045	044127	041511	WHICH: .ASCII	'%WHICH DM11 ARE YOU TESTING 3'
1723	012410	020110	046504	030461		
1724	012416	040440	042522	054440		
1725	012424	052517	052040	051505		
1726	012432	044524	043516	040040		
1727	012440	042045	052101	020101	ERDAT: .ASCII	'%DATA ERR S/B: '
1728	012446	051105	020122	051440		
1729	012454	041057	020072			
1730	012460	020040	020040	020040	AASB: .ASCII	' WAS: '
1731	012466	020040	040527	035123		
1732	012474	040				
1733	012475	040	020040	020040	AWAS: .ASCII	' 3'
1734	012502	020040	100			
1735	012505	114	047111	020105	LINEM: .ASCII	'LINE # '
1736	012512	020043				
1737	012514	020040	040040		ALINE: .ASCII	' 3'
1738	012520	052045	050131	020105	MO: .ASCII	'%TYPE PROGRAM #3'
1739	012526	051120	043517	040522		
1740	012534	020115	040043			
1741	012540	037445	100		M1: .ASCII	'%?3'
1742	012543	045	042524	052123	M2: .ASCII	'%TEST DZDMB COMPLETE3'
1743	012550	042040	042132	041115		
1744	012556	041440	046517	046120		
1745	012564	052105	040105			
1746	012570	051445	052105	051440	M3: .ASCII	'%SET SR OPTIONS. NORMAL OPERATION'
1747	012576	020122	050117	044524		
1748	012604	047117	027123	047040		
1749	012612	051117	040515	020114		
1750	012620	050117	051105	052101		
1751	012626	047511	116			
1752	012631	123	020122	020075	.ASCII	'SR = 000000 PRESS CONT.3'
1753	012636	030060	030060	030060		
1754	012644	050040	042522	051523		
1755	012652	041440	047117	027124		
1756	012660	100				
1757	012661	045	040504	040524	PRGOM: .ASCII	'%DATA TEST ALL LINES 3'
1758	012666	052040	051505	020124		
1759	012674	046101	020114	044514		
1760	012702	042516	020123	100		
1761	012707	045	040504	040524	PRGIM: .ASCII	'%DATA TEST TRANSMIT ON ALL LINES SIMULTANEOUSLY3'
1762	012714	052040	051505	020124		
1763	012722	051124	047101	046523		

1764	012730	052111	047440	020116
1765	012736	046101	020114	044514
1766	012744	042516	020123	044523
1767	012752	052515	052114	047101
1768	012760	047505	051525	054514
1769	012766	100		

1771	012767	045	051124	047101
1772	012774	046523	052111	052040
1773	013002	020117	042524	046522
1774	013010	047111	046101	040123
1775	013016	042445	044103	020117
1776	013024	042524	052123	100

PRG2M: .ASCII '%TRANSMIT TO TERMINALS'

PRG3M: .ASCII '%ECHO TEST'

PRG1: .ASCII '%PUT CHAR IN SR(0-7),DELAY IN SR(8-15)'

1777	013031	045	052520	020124
1778	013036	044103	051101	044440
1779	013044	020116	051123	030050
1780	013052	033455	026051	042504
1781	013060	040514	020131	047111
1782	013066	051440	024122	026470

POPPAR: .ASCII '%TYPE PARITY OPTION (N=NOT DESIRED O=ODD, E=EVEN)'

1783	013074	032461	040051	
1784	013100	052045	050131	020105
1785	013106	040520	044522	054524
1786	013114	047440	052120	047511
1787	013122	020116	047050	047075
1788	013130	052117	042040	051505
1789	013136	051111	042105	047440
1790	013144	047475	042104	020054
1791	013152	036505	053105	047105

1792	013160	100		
1793	013161	045	020122	
1794	013164	020040	020040	041520
1795	013172	020075		
1796	013174	020040	020040	020040
1797	013202	100		

EMO: .ASCII '%R '

ATNUMB: .ASCII ' PC= '

APC: .ASCII ' '

MSG1: .BYTE 15,12

.ASCII ' THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 1234567890'

1798	013203	015	012	
1799	013205	040	044124	020105
1800	013212	052521	041511	020113
1801	013220	051102	053517	020116
1802	013226	047506	020130	052512
1803	013234	050115	042105	047440
1804	013242	042526	020122	044124

1805	013250	020105	040514	054532
1806	013256	042040	043517	020123
1807	013264	040502	045503	030440
1808	013272	031462	032464	033466
1809	013300	034470	060	

1810		013304		
1811	013304	015	012	
1812	013306	177400		
1813	013310	177400		
1814	013312	177400		
1815	013314	177400		
1816	013316	177400		
1817	013320	177400		
1818	013322	177400		
1819	013324	177400		

MSG2: .EVEN
.BYTE 15,12177400
177400
177400
177400
177400
177400
177400
177400
177400

1820	013326	177400				177400
1821	013330	177400				177400
1822	013332	177400				177400
1823	013334	177400				177400
1824	013336	177400				177400
1825	013340	177400				177400
1826	013342	177400				177400
1827	013344	177400				177400
1828	013346	177400				177400
1829	013350	177400				177400
1830	013352	177400				177400
1831	013354	177400				177400
1832	013356	177400				177400
1833	013360	177400				177400
1834	013362	177400				177400
1835	013364	177400				177400
1836	013366	177400				177400
1837	013370	177400				177400
1838	013372	177400				177400
1839	013374	177400				177400
1840	013376	177400				177400
1841	013400	177400				177400
1842	013402	177400				177400
1843	013404	015	012	MSG3:	.BYTE	15,12
1844	013406	125252				ALTO
1845	013410	125252				ALTO
1846	013412	125252				ALTO
1847	013414	125252				ALTO
1848	013416	125252				ALTO
1849	013420	125252				ALTO
1850	013422	125252				ALTO
1851	013424	125252				ALTO
1852	013426	125252				ALTO
1853	013430	125252				ALTO
1854	013432	125252				ALTO
1855	013434	125252				ALTO
1856	013436	125252				ALTO
1857	013440	125252				ALTO
1858	013442	125252				ALTO
1859	013444	125252				ALTO
1860	013446	125252				ALTO
1861	013450	125252				ALTO
1862	013452	125252				ALTO
1863	013454	125252				ALTO
1864	013456	125252				ALTO
1865	013460	125252				ALTO
1866	013462	125252				ALTO
1867	013464	125252				ALTO
1868	013466	125252				ALTO
1869	013470	125252				ALTO
1870	013472	125252				ALTO
1871	013474	125252				ALTO
1872	013476	125252				ALTO
1873	013500	125252				ALTO
1874	013502	125252				ALTO
1875	013504	015	012	MSG4:	.BYTE	15,12

[illegible]

MSG5:

15.12

4000 100010 40040 2000 177377 175775 167767 137737 177500 400 2002 10010 40040 200 177377 175775 167767 137737 177500 400 2002 10010 40040

C04

MAINDEC-11-DZDMB DM11 DATA TESTS
DZDMBB.SRC

MACY11 27(732) 21-SEP-76 15:45 PAGE 41

11933	013666	000300
11934	013670	177377
11935	013672	175775
11936	013674	167767
11937	013676	137737
11938	013678	177500
11939	013700	015
11940	013702	015
11941	013704	000001

015
015

200
177377
175775
167767
137737
177500
.BYTE 15.12
.BYTE 15.12
.END

Variable	Value	1845	1846	1847	1848	1849	1850	1851	1852	1853	1854	1855
ALTSB	012460	1730*										
ALINE	012514	1737*										
ALTO	125252	1844	1845	1846	1847	1848	1849	1850	1851	1852	1853	1854
		1857	1858	1859	1860	1861	1862	1863	1864	1865	1866	1867
		1870	1871	1872	1873	1874						
		1876	1877	1878	1879	1880	1881	1882	1883	1884	1885	1886
		1889	1890	1891	1892	1893	1894	1895	1896	1897	1898	1899
		1902	1903	1904	1905	1906						
ALT1	052525	1796*	540*	543*	549							
APC	013174	550										
APCADD	001642	514*	540*	543*	549							
ATLAST	177777	286*										
ATNUMB	013164	554	1794*									
AWAS	012475	575	1733*									
BAR	001252	452*	628*	783*	960	965*	980*	984	1031*	1513*	1514	1573
BARDAT	001276	462*										
BARIM	001270	459*	1030*									
BASREG	001256	454*	630*	774*	906*							
BAT	001206	448*	449									
BELL	000007	285*										
BIT0	000001	249*	784	943	1512	1555						
BIT1	000002	248*										
BIT10	002000	239*										
BIT11	004000	238*	635									
BIT12	010000	237*	943	1509	1555							
BIT13	020000	236*	527	546	1120	1123	1577	1599	1602			
BIT14	040000	235*	631	1120	1123	1567	1577	1599	1602			
BIT15	100000	233*	234*	1126	1580	1602						
BIT2	000004	247*	784									
BIT3	000010	246*										
BIT4	000020	245*										
BIT5	000040	244*										
BIT6	000100	243*	784	943	1509	1555						
BIT7	000200	242*	1112	1592								
BIT8	000400	241*										
BIT9	001000	240*	610	640								
CHKCSR	001254	452*	627*									
REMOVE	004440	911	916	920	925	1012*	1500	1505				
CHARMSK	001634	511*	778*	787*	1061							
CAT	001106	444*	445	630	774	779*	906	976*				

E04

MAINDEC-11-DZDMB DM11 DATA TESTS

MACY11 27(732)

21-SEP-76

15:45

PAGE 44

DZDMBS.SRC

CROSS REFERENCE TABLE -- USER SYMBOLS

DISPRE	000174	429#	895											
DLXMT	003732	905#	1148	1160	1172	1184	1196	1208	1220	1232	1244	1256	1268	1280
		1292	1304	1316	1328	1340	1352	1364	1376	1388	1400	1412	1424	1436
		1448	1460	1472	1484	1496	1508							
DMPAR	002764	583	731#	744										
DMPAR3	003100	754#	762											
DTCHK	001660	490	524#											
EHALT =	104010	276#												
EMO	013161	557	1793#											
EMTINT	002514	305	667#											
EMTTAB	001554	486#	673											
ERDAT	012440	560	1727#											
ERR	001726	489	539#											
ERRA	001764	541	545#											
ERRB	002026	539*	542*	558#										
ERRC	002034	547	561#											
ERROR =	104003	272#	1080	1085	1122	1579	1601							
ERROR1 =	104015	280#	533	966	1102									
ERRVEC =	000004	230#	605*											
ERR1	001744	499	542#											
ESCOPE	002322	496	625#											
FAJAG	003650	870	882	884#										
FORWD	002462	609	658#											
GETRDY	002212	602	604#	622	641	647	1138							
GTLINE	004526	1038#	1572											
GTRDYA	002240	609#	620											
GTRDYC	002262	611	614#											
GTRDYD	002304	617	619#											
GTRDYX	002220	605#	643											
ICTR	001526	475#	637*	661*										
ID	012146	978	1657#											
IDENT	004264	907	975#											
IDENTO	012206	1657	1673#											
IDENT1	012212	1658	1675#											
IDENT2	012216	1659	1677#											
IDENT3	012222	1660	1679#											
IDENT4	012226	1661	1681#											
IDENT5	012232	1662	1683#											
IDENT6	012236	1663	1685#											
IDENT7	012242	1664	1687#											
IDNT10	012246	1665	1689#											
IDNT11	012252	1666	1691#											
IDNT12	012256	1667	1693#											
IDNT13	012262	1668	1695#											
IDNT14	012266	1669	1697#											
IDNT15	012272	1670	1699#											
IDNT16	012276	1671	1701#											
IDNT17	012302	1672	1703#											
INCRIN	001652	519#	621											
INTAB	012066	1053	1091	1638#										
KBDIN =	104017	282#	545	564	629									
KBDINT	003526	501	856#											
KSTART	001516	471#	604	1133*	1136*									
LINBIT	001274	461#	944*	953*	976*	980	982*	1030	1031	1039*	1042*	1573	1575	1596
LINBUF	001650	517#	1056*	1057*	1058*	1059	1091*	1096*	1097*					
LINE	001646	516#	530	945*	946	952*	976*	977	981*	1027	1040	1055*	1571*	

[illegible]

G04

MAINDEC-11-DZDMB DM11 DATA TESTS MACY11 27(732) 21-SEP-76 15:45 PAGE 46
 DZDMBB.SRC CROSS REFERENCE TABLE -- USER SYMBOLS

PRG1M	012707	1492	1761#				
PRG1R	006116	483	1494	1496#	1525		
PRG2	006264	480	1532#				
PRG2M	012767	1533	1771#				
PRG2R	006270	484	1534#				
PRG3	006274	481	1541#				
PRG3A	006322	931	1543#				
PRG3M	013016	1542	1775#				
PRG3R	006312	485	1544	1546#			
PRTY0	000000	259#					
PRTY1	000040	258#					
PRTY2	000100	257#					
PRTY3	000140	256#					
PRTY4	000200	255#	468	470			
PRTY5	000240	254#	456	458			
PRTY6	000300	253#					
PRTY7	000340	252#	306				
PRVCNT	001644	515#					
PSW	177776	229#	606*	788*			
RCVDAT	001630	509#	524	574	960*	1055*	1099*
RECD	003346	588	735	756	913#	879	
RINT	004666	936	1075#				
RINT3	006372	1551	1561#				
RINT3A	006402	1564#	1591				
RINT3B	006540	1568	1587#				
RINT3X	006556	1565	1592#				
RSTART	001544	482#	596				
RSTAT1	002114	434	585#				
RSTAT2	002154	436	594#				
RSTREC	104014	279#	1005	1019	1070	1114	
RSTRG	002606	498	691#				
RTNNO	001522	473#	553	616	659*	1134*	1137*
RTC	005160	1133	1136	1140#			
RTOR	005170	1143	1148#				
RT!	005200	1141	1152#				
RT1A	005210	1155	1160#				
RT10	005360	1225	1236#				
RT10A	005370	1239	1244#				
RT11	005400	1237	1248#				
RT11A	005410	1251	1256#				
RT11B	005420	1249	1260#				
RT11C	005430	1263	1268#				
RT11D	005440	1261	1272#				
RT11E	005450	1275	1280#				
RT14	005460	1273	1284#				
RT14A	005470	1287	1292#				
RT15	005500	1285	1296#				
RT15A	005510	1299	1304#				
RT16	005520	1297	1308#				
RT16A	005530	1311	1316#				
RT17	005540	1309	1320#				
RT17A	005550	1323	1328#				
RT18	005520	1153	1164#				
RT18A	005530	1167	1172#				
RT19	005560	1321	1322#				
RT19A	005570	1325	1340#				

[illegible]

[illegible]

J04

MAINDEC-11-DZDMB DM11 DATA TESTS MACY11 27(732) 21-SEP-76 15:45 PAGE 50
 DZDMBB.SRC CROSS REFERENCE TABLE -- MACRO NAMES

GET	211#	598	735	756	879										
HEADER	211#	1139	1151	1163	1175	1187	1199	1211	1223	1235	1247	1259	1271	1283	1295
	1307	1319	1331	1343	1355	1367	1379	1391	1403	1415	1427	1439	1451	1463	1475
XMTDLY	211#	1139	1151	1163	1175	1187	1199	1211	1223	1235	1247	1259	1271	1283	1295
	1307	1319	1331	1343	1355	1367	1379	1391	1403	1415	1427	1439	1451	1463	1475

ADD	673	708	750	769	800	952	958	981	994	997	1096	1584		
ASL	591	672	763	764	765	828	929	830	953	982				
ASR	1092													
BCC	954	983												
BEQ	525	598	632	647	715	799	822	924	833	882	891	941	961	1083
	1511	1520	1565	1574	1578	1600								1121
BHI	746													
BHIS	741													
BIC	590	615	768	957	996	999	1061	1088	1112	1123	1126	1569	1580	1592
BICB	787	819	825	860										1602
BIS	980	1030	1031	1512	1586									
BISB	831													
BIT	527	546	610	631	635	640	747	1120	1567	1573	1575	1577	1599	
BITB	826													
BLE	1100													
BLOS	759													
BLT	1069													
BMI	1079	1084												
BNE	528	547	611	617	620	636	638	641	643	712	738	748	771	827
	862	870	930	951	956	959	985	1004	1018	1065	1108	1494	1515	1517
	1568	1576	1588											857
BPL	562	720	817	836	839	842	1044	1125						1544
BR	541	602	613	622	717	726	744	762	801	834	850	892	967	1081
	1111	1525	1558	1591										1086
CLR	606	626	627	628	775	782	786	788	797	814	840	858	909	915
	945	963	964	965	975	1039	1052	1054	1066	1087	1093	1134	1137	1518
CLRB	671													924
CMP	597	619	642	740	745	758	798	856	869	881	890	893	929	950
	1064	1068	1107	1493	1510	1519	1543	1587						970
CMPB	524	616	711	714	821	823	861							
DEC	637	770	832	1003	1017	1057								
EMT	269	270	271	272	273	274	275	276	277	278	279	280	281	282
HALT	293	294	296	298	300	302	304	308	310	312	314	316	318	320
	324	326	328	330	332	334	336	338	340	342	344	346	348	350
	354	356	358	360	362	364	366	368	370	372	374	376	378	380
	384	386	388	390	392	394	396	398	400	402	404	406	408	410
	414	416	418	420	563	601								412
INC	1058	1063	1067	1516	1557									
INCB	1095													
JMP	433	434	436	593	596	612	618	931	942	1135	1138	1534		
JSR	526	529	544	548	552	569	573	583	584	588	609	621	653	716
	725	732	735	756	873	879	907	911	916	920	925	948	969	1148
	1172	1184	1196	1208	1220	1232	1244	1256	1268	1280	1292	1304	1316	1328
	1352	1364	1376	1388	1400	1412	1424	1436	1448	1460	1472	1484	1496	1500
	1521	1546	1572											1505
MOV	539	540	542	543	581	585	592	594	595	604	605	607	614	630
	646	658	659	660	661	662	663	667	669	670	674	677	678	679
	681	682	683	684	685	692	693	694	695	696	697	698	699	700
	709	739	749	751	766	767	774	776	777	778	779	780	781	783
	794	795	796	813	815	837	844	883	887	888	889	894	895	896
	906	910	932	934	935	936	937	938	939	943	944	946	960	976
	978	979	991	992	993	995	1013	1014	1015	1026	1027	1028	1029	1033
	1040	1045	1053	1055	1056	1077	1078	1082	1090	1091	1099	1101	1109	1113
	1136	1499	1504	1509	1513	1524	1549	1550	1551	1552	1553	1554	1555	1556
	1564	1571	1582	1583	1585	1589	1593							1563
MOVB	710	718	722	724	818	820	859	998	1016	1059	1060	1089	1094	1097
														1570

NOP	1591													
	625	649	650	651	652	654	655	656	691	706	731	802	905	947
RESET	968	986	1025	1034	1051	1075	1098	1115	1119	1548	1561	1562	1594	1598
ROL	608	649	1104											
ROR	1042													
RTI	1000	1001	1002											
RTS	536	565	634	686	701	713	864	884	898	1116	1127	1596	1603	
SEC	521	577	664	721	790	803	845	987	1006	1020	1035	1046	1071	
SUB	1041													
TST	668	1043												
TSTB	561	737	843	940	955	984	1110	1124	1514	1590				
WAIT	719	816	835	838	841									
.ASCII	785													
	1710	1716	1718	1721	1722	1727	1730	1733	1735	1737	1738	1741	1742	1752
.BYTE	1757	1761	1771	1775	1777	1784	1793	1794	1796	1799				
.ENABL	1798	1811	1843	1875	1907	1938	1939							
.END	209													
.EVEN	1940													
.LIST	1810													
.MACR	208	211												
.NLIST	211													
.REM	207	211												
.REPT	3													
.TITLE	1812	1844	1876	1908										
	206													

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*,DZDMBB.SRC/SOL/CRF=DZDMBB.SRC
RUN-TIME: 5 12 2 SECONDS
RUN-TIME RATIO: 120/20=5.7
CORE USED: 8K (15 PAGES)

[illegible]