

BMGR DEC/X11 SYSTEM EXERCISER MODULE
XBMGB0.P11 12-OCT-78 11:51

MACV11 30A(1052) 12-OCT-78 16:21 PAGE 2

.REM

SEQ 0001

IDENTIFICATION

PRODUCT CODE: AC-F069B-MC
PRODUCT NAME: CX8MG80 BM873-YJ MODULE
PRODUCT DATE: SEPTEMBER 1978
MAINTAINER: DEC/X11 SUPPORT GROUP

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE OR EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1976,1978 DIGITAL EQUIPMENT CORPORATION

1.0 ABSTRACT

BMG IS A BACKGROUND MODULE THAT EXERCISES A SINGLE BM873-YJ BOOTSTRAP ROM OPTION. IT COMPARES THE CONTENTS OF EACH OF THE 256(10) LOCATIONS STORED IN THE ROM WITH THE CONTENTS OF A 256(10) WORD CORE MEMORY BUFFER TO VERIFY THAT EACH LOCATION IN THE ROM CAN BE UNIQUELY ADDRESSED AND CONTAINS THE CORRECT DATA. ALL ERRORS ARE REPORTED VIA THE CONSOLE DEVICE.

2.0 REQUIREMENTS

HARDWARE: A PDP11 COMPUTER WITH A BM873-YJ OPTION

STORAGE: BMG REQUIRES:
1. DECIMAL WORDS: 368
2. OCTAL WORDS: 0560
3. OCTAL BYTES: 1340

3.0 PASS DEFINITION

THE INITIAL PASS CONSISTS OF EXECUTING THE BASIC TEST SEQUENCE ONE TIME BEFORE REPORTING END OF PASS. SUBSEQUENT PASSES OF THE BMCR MODULE CONSISTS OF 100(8) ITERATIONS OF THE BASIC TEST SEQUENCE DESCRIBED IN PARA. 7 BELOW.

4.0 EXECUTION TIME

PASS TIME VARIES DEPENDENT UPON CPU TYPE AND THE CONFIGURATION BEING EXERCISED.

5.0 CONFIGURATION OPTIONS

DEFAULT PARAMETERS:

DVA: 173000

REQUIRED PARAMETERS:

NONE

DEVICE OPTION SETUP

NONE REQUIRED

MODULE OPERATION

TEST SEQUENCE:

- 1. R1 IS SET UP TO POINT TO THE FIRST WORD IN THE KOM
- 2. R2 IS SET UP TO POINT TO THE CORRESPONDING WORD IN THE CORE MEMORY BUFFER.

THE ADDRESS IN R1 IS CHECKED FOR EQUALITY TO EITHER 173024 OR 173024 AND IF FOUND EQUAL GOES TO STEP (5) - IF NOT IT PROCEEDS WITH STEP (3). THESE TWO ADDRESSES ARE NOT CHECKED BECAUSE THEIR CONTENTS AS READ ON THE BUS WILL VARY DEPENDENT UPON WHICH PARTICULAR LOGRAM HAD BEEN INITIALLY DEPTED TO LOAD THE PROGRAM. R1 AND R2 ARE USED TO COMPARE A ROW WORD WITH ITS CORE IMAGE COUNTED UP IF THE WORDS DON'T COMPARE A SUB-REPORT LINE IS CALLED TO SET UP THE ERROR INFORMATION AND REPORT VIA A DATA CALL TO THE MONITOR.

STEP (3) IS REPEATED.
 4. TEST MADE ON R2 TO SEE IF 256(10) WORDS HAVE BEEN CHECKED.
 IF YES, GO TO STEP (6) IF NOT REPEAT (3) THRU (5)
 5. A PASS COUNTER IS DECREMENTED AND TESTED TO SEE IF 100(8) ITERATIONS OF STEPS (1) THRU (5) HAVE OCCURRED - IF YES GO TO STEP (7) IF NOT REPEAT (1) THRU (5)
 6. REPORT END OF PASS AND REPEAT (1) THRU (6).

OPERATOR OPTIONS

8-0

(NONE)

NON-STANDARD PRINTOUTS

9-0

(NONE)

```

000000*      BKMOD <BMGB > 173000,0,0,0,0,100,156
000000*      MODULE 40020,8MGB,173000,0,0,0,0,100,156
;      .TITLE BMGB DEC/X11 SYSTEM EXERCISER MODULE
DDXCOM VERSION 6 23-MAY-78
;      .LIST BIN
*****
000000*      BEGIN:
000000* 046502 041107 040      MODNAM: - ASCII /BMGB / ;MODULE NAME
000005*      000      XPLAC: - BYTE OPEN      ;USED TO KEEP TRACK OF WBOFF USAGE
000006* 173000      ADDR: 173000+0      ;1ST DEVICE ADDR.
000010* 000000      VECTOR: 0+0      ;1ST DEVICE VECTOR.
000012* 000      BR1: - BYTE PRTYO+0      ;1ST BR LEVEL.
000013* 000      BR2: - BYTE PRTYO+0      ;2ND BR LEVEL.
000014* 000001      DVID1: 0+1      ;DEVICE INDICATOR 1.
000016* 000000      SR1: OPEN      ;SWITCH REGISTER 1.
000020* 000000      SR2: OPEN      ;SWITCH REGISTER 2.
000022* 000000      SR3: OPEN      ;SWITCH REGISTER 3.
000024* 000000      SR4: OPEN      ;SWITCH REGISTER 4.
*****
000026* 040020      STAT: 40020      ;STATUS WORD.
000030* 000224*      INIT: START      ;MODULE START ADDR.
000032* 000224*      SPOINT: MODSP      ;MODULE STACK POINTER.
000034* 000000      PASCNT: 0      ;PASS COUNTER.
000040* 000000      ICDWT: 100      ;# OF ITERATIONS PER PASS=100
000042* 000000      SOFCMT: 0      ;LOC TO COUNT ITERATIONS
000044* 000000      HRDCMT: 0      ;LOC TO SAVE TOTAL SOFT ERRORS
000046* 000000      SOPPAS: 0      ;LOC TO SAVE TOTAL HARD ERRORS
000050* 000000      HOPPAS: 0      ;LOC TO SAVE SOFT ERRORS PER PASS
000052* 000000      SYSCMT: 0      ;LOC TO SAVE HARD ERRORS PER PASS
000054* 000000      RANWUN: 0      ;# OF SYS ERRORS ACCUMULATED
000056* 000000      COMFIG: 0      ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000060* 000000      RES1: 0      ;RESERVED FOR MONITOR USE
000062* 000000      RES2: 0      ;RESERVED FOR MONITOR USE
000064* 000000      SVR0: OPEN      ;LOC TO SAVE R0.
000066* 000000      SVR1: OPEN      ;LOC TO SAVE R1.
000070* 000000      SVR2: OPEN      ;LOC TO SAVE R2.
000072* 000000      SVR3: OPEN      ;LOC TO SAVE R3.
000074* 000000      SVR4: OPEN      ;LOC TO SAVE R4.
000076* 000000      SVR5: OPEN      ;LOC TO SAVE R5.
000100* 000000      SVR6: OPEN      ;LOC TO SAVE R6.
000102* 000000      CSRA: OPEN      ;ADDR OF CURRENT CSR.
000104* 000000      SBADR: OPEN      ;ADDR OF GOOD DATA, OR
000106* 000000      ACSR: OPEN      ;CONTENTS OF CSR.
000110* 000000      WASADR: OPEN      ;ADDR OF BAD DATA, OR
000112* 000230*      ASTAT: OPEN      ;STATUS REG CONTENTS.
000114* 000000      ERRTYP: OPEN      ;TYPE OF ERROR.
000116* 000000      ASB: OPEN      ;EXPECTED DATA.
000120* 000000      RSTRT: RESTRT      ;RESTART ADDRESS AFTER END OF PASS
000122* 000156*      WDTO: OPEN      ;WORDS TO MEMORY PER ITERATION
000040* 000040*      WDFR: OPEN      ;WORDS FROM MEMORY PER ITERATION
000122* 000156*      ITR: OPEN      ;# OF INTERRUPTS PER ITERATION
000040* 000040*      IDNUM: 156      ;MODULE IDENTIFICATION NUMBER=156
;      .REPT SPSIZ      ;MODULE STACK STARTS HERE.
;      .NLIST

```

```

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

```

```

199
200 000224 016705 177556
201
202 000230
203 000230 010501
204 000232 012702 000336
205 000236 022701 173024
206 000244 001413
207 000244 022701 173224
208 000250 001410
209 000252 021112
210 000254 001404
211 000256 004767 000026
212 000262 021112
213 000266 004767 000016
214 000266 004767
215 000272 022702
216 000274 022702 001336
217 000300 001356
218 000302
219 000302 104413 000000
220
221 000306 000750
222
223
224
225
226 000310 010267 177566
227 000314 010167 177564
228 000320 011167 177564
229 000324 011267 177556
230
231 000330 104404 000000
232
233 000334 000207
234
235
236
237
238
239
240 000336
241 000336 010037
242 000340 000040
243 000344 013700
244 000344 017570
245 000346 032700
246 000350 000001
247 000352 001007
248 000354 000510
249
250 000356 005000
251 000360 000404
252
253 000362 173000
254 000364 000340

```

```

START: MOV ADDR,R5 ;GET FIRST ROW ADDRESS INTO R5
RESTART:
AGAIN: MOV R5,R1 ;R1 POINTS TO ROW WORD
MOV R6,R2 ;R2 POINTS TO ROW IMAGE IN CORF
1$: CMP #173024,R1 ;ROW ADDRESS = 173024 ??
BGE #173224,R1 ;BR IF YES
MOV #173224,R1 ;ROW ADDRESS = 173224 ??
BGE #35 ;BR IF YES
MOV (R1),(R2) ;CHECK ONE LOCATION
BGE #CORE ;BR IF [ROW] = [CORE]
JSR PC,RMERR ;GO SETUP AND REPORT ERROR
CMP (R1),(R2) ;CHECK IT AGAIN
BGE #CORE ;BR IF [ROW] = [CORE]
JSR PC,RMERR ;GO SETUP AND REPORT ERROR
CMP (R1)+(R2)+ ;ADD +2 TO BOTH POINTERS
BNE #1 ;DONE LAST WORD ??
;BR IF NOT

4$: ENDITS,BEGIN ;SIGNAL END OF ITERATION.
BR AGAIN ;MONITOR SHALL TEST END OF PASS

;THIS ROUTINE SETS UP AND REPORTS ALL DATA COMPARE ERRORS
BMERR: MOV R2,SBADR ;SAVE THE ADDR. OF GOOD DATA
MOV R1,WASADR ;SAVE ADDR OF THE BAD DATA
MOV (R1),WAS ;GET WAS DATA
MOV (R2),ASB ;GET THE S/R DATA
;*****
DATERS,BEGIN ;DATA ERROR!!!
RTS PC ;CONTINUE CHECKING

;256(10) WORD TABLE THAT STORES A CORE IMAGE OF THE CONTENTS OF THE ROM
BMTAB:
010037 ;173000 010037 BUTON1: MOV R0,RSSGSV+0 ;SAVE R0 IN LOCA
000040 ;173002 000040
017700 ;173004 013700 MOV @#SWR,R0 ;GET SWITCH REGI
17570 ;173006 177570
032700 ;173010 032700 BIT #BIT0,R0 ;IS LOW-ORDFR BY
000001 ;173012 000001
001007 ;173014 001007 BUTON3: CLR R0 ;SAY LOAD FROM P
000510 ;173016 000510 BR RECSAV ;NO-- SAVE R1-R7
005000 ;173020 005000 BR R0 ;GO TO COMMON CD
000404 ;173022 000404 BR BUTONX
173000 ;173024 173000 FILLTO 24 ;WORD
000340 ;173026 000340

```

```

255 000366 012700
256 000370 000200
257
258 000372 010005
259 000374 105300
260 000376 125300
261 000400 000060
262 000402 101001
263 000404 005000
264
265 000406 000300
266 000410 042700
267 000412 177770
268 000414 105705
269 000416 105551
270 000420 105551
271 000422 173774
272 000424 000004
273 000426 005037
274 000430 000006
275
276 000432 012706
277 000434 000020
278 000436 012701
279 000440 177170
280 000442 005711
281
282 000444 005705
283 000446 100402
284 000450 005306
285 000452 002531
286
287 000454 000005
288
289 000456 032711
290 000460 000040
291 000462 001775
292 000464 111704
293 000466 010002
294 000470 001402
295 000472 012702
296 000474 000020
297
298 000476 052702
299 000500 000007
300 000504 010103
301 000504 010223
302
303 000506 105711
304 000510 100376
305 000512 000001
306 000514 000001
307 000516 005304
308 000520 001372
309
310

```

```

012700 ;173030 012700 BUTON2: MOV #BIT7,R0 ;BIT 7 MEANS LOA
000200 ;173032 000200
;173034
010005 ;173034 010005 BUTONX: MOV R0,R5 ;SAVE PARAMETER
106300 ;173036 106300 ASLR R0 ;LEFT-ALIGN SPEE
122700 ;173038 122700 CMPR #3*BIT4,R0 ;IS SPEED 0, 1,
000060 ;173042 000060
101001 ;173044 101001 BHI R0 ;YES-- UNIT IS U
005000 ;173046 005000 CLR R0 ;NO-- USE UNIT #
;173050
000300 ;173050 000300 10S: SWAB R0 ;GET UNIT # IN L
042700 ;173052 042700 BIC #C7,R0 ;TRM TO 3 BITS
105705 ;173056 105705 TSTB R5 ;WHERE SHOULD WE
100551 ;173058 100551 RPB00T ;BIT 7 = 1 -- BO
012737 ;173062 012737 BMI MOV #TCBOOT,#4 ;BOOT FR
;173064 173274
000004 ;173066 000004
005037 ;173068 005037
000006 ;173072 000006
;173074
012706 ;173074 012706 RXBOOT: MOV #R$STRY,SP ;SET RETRY COUNT
000020 ;173076 000020
012701 ;173078 012701 MOV #R$X11+RXCS,R1 ;ADDRESS CONTROL
177170 ;173102 177170
005711 ;173104 005711 RXRTRY: TST (R1) ;RX11 EXIST?
;173106
005705 ;173106 005705 TST R5 ;INDEFINITE RETR
100402 ;173110 100402 BMI RXRSET ;YES-- TRY FAITH
005306 ;173112 005306 DEC SP ;NO-- DECREMENT
002531 ;173114 002531 BLI RXEHLT ;GIVE UP IF RUN
;173116
000005 ;173116 000005 RXRSET: RESET ;CLEAR THE WORLD
;173120
032711 ;173120 032711 20S: BIT #RXDONE,(R1) ;WAIT UNTIL READ
000040 ;173124 000040
001775 ;173124 001775 BEQ 20S ;NOT YET-- WAIT
111704 ;173126 111704 MOVB (PC),R4 ;SET TRACK/SECTO
010002 ;173130 010002 MOV R0,R2 ;SET UNIT # ***
001402 ;173132 001402 BEQ 30S ;ZERO-- USE ZERO
012702 ;173134 012702 MOV #RXUNIT,R2 ;NON-ZERO-- ASSU
;173136 000020
;173140
052702 ;173140 052702 30S: BIS #RXREAD+RXGO,P2 ;SET READ FUNCTI
000007 ;173142 000007
010103 ;173144 010103 MOV R1,R3 ;COPY ADDRESS OF
010223 ;173146 010223 MOV R2,(R3)+ ;START UNIT #
;173148
105711 ;173150 105711 40S: TSTR (R1) ;READY?
100376 ;173152 100376 BPI 40S ;NO-- WAIT
012713 ;173154 012713 MOV #1,(R3) ;SET SECTOR #, T
;173156 000001
005304 ;173160 005304 DEC R4 ;COUNT DOWN SECT
001372 ;173162 001372 BNE 40S ;TRACK TO SET ST

```

311	000522	032711	032711	173164	032711	BIT	#RXERRIRXDONE,(R1)	;DOME OR ERRO	
312	000523	100040	100040	173166	100040				
313	000524	001775	001775	173170	001775	BEQ	50\$;NO-- WAIT	
314	000525	100745	100745	173172	100745	BMI	RXRTRY	;YES-- ERROR IN	
315	000532	012711	012711	173176	012711	MOV	#RXEMPT+RXGO,(R1)	;START EMPTY	
316	000534	000003	000003	173200	000003				
317									
318	000536	132711	132711	173200	132711	60\$: BITB	#RXTRREQIRXDONE,(R1)	;READY FOR W	
319	000538	000425	000425	173204	000425				
320	000541	001775	001775	173206	001775	BEQ	60\$;NOT READY-- WAI	
321	000544	100147	100147	173210	100147	BPL	CLRPC	;DONE-- GO TO LO	
322	000546	111324	111324	173214	111324	MOVB	(R3),(R4)+	;NOT DONE-- GET	
323	000550	000772	000772	173217	000772	BR	60\$;WAIT FOR NEXT B	
324									
325	000552	005000	005000	173214	005000	BUT0M5:	CLR	R0	;HERE TO START W
326	000554	000425	000425	173216	000425	BR	TCBOTO		;BOOT FROM TAPE
327									
328	000556	000000	000000	173220	000000	BUT0N0:	HALT		;HALT NOW
329	000560	000666	000666	173222	000666	BR	BUT0M1		;BOOT LOOK AT SWR
330						FILLTO	224		
331	000562	173230	173230	173224	173230	WORD	BM#73,PR7		
332	000564	000340	000340	173226	000340				
333									
334	000566	010037	010037	173230	010037	BUT0M4:	MOV	R0,RSSGSV+0	;SAVE R0 IN 40
335	000570	000040	000040	173232	000040				
336	000572	012700	012700	173234	012700		MOV	#RTEDMP,R0	;SET RET
337	000574	173622	173622	173236	173622				
338									
339	000576	010037	010037	173240	010037	REGSAV:	MOV	R0,RSSGSV+16	;SAVE R0 AS PC
340	000600	000056	000056	173242	000056		MOV	#RSSGSV+16,R0	;R0 NOW POINTS T
341	000602	012700	012700	173244	012700				
342	000604	000056	000056	173246	000056		MOV	SP,(R0)	;SAVE SP IN 54
343	000606	010640	010640	173250	010640		MOV	R5,(R0)	;SAVE R5 IN 52
344	000610	010540	010540	173252	010540		MOV	R4,(R0)	;SAVE R4 IN 50
345	000612	010440	010440	173254	010440		MOV	R3,(R0)	;SAVE R3 IN 48
346	000614	010340	010340	173256	010340		MOV	R2,(R0)	;SAVE R2 IN 46
347	000616	010240	010240	173260	010240		MOV	R1,(R0)	;SAVE R1 IN 44
348	000620	010140	010140	173262	010140		MOV	(R0),R0	;RESTORE R0 FROM
349	000622	014000	014000	173264	014000		JMP	GRSSGSV+16	;GO TO SAVED PC
350	000624	000177	000177	173266	000177				
351	000626	004564	004564	173268	004564				
352									
353	000630	005005	005005	173272	005005	TCBOTO:	CLR	R5	;SET SWR PARAMET
354						TCBOOT:	MOV	#DLBOOT,#4	;IN CASE
355	000632	012737	012737	173274	012737	000004			
356	000634	173530	173530	173276	173530				
357	000636	000004	000004	173300	000004				
358	000640	012706	012706	173302	012706		MOV	#RSSTRY,SP	;INIT RETRY COUN
359	000642	000020	000020	173304	000020		MOV	#TSSC11+TCCM,R1	;POINT TO COMMAN
360	000644	012701	012701	173306	012701				
361	000646	177342	177342	173310	177342				
362									
363	000650	105011	105011	173312	105011	TCRTRY:	CLRB	(R1)	;STOP ALL TAPE M
364	000652	005705	005705	173314	005705		TST	R5	;INDEFINITE RETR
365	000654	100402	100402	173316	100402		BMI	10\$;YES-- TRY HARDE
366	000656	005306	005306	173320	005306		DEC	SP	;NO-- DECREMENT

367	000660	002426	002426	173322	002426	10\$: BLT	TCEHLT		;TOO MANY-- GIVF
368									
369	000662	000005	000005	173324	000005	RESET			;CLEAR TC11
370	000664	110061	110061	173326	110061	MOVB	R0,1(R1)		;SELECT PROPER U
371	000666	000001	000001	173330	000001				
372	000670	052711	052711	173332	052711	BIS	#TCREV+TCRNUM+TCGO,(R1)		;START T
373	000672	004003	004003	173336	004003				
374									
375	000674	005711	005711	173336	005711	20\$: TST	(R1)		;ERROR?
376	000676	100376	100376	173340	100376	BPL	20\$;NO-- WAIT FOR E
377	000700	005761	005761	173342	005761	TST	TCST-TCCM(R1)		;END-ZONE UP YET
378	000702	177776	177776	173344	177776				
379	000704	100361	100361	173346	100361				
380	000706	012761	012761	173350	012761	BPL	TCRTRY		;NO-- MUST BE T
381	000710	177400	177400	173352	177400	MOV	#-256,TCWC-TCCM(R1)		;SFT WORD
382	000712	000002	000002	173354	000002				
383	000714	042711	042711	173356	042711	BIC	#TCREV,(R1)		;SET FORWARD MOD
384	000716	004000	004000	173360	004000				
385	000720	000005	000005	173362	000005	MOVB	#TCREAD+TCGO,(R1)		;START READ, F
386	000722	000005	000005	173364	000005				
387									
388	000724	105711	105711	173366	105711	30\$: TSTB	(R1)		;TRANSFER DONE?
389	000726	100376	100376	173370	100376	BPL	30\$;NO-- WAIT SOME
390	000730	005711	005711	173372	005711	TST	(R1)		;YES-- ERROR?
391	000732	100745	100745	173374	100745	BMI	TCRTRY		;YES-- RETRY
392	000734	005007	005007	173376	005007	CLR	PC		;NO-- DONE-- GOT
393									
394									
395									
396									
397									
398	000736	000000	000000	173400	000000	-IIF DF	RSSP04,	RPEHLT:	
399	000740	000776	000776	173402	000776	-IIF DF	RSSX11,	RXEHLT:	
400						-IIF DF	TSSC11,	TCEHLT:	
401	000742	012706	012706	173404	012706	-IIF DF	DSSL11,	DLEHLT:	
402	000744	000020	000020	173406	000020	HALTED:			;DIE
403	000746	012701	012701	173410	012701	BR	HALTED		;STAV DEAD
404	000750	176700	176700	173412	176700	RPBOOT:	MOV	#RSSTRY,SP	;RETRY RETRY TIM
405	000752	012702	012702	173414	012702		MOV	#RSSP04+RPCS1,R1	;ADDRESS
406	000754	004000	004000	173416	004000		MOV	#RPECCI,R2	;SET ECC INHIBIT
407									
408	000756	005705	005705	173420	005705	RPRTY:	TST	R5	;INFINITE RETRY?
409	000760	100402	100402	173422	100402		BMI	RPRSET	;YES-- TRY AGAIN
410	000762	005306	005306	173424	005306		DEC	SP	;RETRY COUNT EXH
411	000764	002764	002764	173426	002764		BLT	RPEHLT	;YES-- GIVE UP
412									
413									
414	000766	000005	000005	173430	000005	RPRSET:	RESET		;ZAPI1
415	000770	110061	110061	173432	110061		MOVB	R0,RPCS2(R1)	;SELECT PROPER U
416	000772	000010	000010	173434	000010				
417	000774	012711	012711	173436	012711		MOV	#RPPRST+RPGO,(R1)	;DO READ-IN P
418	000776	000021	000021	173440	000021		CLR	RPDC(R1)	;SET CYLINDER 0
419	001000	005061	005061	173442	005061		CLR	RPDA(R1)	;TRACK 0, SECTO
420	001002	000034	000034	173444	000034				
421	001004	005061	005061	173446	005061		CLR	RPDA(R1)	;TRACK 0, SECTO
422	001010	050261	050261	173452	050261		BIS	R2,RPOF(R1)	;SET INHIBIT ECC

423	001012	000032	000032	173454	000032				
424	001014	012761	012761	173456	012761	MOV	#-256-,R1PC(R1)	SET UP WORD COU	
425	001016	177400	177400	173460	177400				
426	001018	000007	000007	173462	000007	MOV	#RPRD+RPGO,(R1)	START READ PO	
427	001020	000007	000007	173464	000007				
428	001022	000071	000071	173470	000071	20\$:			
429				173470					
430	001026	105711	105711	173470	105711	ISTB	(R1)	READY?	
431	001030	100376	100376	173472	100376	BPL	R1	NO-- WAIT UNTIL	
432	001032	000020	000020	173474	000020	BIT	#RPFER,RPER1(R1)	FORNAT ERROR?	
433	001034	000020	000020	173476	000020				
434	001036	000014	000014	173478	000014				
435	001040	001402	001402	173502	001402	BEG	30\$	NO-- TRY AGAIN,	
436	001042	052702	052702	173504	052702	BYS	#RPFM22,R2	YES-- TRY FOR 2	
437	001044	010000	010000	173506	010000				
438				173510					
439	001046	032711	032711	173512	032711	30\$:			
440	001050	060000	060000	173514	060000	BIT	#RPTREIRPMPCE,(R1)	TRANSFER OR	
441	001052	001341	001341	173516	001341	BNE	RPRTRY	YES-- ERROR-- T	
442	001054	013000	013000	173518	013000	BIT	#RDATAIRPERR,PPDS(R1)	ATTN OR O	
443	001056	000012	000012	173520	000012				
444	001060	000012	000012	173522	140000				
445	001062	001335	001335	173524	001335	CLRPC:	BNE	RPRTRY	YES-- ERROR-- T
446				173526		CLR	PC	JMP 0	
447	001064	005007	005007	173528	005007	BUTDM6:			
448				173530		DLBOOT:			
449				173530		MOV	#DSSL1+DLRCSR,R1	GET DL11 EXTE	
450	001066	012701	012701	173532	012701	MOV	(PC),SP	SET TEMP STACK	
451	001070	177560	177560	173534	177560	MOV	R4	RESET MEMORY AD	
452	001072	011706	011706	173536	011706	MOV	#DLCHAR,R2	SET ADD	
453	001074	015004	015004	173538	015004				
454	001076	015004	015004	173540	015004				
455	001100	173610	173610	173542	173610				
456				173544					
457				173544		10\$:			
458	001102	004712	004712	173544	004712	CALL	(R2)	GET A CHARACTER	
459	001104	124422	124422	173546	124422	JSE PC,(R2)			
460	001106	000220	000220	173548	000220	CMPB	-(R4),#220	DLR?	
461	001110	001374	001374	173550	001374	BNE	10\$	NO-- KEEP ON LO	
462				173552		CALL	-(R2)	SET TWO BYTES	
463	001112	004742	004742	173554	004742	JSE PC,(R2)			
464	001114	014403	014403	173556	014403	MOV	-(R4),R3	GET BYTE COUNT	
465	001116	042703	042703	173558	042703	BIC	#BIT15IBIT14,R3	CLEAR QSYNC AND	
466	001120	140000	140000	173560	140000				
467				173562		CALL	(R2)	SKIP TWO MOPF R	
468	001122	004712	004712	173564	004712	JSE PC,(R2)			
469				173566		CALL	(R2)	AND TWO MOPF	
470	001124	004712	004712	173568	004712	JSE PC,(R2)			
471	001126	005722	005722	173570	005722	TST	(R2)	SKIP THE CALL (
472				173572		CALL	(R2)	+1 MAKES 5 BYT	
473	001130	004712	004712	173574	004712	JSE PC,(R2)			
474	001132	005004	005004	173576	005004	CLR	R4	RESET BACK TO 0	
475				173578		20\$:			
476				173578		CALL	(R2)	GET A CHARACTER	
477	001134	004712	004712	173580	004712	JSE PC,(R2)			
478	001136	005303	005303	173580	005303	DEC	(R3)	REDUCE COUNT	

479	001140	003375	003375	173602	003375	BCT	20\$	BACK IF MORE	
480	001142	005007	005007	173604	005007	CLR	PC	ELSE GO TO LOAD	
481				173606		CALL	(PC)	GET A BYTE, THF	
482	001144	004717	004717	173610	004717	JSE PC,(PC)			
483				173610		DLCHAR:			
484	001146	105711	105711	173610	105711	ISTB	(R1)	READY WITH A CH	
485	001150	100376	100376	173612	100376	BPL	DLCHAR	NO-- WAIT SOME	
486	001152	116124	116124	173614	116124	MOV	DLRBUF-DLRCSR(R1),(R4)	YES-- S	
487	001154	000002	000002	173616	000002				
488				173618		RETURN			
489	001156	000207	000207	173620	000207	RTS PC		AND RETURN FROM	
490				173622		DTRDMP:			
491	001160	005005	005005	173624	005005	CLR	R5	POINT TO LOCATI	
492	001162	012500	012500	173626	012500	MOV	(R5)+,R0	SAVE LOCATION 0	
493	001164	012500	012500	173628	012500	MOV	(R5)+,R1	AND LOCATION 1	
494	001166	011502	011502	173630	011502	MOV	(R5)+,R2	SAVE 4 IN R2	
495	001170	012725	012725	173632	012725	MOV	(R5),R2	#21\$(R5)+	SET NXM
496	001172	173646	173646	173634	173646				
497	001174	011503	011503	173636	011503	MOV	(R5),R3	SAVE 6 IN R3	
498	001176	005015	005015	173638	005015	CLR	(R5)	SET PS FOR TRAP	
499				173640		20\$:			
500	001200	012704	012704	173642	012704	MOV	#D\$ST20+DLVcnt-DTESIZ,R4	POINT	
501	001202	174340	174340	173644	174340				
502				173646		21\$:			
503	001204	012706	012706	173646	012706	MOV	#S\$SDTE+10,SP	SET STACK TO SA	
504	001206	000140	000140	173650	000140				
505				173652		22\$:			
506	001210	062704	062704	173654	062704	ADD	#DTESIZ,R4	BUMP TO NEXT DT	
507	001212	000040	000040	173656	000040				
508	001214	105704	105704	173658	105704	ISTB	R4	IS THIS THE END	
509	001216	100770	100770	173660	100770	BNI	20\$	YES-- START ALL	
510	001218	032764	032764	173662	032764	BIT	#T011DB,STAT-DLVcnt(R4)	DOORBEL	
511	001222	004000	004000	173664	004000				
512	001224	000034	000034	173666	000034				
513	001226	001770	001770	173670	001770	BEG	22\$	NO-- TRY NEXT D	
514	001230	026417	026417	173672	026417	CMP	T010BC-DLVcnt(R4),(PC)	DOES THI	
515	001232	000014	000014	173674	000014				
516	001234	001365	001365	173676	001365	BNE	22\$	NO-- TRY ANOTHE	
517	001236	010315	010315	173678	010315	MOV	R3,(R5)	RESTORE LOCATIO	
518	001240	010245	010245	173700	010245	MOV	R3,(R5)	RESTORE LOCATIO	
519	001242	010145	010145	173702	010145	MOV	R1,(R5)	RESTORE 2	
520	001244	010045	010045	173704	010045	MOV	R0,(R5)	AND 0	
521	001246	012700	012700	173706	012700	MOV	#S\$SDTE,R0	POINT TO SAVP A	
522	001250	000130	000130	173710	000130				
523				173712		29\$:			
524	001252	012420	012420	173714	012420	MOV	(R4)+(R0)+	SAVE A REGISTER	
525	001254	022700	022700	173716	022700	CMP	#T011DB-DLVcnt+S\$SDTE,R0	FINISH	
526	001256	000156	000156	173718	000156				
527	001260	103374	103374	173720	103374				
528				173722		BHIS	29\$	NO-- SAVE SOME	
529	001262	005724	005724	173724	005724	ADDX	(R4)+	R4 R4 PRINTS +	
530	001264	010401	010401	173726	010401	TST	(R4)+		
531	001266	012700	012700	173728	012700	MOV	R4,R1	SO DOES R1	
532	001270	000100	000100	173730	000100	MOV	#D\$RSET,R0	SETUP R0 FOR 'D	
533	001272	010021	010021	173732	010021	MOV	R0,(R1)	R1 POINTS TO ST	
534	001274	005061	005061	173734	005061	CLR	DLVcnt-STAT(R1)	SET DTE20 FOR M	

535 001276 177744
536 001300 005061
537 001302 177764
538
539 001304 032711
540 001306 004000
541 001310 001775
542 001312 010014
543 001314 005061
544 001316 177766
545 001320 017761
546 001322 017400
547 001324 177762
548
549 001326 105711
550 001330 100376
551 001332 005007
552
553 001334 000000
554
555 001336 177777
556
557 000001

177744 173740 177744
005061 173742
177764 173744 177764 005061
173746
032711 173746
004000 173750 004000 032711
001775 173752
010014 173754 001775 010014
005061 173756 005061
177766 173760 177766 012761
017761 173762
107400 173764 107400
177762 173766 177762
173770
105711 173770 105711
100376 173772 100376
005007 173774 005007
173776
000000 173778 000
173779 000
TABEND: 177777

CLR TOLOAD-STAT(R1) ;START DUMPING -
30\$: BIT #TO11DB,(R1) ;I> DOORBELL RIN
BEG 30\$;HI-- WAIT FOR D
MOV 80,(R4) ;VLS-- CLEAR DDB
CLR TOLOAD-STAT(R1) ;S*ART INPUT TO
MOV #IFLOP1<<-256.><BCOUNT>,TO11BC-S
40\$: TSTR (R1) ;TRANSFER COMPLE
BPL 40\$;NO-- WAIT SOME
CLR PC ;GO TO LOADED CO
FILL TO 1000
-BYTE 0
-BITE 0

-END

ACSR 000102R 181#
ADDR 000006R 147# 200
ADDR22= 001000 199#
AERIN 000230R 203# 221
ASB 000106R 185# 229*
ASTAT 000104R 183#
AWAS 000110R 186# 228*
BEGIN 000000R 144# 219
BIT0 = 000001 199#
BIT1 = 000000 199#
BIT10 = 002000 199#
BIT11 = 004000 199#
BIT12 = 010000 199#
BIT13 = 020000 199#
BIT14 = 040000 199#
BIT15 = 100000 199#
BIT2 = 000004 199#
BIT3 = 000010 199#
BIT4 = 000020 199#
BIT5 = 000040 199#
BIT6 = 000100 199#
BIT7 = 000200 199#
BIT8 = 000400 199#
BIT9 = 001000 199#
BMERR 000310R 211# 214 226#
BMTAB 000336R 204# 240#
BREARS= 104407 199#
BP1 000012R 149#
BP2 000013R 150#
BTODS = 104421 199#
CDATAS= 104412 199#
CONFIG 000056R 163#
CSRA 000109R 179#
DATCKS= 104410 199#
DATERS= 104404 199# 231
DVID1 000014R 151#
ENDITS = 104413 199# 219
ENDS = 104419 199#
ERTYP 000106R 184#
EXITS = 104400 199#
GETPAS= 104415 199#
GWBOPS= 104414 199#
HROCHY 000044R 164#
HROERS 104405 178#
HROPAS 000050R 166#
ICONT 000036R 161#
ICOUNT 000040R 162#
IDNPH 000122R 191#
INIT 000030R 158#
ITR 000120R 190#
MAP22S= 104416 199#
MODNAM 000000R 145#
MODSP 000224R 159# 197#
MSGNS = 104403 199#
MSGSS = 104402 199#
MSGSS = 104401 199#

NULL = 000000	199#																		
OPEN = 000000	146																		
UPOAS = 104420	181	152	153	154	155	172	173	174	175	176	177	178	179						
PASCNT = 000034R	183	185	186	188	189	190	199#												
PIRQS = 000004	199#																		
POPS = 005726	199#																		
POPS2 = 022626	199#																		
PRTV = 000000	146																		
PRTV0 = 000000	146																		
PRTV1 = 000040	199#	150	199#																
PRTV2 = 000100	199#																		
PRTV3 = 000140	199#																		
PRTV4 = 000200	199#																		
PRTV5 = 000240	199#																		
PRTV6 = 000300	199#																		
PRTV7 = 000340	199#																		
PS = 177776	199#																		
PSW = 177776	199#																		
PUSH = 005746	199#																		
PUSH2 = 024646	199#																		
RANDS = 104417	199#																		
RANNUM = 000054R	168#																		
RFSRT = 000230R	187	202#																	
RES1 = 000056R	170#																		
RES2 = 000060R	171																		
RSTRT = 000112R	180#																		
SBADR = 000102R	180#	226*																	
SOPCNT = 000042R	163#																		
SOPFRS = 104406	199#																		
SOPFAS = 000046R	165#																		
SPOINT = 000042R	159#																		
SPSIZ = 000040	192																		
SR1 = 000016R	152#																		
SR2 = 000020R	153#																		
SR3 = 000022R	154#																		
SR4 = 000024R	155#																		
START = 000024R	159#	200#																	
STAT = 000076R	177#																		
SVR0 = 000062R	172#																		
SVR1 = 000064R	173#																		
SVR2 = 000066R	174#																		
SVR3 = 000070R	175#																		
SVR4 = 000072R	176#																		
SVR5 = 000074R	177#																		
SVR6 = 000076R	178#																		
SVSCNT = 000052R	167#																		
TABEND = 001336R	216	555#																	
TROFD = 000022	199#																		
VECTOR = 000010R	148#																		
WASADR = 000104R	182#	227*																	
WDFR = 000116R	189#																		
WDTN = 000114R	188#																		
XFLAG = 000005R	146#																		

. ARS. 000000 000
 001340 001

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

XRMGR0,XRMGR0/SOL/CRF:SYM=DDXCON,XRMGR0
 RUN-TIME: 1.1 SECONDS
 RUN-TIME RATIO: 38/3=10.2
 CORE USED: 7K (13 PAGES)