

DH11

MULTI-LINE PARITY
MD-11-DZDHG-B

EP-DZDHG-B-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN USA

This microfiche card contains a grid of frames. The first column on the left contains frames with vertical bars, likely representing parity or control data. The remaining columns contain frames with alphanumeric characters and symbols, representing the main data. The data is organized in a structured, grid-like format typical of microfiche storage.

000000
000001
000002
000003
000004
000005
000006
000007

177570
177570
177776
013564

005746
005726
010046
012600
024646
022626

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

;REGISTER DEFINITIONS

```

000000      RD=%0          ;GENERAL REGISTER
000001      R1=%1          ;GENERAL REGISTER
000002      R2=%2          ;GENERAL REGISTER
000003      R3=%3          ;GENERAL REGISTER
000004      R4=%4          ;GENERAL REGISTER
000005      R5=%5          ;GENERAL REGISTER
000006      SP=%6         ;PROCESSOR STACK POINTER
000007      PC=%7         ;PROGRAM COUNTER
    
```

;LOCATION EQUIVALENCIES

```

177570      SWR=177570    ;CONSOLE SWITCH REGISTER
177570      LIGHTS=177570 ;PDP-11/45 DISPLAY REGISTER
177776      PS=177776    ;PROCESSOR STATUS WORD
013564      STACK=ENDCOD+200;START OF PROCESSOR STACK
    
```

;INSTRUCTION DEFINITIONS

```

005746      PUSH1SP=5746  ;DECREMENT PROCESSOR STACK 1 WORD
005726      POP1SP=5726   ;INCREMENT PROCESSOR STACK 1 WORD
010046      PUSHRO=10046  ;SAVE R0 ON STACK
012600      POPRO=12600   ;RESTORE R0 FROM STACK
024646      PUSH2SP=24646 ;DECREMENT STACK TWICE
022626      POP2SP=22626  ;INCREMENT STACK TWICE
.EQUIV EMT,HLT ;BASIC DEFINITION OF ERROR CALL
    
```

```

BIT15=100000
BIT14=40000
BIT13=20000
BIT12=10000
BIT11=4000
BIT10=2000
BIT09=1000
BIT08=400
BIT07=200
BIT06=100
BIT05=40
BIT04=20
BIT03=10
BIT02=4
BIT01=2
BIT00=1
    
```



```

86                                     ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
87                                     . = 0
88 000000 000002                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
89 000002 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
90 000004 000006                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
91 000006 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
92 000010 000012                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
93 000012 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
94 000014 000016                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
95 000016 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
96 000020 000022                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
97 000022 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
98 000024 000026                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
99 000026 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
100 000030 000032                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
101 000032 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
102 000034 000036                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
103 000036 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
104 000040 000042                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
105 000042 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
106 000044 000046                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
107 000046 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
108 000050 000052                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
109 000052 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
110 000054 000056                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
111 000056 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
112 000060 000062                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
113 000062 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
114 000064 000066                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
115 000066 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
116 000070 000072                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
117 000072 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
118 000074 000076                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
119 000076 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
120 000100 000102                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
121 000102 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
122 000104 000106                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
123 000106 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
124 000110 000112                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
125 000112 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
126 000114 000116                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
127 000116 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
128 000120 000122                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
129 000122 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
130 000124 000126                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
131 000126 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
132 000130 000132                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
133 000132 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
134 000134 000136                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
135 000136 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
136 000140 000142                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
137 000142 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
138 000144 000146                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
139 000146 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE
140 000150 000152                      .+2      ;UNEXPECTED TRAP TO THIS LOCATION
141 000152 000000                      HALT     ;EXAMINE STACK TO FIND CAUSE

```


E01

142	000154	000156	.+2	:UNEXPECTED TRAP TO THIS LOCATION
143	000156	000000	HALT	:EXAMINE STACK TO FIND CAUSE
144	000160	000162	.+2	:UNEXPECTED TRAP TO THIS LOCATION
145	000162	000000	HALT	:EXAMINE STACK TO FIND CAUSE
146	000164	000166	.+2	:UNEXPECTED TRAP TO THIS LOCATION
147	000166	000000	HALT	:EXAMINE STACK TO FIND CAUSE
148	000170	000172	.+2	:UNEXPECTED TRAP TO THIS LOCATION
149	000172	000000	HALT	:EXAMINE STACK TO FIND CAUSE
150	000174	000176	.+2	:UNEXPECTED TRAP TO THIS LOCATION
151	000176	000000	HALT	:EXAMINE STACK TO FIND CAUSE
152	000200	000202	.+2	:UNEXPECTED TRAP TO THIS LOCATION
153	000202	000000	HALT	:EXAMINE STACK TO FIND CAUSE
154	000204	000206	.+2	:UNEXPECTED TRAP TO THIS LOCATION
155	000206	000000	HALT	:EXAMINE STACK TO FIND CAUSE
156	000210	000212	.+2	:UNEXPECTED TRAP TO THIS LOCATION
157	000212	000000	HALT	:EXAMINE STACK TO FIND CAUSE
158	000214	000216	.+2	:UNEXPECTED TRAP TO THIS LOCATION
159	000216	000000	HALT	:EXAMINE STACK TO FIND CAUSE
160	000220	000222	.+2	:UNEXPECTED TRAP TO THIS LOCATION
161	000222	000000	HALT	:EXAMINE STACK TO FIND CAUSE
162	000224	000226	.+2	:UNEXPECTED TRAP TO THIS LOCATION
163	000226	000000	HALT	:EXAMINE STACK TO FIND CAUSE
164	000230	000232	.+2	:UNEXPECTED TRAP TO THIS LOCATION
165	000232	000000	HALT	:EXAMINE STACK TO FIND CAUSE
166	000234	000236	.+2	:UNEXPECTED TRAP TO THIS LOCATION
167	000236	000000	HALT	:EXAMINE STACK TO FIND CAUSE
168	000240	000242	.+2	:UNEXPECTED TRAP TO THIS LOCATION
169	000242	000000	HALT	:EXAMINE STACK TO FIND CAUSE
170	000244	000246	.+2	:UNEXPECTED TRAP TO THIS LOCATION
171	000246	000000	HALT	:EXAMINE STACK TO FIND CAUSE
172	000250	000252	.+2	:UNEXPECTED TRAP TO THIS LOCATION
173	000252	000000	HALT	:EXAMINE STACK TO FIND CAUSE
174	000254	000256	.+2	:UNEXPECTED TRAP TO THIS LOCATION
175	000256	000000	HALT	:EXAMINE STACK TO FIND CAUSE
176	000260	000262	.+2	:UNEXPECTED TRAP TO THIS LOCATION
177	000262	000000	HALT	:EXAMINE STACK TO FIND CAUSE
178	000264	000266	.+2	:UNEXPECTED TRAP TO THIS LOCATION
179	000266	000000	HALT	:EXAMINE STACK TO FIND CAUSE
180	000270	000272	.+2	:UNEXPECTED TRAP TO THIS LOCATION
181	000272	000000	HALT	:EXAMINE STACK TO FIND CAUSE
182	000274	000276	.+2	:UNEXPECTED TRAP TO THIS LOCATION
183	000276	000000	HALT	:EXAMINE STACK TO FIND CAUSE
184	000300	000302	.+2	:UNEXPECTED TRAP TO THIS LOCATION
185	000302	000000	HALT	:EXAMINE STACK TO FIND CAUSE
186	000304	000306	.+2	:UNEXPECTED TRAP TO THIS LOCATION
187	000306	000000	HALT	:EXAMINE STACK TO FIND CAUSE
188	000310	000312	.+2	:UNEXPECTED TRAP TO THIS LOCATION
189	000312	000000	HALT	:EXAMINE STACK TO FIND CAUSE
190	000314	000316	.+2	:UNEXPECTED TRAP TO THIS LOCATION
191	000316	000000	HALT	:EXAMINE STACK TO FIND CAUSE
192	000320	000322	.+2	:UNEXPECTED TRAP TO THIS LOCATION
193	000322	000000	HALT	:EXAMINE STACK TO FIND CAUSE
194	000324	000326	.+2	:UNEXPECTED TRAP TO THIS LOCATION
195	000326	000000	HALT	:EXAMINE STACK TO FIND CAUSE
196	000330	000332	.+2	:UNEXPECTED TRAP TO THIS LOCATION
197	000332	000000	HALT	:EXAMINE STACK TO FIND CAUSE

F01

198	000334	000336	.+2	:UNEXPECTED TRAP TO THIS LOCATION
199	000336	000000	HALT	:EXAMINE STACK TO FIND CAUSE
200	000340	000342	.+2	:UNEXPECTED TRAP TO THIS LOCATION
201	000342	000000	HALT	:EXAMINE STACK TO FIND CAUSE
202	000344	000346	.+2	:UNEXPECTED TRAP TO THIS LOCATION
203	000346	000000	HALT	:EXAMINE STACK TO FIND CAUSE
204	000350	000352	.+2	:UNEXPECTED TRAP TO THIS LOCATION
205	000352	000000	HALT	:EXAMINE STACK TO FIND CAUSE
206	000354	000356	.+2	:UNEXPECTED TRAP TO THIS LOCATION
207	000356	000000	HALT	:EXAMINE STACK TO FIND CAUSE
208	000360	000362	.+2	:UNEXPECTED TRAP TO THIS LOCATION
209	000362	000000	HALT	:EXAMINE STACK TO FIND CAUSE
210	000364	000366	.+2	:UNEXPECTED TRAP TO THIS LOCATION
211	000366	000000	HALT	:EXAMINE STACK TO FIND CAUSE
212	000370	000372	.+2	:UNEXPECTED TRAP TO THIS LOCATION
213	000372	000000	HALT	:EXAMINE STACK TO FIND CAUSE
214	000374	000376	.+2	:UNEXPECTED TRAP TO THIS LOCATION
215	000376	000000	HALT	:EXAMINE STACK TO FIND CAUSE
216	000400	000402	.+2	:UNEXPECTED TRAP TO THIS LOCATION
217	000402	000000	HALT	:EXAMINE STACK TO FIND CAUSE
218	000404	000406	.+2	:UNEXPECTED TRAP TO THIS LOCATION
219	000406	000000	HALT	:EXAMINE STACK TO FIND CAUSE
220	000410	000412	.+2	:UNEXPECTED TRAP TO THIS LOCATION
221	000412	000000	HALT	:EXAMINE STACK TO FIND CAUSE
222	000414	000416	.+2	:UNEXPECTED TRAP TO THIS LOCATION
223	000416	000000	HALT	:EXAMINE STACK TO FIND CAUSE
224	000420	000422	.+2	:UNEXPECTED TRAP TO THIS LOCATION
225	000422	000000	HALT	:EXAMINE STACK TO FIND CAUSE
226	000424	000426	.+2	:UNEXPECTED TRAP TO THIS LOCATION
227	000426	000000	HALT	:EXAMINE STACK TO FIND CAUSE
228	000430	000432	.+2	:UNEXPECTED TRAP TO THIS LOCATION
229	000432	000000	HALT	:EXAMINE STACK TO FIND CAUSE
230	000434	000436	.+2	:UNEXPECTED TRAP TO THIS LOCATION
231	000436	000000	HALT	:EXAMINE STACK TO FIND CAUSE
232	000440	000442	.+2	:UNEXPECTED TRAP TO THIS LOCATION
233	000442	000000	HALT	:EXAMINE STACK TO FIND CAUSE
234	000444	000446	.+2	:UNEXPECTED TRAP TO THIS LOCATION
235	000446	000000	HALT	:EXAMINE STACK TO FIND CAUSE
236	000450	000452	.+2	:UNEXPECTED TRAP TO THIS LOCATION
237	000452	000000	HALT	:EXAMINE STACK TO FIND CAUSE
238	000454	000456	.+2	:UNEXPECTED TRAP TO THIS LOCATION
239	000456	000000	HALT	:EXAMINE STACK TO FIND CAUSE
240	000460	000462	.+2	:UNEXPECTED TRAP TO THIS LOCATION
241	000462	000000	HALT	:EXAMINE STACK TO FIND CAUSE
242	000464	000466	.+2	:UNEXPECTED TRAP TO THIS LOCATION
243	000466	000000	HALT	:EXAMINE STACK TO FIND CAUSE
244	000470	000472	.+2	:UNEXPECTED TRAP TO THIS LOCATION
245	000472	000000	HALT	:EXAMINE STACK TO FIND CAUSE
246	000474	000476	.+2	:UNEXPECTED TRAP TO THIS LOCATION
247	000476	000000	HALT	:EXAMINE STACK TO FIND CAUSE
248	000500	000502	.+2	:UNEXPECTED TRAP TO THIS LOCATION
249	000502	000000	HALT	:EXAMINE STACK TO FIND CAUSE
250	000504	000506	.+2	:UNEXPECTED TRAP TO THIS LOCATION
251	000506	000000	HALT	:EXAMINE STACK TO FIND CAUSE
252	000510	000512	.+2	:UNEXPECTED TRAP TO THIS LOCATION
253	000512	000000	HALT	:EXAMINE STACK TO FIND CAUSE

GO1

254	000514	000516	.+2	:UNEXPECTED TRAP TO THIS LOCATION
255	000516	000000	HALT	:EXAMINE STACK TO FIND CAUSE
256	000520	000522	.+2	:UNEXPECTED TRAP TO THIS LOCATION
257	000522	000000	HALT	:EXAMINE STACK TO FIND CAUSE
258	000524	000526	.+2	:UNEXPECTED TRAP TO THIS LOCATION
259	000526	000000	HALT	:EXAMINE STACK TO FIND CAUSE
260	000530	000532	.+2	:UNEXPECTED TRAP TO THIS LOCATION
261	000532	000000	HALT	:EXAMINE STACK TO FIND CAUSE
262	000534	000536	.+2	:UNEXPECTED TRAP TO THIS LOCATION
263	000536	000000	HALT	:EXAMINE STACK TO FIND CAUSE
264	000540	000542	.+2	:UNEXPECTED TRAP TO THIS LOCATION
265	000542	000000	HALT	:EXAMINE STACK TO FIND CAUSE
266	000544	000546	.+2	:UNEXPECTED TRAP TO THIS LOCATION
267	000546	000000	HALT	:EXAMINE STACK TO FIND CAUSE
268	000550	000552	.+2	:UNEXPECTED TRAP TO THIS LOCATION
269	000552	000000	HALT	:EXAMINE STACK TO FIND CAUSE
270	000554	000556	.+2	:UNEXPECTED TRAP TO THIS LOCATION
271	000556	000000	HALT	:EXAMINE STACK TO FIND CAUSE
272	000560	000562	.+2	:UNEXPECTED TRAP TO THIS LOCATION
273	000562	000000	HALT	:EXAMINE STACK TO FIND CAUSE
274	000564	000566	.+2	:UNEXPECTED TRAP TO THIS LOCATION
275	000566	000000	HALT	:EXAMINE STACK TO FIND CAUSE
276	000570	000572	.+2	:UNEXPECTED TRAP TO THIS LOCATION
277	000572	000000	HALT	:EXAMINE STACK TO FIND CAUSE
278	000574	000576	.+2	:UNEXPECTED TRAP TO THIS LOCATION
279	000576	000000	HALT	:EXAMINE STACK TO FIND CAUSE
280	000600	000602	.+2	:UNEXPECTED TRAP TO THIS LOCATION
281	000602	000000	HALT	:EXAMINE STACK TO FIND CAUSE
282	000604	000606	.+2	:UNEXPECTED TRAP TO THIS LOCATION
283	000606	000000	HALT	:EXAMINE STACK TO FIND CAUSE
284	000610	000612	.+2	:UNEXPECTED TRAP TO THIS LOCATION
285	000612	000000	HALT	:EXAMINE STACK TO FIND CAUSE
286	000614	000616	.+2	:UNEXPECTED TRAP TO THIS LOCATION
287	000616	000000	HALT	:EXAMINE STACK TO FIND CAUSE
288	000620	000622	.+2	:UNEXPECTED TRAP TO THIS LOCATION
289	000622	000000	HALT	:EXAMINE STACK TO FIND CAUSE
290	000624	000626	.+2	:UNEXPECTED TRAP TO THIS LOCATION
291	000626	000000	HALT	:EXAMINE STACK TO FIND CAUSE
292	000630	000632	.+2	:UNEXPECTED TRAP TO THIS LOCATION
293	000632	000000	HALT	:EXAMINE STACK TO FIND CAUSE
294	000634	000636	.+2	:UNEXPECTED TRAP TO THIS LOCATION
295	000636	000000	HALT	:EXAMINE STACK TO FIND CAUSE
296	000640	000642	.+2	:UNEXPECTED TRAP TO THIS LOCATION
297	000642	000000	HALT	:EXAMINE STACK TO FIND CAUSE
298	000644	000646	.+2	:UNEXPECTED TRAP TO THIS LOCATION
299	000646	000000	HALT	:EXAMINE STACK TO FIND CAUSE
300	000650	000652	.+2	:UNEXPECTED TRAP TO THIS LOCATION
301	000652	000000	HALT	:EXAMINE STACK TO FIND CAUSE
302	000654	000656	.+2	:UNEXPECTED TRAP TO THIS LOCATION
303	000656	000000	HALT	:EXAMINE STACK TO FIND CAUSE
304	000660	000662	.+2	:UNEXPECTED TRAP TO THIS LOCATION
305	000662	000000	HALT	:EXAMINE STACK TO FIND CAUSE
306	000664	000666	.+2	:UNEXPECTED TRAP TO THIS LOCATION
307	000666	000000	HALT	:EXAMINE STACK TO FIND CAUSE
308	000670	000672	.+2	:UNEXPECTED TRAP TO THIS LOCATION
309	000672	000000	HALT	:EXAMINE STACK TO FIND CAUSE

310	000674	000676	.+2	:UNEXPECTED TRAP TO THIS LOCATION
311	000676	000000	HALT	:EXAMINE STACK TO FIND CAUSE
312	000700	000702	.+2	:UNEXPECTED TRAP TO THIS LOCATION
313	000702	000000	HALT	:EXAMINE STACK TO FIND CAUSE
314	000704	000706	.+2	:UNEXPECTED TRAP TO THIS LOCATION
315	000706	000000	HALT	:EXAMINE STACK TO FIND CAUSE
316	000710	000712	.+2	:UNEXPECTED TRAP TO THIS LOCATION
317	000712	000000	HALT	:EXAMINE STACK TO FIND CAUSE
318	000714	000716	.+2	:UNEXPECTED TRAP TO THIS LOCATION
319	000716	000000	HALT	:EXAMINE STACK TO FIND CAUSE
320	000720	000722	.+2	:UNEXPECTED TRAP TO THIS LOCATION
321	000722	000000	HALT	:EXAMINE STACK TO FIND CAUSE
322	000724	000726	.+2	:UNEXPECTED TRAP TO THIS LOCATION
323	000726	000000	HALT	:EXAMINE STACK TO FIND CAUSE
324	000730	000732	.+2	:UNEXPECTED TRAP TO THIS LOCATION
325	000732	000000	HALT	:EXAMINE STACK TO FIND CAUSE
326	000734	000736	.+2	:UNEXPECTED TRAP TO THIS LOCATION
327	000736	000000	HALT	:EXAMINE STACK TO FIND CAUSE
328	000740	000742	.+2	:UNEXPECTED TRAP TO THIS LOCATION
329	000742	000000	HALT	:EXAMINE STACK TO FIND CAUSE
330	000744	000746	.+2	:UNEXPECTED TRAP TO THIS LOCATION
331	000746	000000	HALT	:EXAMINE STACK TO FIND CAUSE
332	000750	000752	.+2	:UNEXPECTED TRAP TO THIS LOCATION
333	000752	000000	HALT	:EXAMINE STACK TO FIND CAUSE
334	000754	000756	.+2	:UNEXPECTED TRAP TO THIS LOCATION
335	000756	000000	HALT	:EXAMINE STACK TO FIND CAUSE
336	000760	000762	.+2	:UNEXPECTED TRAP TO THIS LOCATION
337	000762	000000	HALT	:EXAMINE STACK TO FIND CAUSE
338	000764	000766	.+2	:UNEXPECTED TRAP TO THIS LOCATION
339	000766	000000	HALT	:EXAMINE STACK TO FIND CAUSE
340	000770	000772	.+2	:UNEXPECTED TRAP TO THIS LOCATION
341	000772	000000	HALT	:EXAMINE STACK TO FIND CAUSE
342	000774	000776	.+2	:UNEXPECTED TRAP TO THIS LOCATION
343	000776	000000	HALT	:EXAMINE STACK TO FIND CAUSE


```

344                                     ;STANDARD INTERRUPT VECTORS
345
346
347                                     . =24
348 000024 012354 PFAIL                ;POWER FAIL HANDLER
349 000026 000340 340                  ;SERVICE AT LEVEL 7
350 000030 010552 ERRORS              ;ERROR HANDLER
351 000032 000340 340                  ;SERVICE AT LEVEL 7
352 000034 010754 TRPSRV             ;GENERAL HANDLER DISPATCH SERVICE
353 000036 000340 340                  ;SERVICE AT LEVEL 7
354                                     . =200
355 000200 000167 000574 JMP START    ;GO TO START OF PROGRAM
356
357
358
359                                     ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
360                                     ;POINTERS TO SUBROUTINES CAN BE FOUND STARTING
361                                     ;AT LOCATION "TRPTAB"
362
363 104400 SCOPE=TRAP+Y                ;SCOPE LOOP AND ITERATION HANDLER
364 104401 TYPE=TRAP+Y                ;TELETYPE OUTPUT ROUTINE
365 104402 OCTASC=TRAP+Y              ;OCTAL TO ASCII CONVERSION
366 104403 INSTR=TRAP+Y              ;INPUT ASCII STRING
367 104404 INSTER=TRAP+Y             ;STRING INPUT ERROR
368 104405 PARAM=TRAP+Y              ;CONVERT STRING TO OCTAL, CHECK LIMITS
369 104406 SAVOSP=TRAP+Y             ;SAVE R0-R5, PC
370 104407 RESO5=TRAP+Y             ;RESTORE R0-R5
371 104410 SCOPE1=TRAP+Y             ;CHECK FOR FREEZE ON CURRENT DATA

```



```

372          001000          . =1000
373
374          ;PROGRAM INITIALIZATION
375          ;LOCK OUT INTERRUPTS
376          ;SET UP PROCESSOR STACK
377          ;SET UP POWER FAIL VECTOR
378          ;CLEAR PROGRAM FLAGS AND COUNTS
379          ;TYPE TITLE MESSAGE
380
381 001000 012767 000340 176770 START: MOV #340,PS          ;LOCK OUT INTERRUPTS
382 001006 012706 013564          MOV #STACK,SP        ;SET UP PROCESSOR STACK
383 001012 012737 012354 000024 MOV #PFAIL,2#24      ;SET UP POWER FAIL TRAP
384 001020 005067 010656          CLR STFLG           ;CLEAR TEST START FLAG
385 001024 005067 010612          CLR PASCNT         ;CLEAR PASS COUNT
386 001030 005067 010610          CLR ERRCNT         ;CLEAR ERROR COUNT
387 001034 005067 010600          CLR ERRFLG        ;CLEAR ERROR FLAG
388 001040 005067 010574          CLR ERRFLG        ;CLEAR LAST ERROR PC
389 001044 104401 012520          TYPE ,MTITLE      ;TYPE TITLE MESSAGE
390 001050 005767 010624          TST INIFLG        ;CHECK INITIALIZATION FLAG
391 001054 001001          BNE VEC1          ;IF NOT 0, CHECK SWITCHES
392          ;FOR REINITIALIZATION
393 001056 000404          BR VEC2
394 001060 032767 000001 176502 VEC1: BIT #SW00,SWR        ;IF SW00=1, GET NEW VECTOR
395 001066 001445          BEQ BEGIN         ;AND CSR
396 001070 012701 000300          VEC2: MOV #300,R1
397 001074 012702 000302          MOV #302,R2
398 001100 012703 000004          MOV #4,R3
399 001104 010211          IS: MOV R2,(R1)      ;RESTORE TRAPCATCHER
400 001106 005012          CLR (R2)         ;IN FLOATING VECTOR AREA
401 001110 060301          ADD R3,R1
402 001112 060302          ADD R3,R2
403 001114 020127 001000          CMP R1,#1000
404 001120 001371          BNE IS
405 001122 104403          INSTR           ;INPUT ADDRESS OF DEVICE VECTOR
406 001124 012613          MVECTOR        ;MESSAGE "VECTOR ADDRESS--"
407 001126 104405          PARAM         ;CONVERT STRING TO OCTAL
408 001130 000300          300           ;LOW LIMIT
409 001132 000770          770           ;HIGH LIMIT
410 001134 011630          DHRVEC        ;LOCATIONS TO BE FILLED
411 001136          003         ;NUMBER OF LOCATIONS
412 001137          004         ;LSB MASK
413 001140 104403          INSTR         ;INPUT ADDRESS OF DEVICE CSR
414 001142 012635          MREGAD        ;MESSAGE "CONTROL REGISTER ADDRESS--"
415 001144 104405          PARAM         ;CONVERT STRING TO OCTAL
416 001146 000000          0            ;LOW LIMIT
417 001150 177776          177776       ;HIGH LIMIT
418 001152 011606          DHSCR        ;LOCATIONS TO BE FILLED
419 001154          007         ;NUMBER OF LOCATIONS
420 001155          010        ;LSB MASK
421 001156 016767 010442 010442 .BYTE MOV DHSSR,DHSLR  ;SET UP ADDRESS OF SILO
422 001164 005267 010436          INC DHSLR       ;STATUS REGISTER HIGH BYTE
423 001170 005767 010504          TST INIFLG     ;IF INITIALIZATION FLAG
424 001174 001002          BNE BEGIN     ;IS CLEARED
425 001176 005167 010476          COM INIFLG     ;SET IT
426
427          ;PROGRAM START

```


K01

DZDHG MACY11 27(732) 28-SEP-76 15:16 PAGE 11
 DZDHGB.P11

```

428                                     ;CHECK FOR PROGRAM START AT SELECTED ADDRESS
429
430 001202 012767 000340 176566 BEGIN: MOV #340,PS ;LOCK OUT INTERRUPTS
431 001210 012706 013564 MOV #STACK,SP ;SET UP PROCESSOR STACK
432 001214 032767 000002 176346 BIT #SW01,SWR ;IF SW01=1
433 001222 001410 BEQ 1$ ;GET PC FOR PROGRAM START
434 001224 104403 INSTR ;GET PC
435 001226 013001 MTSTPC ;MESSAGE "TEST PC"
436 001230 104405 PARAM ;CONVERT STRING TO OCTAL
437 001232 000000 0
438 001234 017500 17500
439 001236 000207 RETURN
440 001240 001 .BYTE 1
441 001241 001 .BYTE 1
442 001242 000410 BR 2$
443 001244 012767 001274 010374 1$: MOV #T1,RETURN ;NORMAL START, TEST 1
444 001252 005767 010424 TST STFLG ;IF LOOPING, BYPASS TYPEOUT
445 001256 001004 BNE 3$
446 001260 005167 010416 COM STFLG
447 001264 104401 012775 2$: TYPE MR ;TYPE "R" TO INDICATE START
448 001270 000177 010352 3$: JMP @RETURN ;START TESTING
  
```

L01

```

449
450 ;RECEIVER PARITY LOGIC TEST ON LINE 0.
451 ;TRANSMIT ONE CHARACTER TO THE RECEIVER
452 ;WITH ODD PARITY SET,BREAK SET,HDX SET.
453 ;THE RECEIVER WILL RECEIVE AN ALL 0'S CHARACTER
454 ;WITH EVEN PARITY AND THEREBY
455 ;CAUSE A PARITY ERROR AND OVERRUN
456 ;CHARACTER LENGTH: 8 BITS
457 ;LINE SPEED: 9600 BAUD
458
459 001274 012767 000340 176474 T1: MOV #340,PS ;DISABLE ALL INTERRUPTS
460 001302 012767 000010 010344 MOV #10,ICOUNT ;SET UP FOR 10 ITERATIONS
461 001310 012767 001434 010332 MOV #4$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
462 001316 012767 001364 010326 MOV #1$,FREEZ1 ;SET UP TO LOOP WITH DATA
463 001324 012777 004000 010254 MOV #BIT11,ADHSCR ;MASTER CLEAR INTERFACE
464 001332 012703 000000 MOV #0,R3 ;SET UP LINE #
465 001336 012705 130000 MOV #0*400+130000,R5 ;SET EXPECTED LINE NUMBER
466 ;VALID DATA FLAG,
467 ;PARITY ERROR,DATA OVERRUN,
468 ;AND EXPECTED DATA
469 001342 012767 000377 010340 MOV #377,TDATA ;ACTUAL DATA TO BE TRANSMITTED
470 001350 012777 000000 010230 MOV #0,ADHSCR ;SELECT LINE 0
471 001356 012777 073563 010226 MOV #73563,ADHLPR ;SELECT 8 BITS/CHAR,ODD PARITY HDX,
472 ;9600 BAUD SPEED FOR LINE 0
473 001364 012777 177777 010224 1$: MOV #-1,ADHBC ;TRANSMIT 1 CHARACTER
474 001372 012777 011710 010214 MOV #TDATA,ADHBA ;ADDRESS OF TRANSMITTED DATA
475 001400 012777 000001 010214 MOV #1,ADHBCR ;SET BREAK FOR LINE 0.
476 001406 012777 000001 010204 MOV #1,ADHBAR ;START TRANSMITTER
477 001414 105777 010166 2$: TSTB ADHSCR ;WAIT FOR CHARACTER TO BE RECEIVED
478 001420 100375 BPL 2$ ;
479 001422 017701 010162 MOV ADHNR,R1 ;GET RECEIVED CHARACTER
480 001426 020501 CMP R5,R1 ;COMPARE EXPECTED AND RECEIVED DATA
481 001430 001401 BEQ +4 ;
482 001432 104003 HLT 3 ;DATA ERROR
483 001434 104400 4$: SCOPE ;CHECK FOR ITERATIONS,LOOP
484
485 ;RECEIVER PARITY LOGIC TEST ON LINE 1.
486 ;TRANSMIT ONE CHARACTER TO THE RECEIVER
487 ;WITH ODD PARITY SET,BREAK SET,HDX SET.
488 ;THE RECEIVER WILL RECEIVE AN ALL 0'S CHARACTER
489 ;WITH EVEN PARITY AND THEREBY
490 ;CAUSE A PARITY ERROR AND OVERRUN
491 ;CHARACTER LENGTH: 8 BITS
492 ;LINE SPEED: 9600 BAUD
493
494 001436 012767 000340 176332 T2: MOV #340,PS ;DISABLE ALL INTERRUPTS
495 001444 012767 000010 010202 MOV #10,ICOUNT ;SET UP FOR 10 ITERATIONS
496 001452 012767 001576 010170 MOV #4$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
497 001460 012767 001526 010164 MOV #1$,FREEZ1 ;SET UP TO LOOP WITH DATA
498 001466 012777 004000 010112 MOV #BIT11,ADHSCR ;MASTER CLEAR INTERFACE
499 001474 012703 000001 MOV #1,R3 ;SET UP LINE #
500 001500 012705 130400 MOV #1*400+130000,R5 ;SET EXPECTED LINE NUMBER
501 ;VALID DATA FLAG,
502 ;PARITY ERROR,DATA OVERRUN,
503 ;AND EXPECTED DATA
504 001504 012767 000377 010176 MOV #377,TDATA ;ACTUAL DATA TO BE TRANSMITTED

```


NO1

```

561                                     ;CHARACTER LENGTH: 8 BITS
562                                     ;LINE SPEED: 9600 BAUD
563
564 001742 012767 000340 176026 T4:  MOV    #340,PS                ;DISABLE ALL INTERRUPTS
565 001750 012767 000010 007676      MOV    #10,ICOUNT           ;SET UP FOR 10 ITERATIONS
566 001756 012767 002102 007664      MOV    #4$,ESCAPE         ;SET UP TO ESCAPE TO NEXT TEST
567 001764 012767 002032 007660      MOV    #1$,FREEZ1        ;SET UP TO LOOP WITH DATA
568 001772 012777 004000 007606      MOV    #BIT11,ADHSCR      ;MASTER CLEAR INTERFACE
569 002000 012703 000003              MOV    #3,R3              ;SET UP LINE #
570 002004 012705 131400              MOV    #3*400+130000,R5   ;SET EXPECTED LINE NUMBER
571                                     ;VALID DATA FLAG,
572                                     ;PARITY ERROR,DATA OVERRUN,
573                                     ;AND EXPECTED DATA
574 002010 012767 000377 007672      MOV    #377,TDATA        ;ACTUAL DATA TO BE TRANSMITTED
575 002016 012777 000003 007562      MOV    #3,ADHSCR         ;SELECT LINE 3
576 002024 012777 073563 007560      MOV    #73563,ADHLPR     ;SELECT 8 BITS/CHAR,ODD PARITY HDX,
577                                     ;9600 BAUD SPEED FOR LINE 3
578 002032 012777 177777 007556 1$:  MOV    #-1,ADHBC         ;TRANSMIT 1 CHARACTER
579 002040 012777 011710 007546      MOV    #TDATA,ADHBA      ;ADDRESS OF TRANSMITTED DATA
580 002046 012777 000010 007546      MOV    #10,ADHBCR        ;SET BREAK FOR LINE 3.
581 002054 012777 000010 007536      MOV    #10,ADHBAR        ;START TRANSMITTER
582 002062 105777 007520 2$:  TSTB  ADHSCR ;WAIT FOR CHARACTER TO BE RECEIVED
583 002066 100375              BPL   2$
584 002070 017701 007514      MOV    ADHNR,R1          ;GET RECEIVED CHARACTER
585 002074 020501              CMP   R5,R1              ;COMPARE EXPECTED AND RECEIVED DATA
586 002076 001401              BEQ  +4
587 002100 104003              HLT  3                   ;DATA ERROR
588 002102 104400 4$:  SCOPE ;CHECK FOR ITERATIONS,LOOP
589
590                                     ;RECEIVER PARITY LOGIC TEST ON LINE 4.
591                                     ;TRANSMIT ONE CHARACTER TO THE RECEIVER
592                                     ;WITH ODD PARITY SET,BREAK SET,HDx SET.
593                                     ;THE RECEIVER WILL RECEIVE AN ALL 0'S CHARACTER
594                                     ;WITH EVEN PARITY AND THEREBY
595                                     ;CAUSE A PARITY ERROR AND OVERRUN
596                                     ;CHARACTER LENGTH: 8 BITS
597                                     ;LINE SPEED: 9600 BAUD
598
599 002104 012767 000340 175564 T5:  MOV    #340,PS                ;DISABLE ALL INTERRUPTS
600 002112 012767 000010 007534      MOV    #10,ICOUNT           ;SET UP FOR 10 ITERATIONS
601 002120 012767 002244 007522      MOV    #4$,ESCAPE         ;SET UP TO ESCAPE TO NEXT TEST
602 002126 012767 002174 007516      MOV    #1$,FREEZ1        ;SET UP TO LOOP WITH DATA
603 002134 012777 004000 007444      MOV    #BIT11,ADHSCR      ;MASTER CLEAR INTERFACE
604 002142 012703 000004              MOV    #4,R3              ;SET UP LINE #
605 002146 012705 132000              MOV    #4*400+130000,R5   ;SET EXPECTED LINE NUMBER
606                                     ;VALID DATA FLAG,
607                                     ;PARITY ERROR,DATA OVERRUN,
608                                     ;AND EXPECTED DATA
609 002152 012767 000377 007530      MOV    #377,TDATA        ;ACTUAL DATA TO BE TRANSMITTED
610 002160 012777 000004 007420      MOV    #4,ADHSCR         ;SELECT LINE 4
611 002166 012777 073563 007416      MOV    #73563,ADHLPR     ;SELECT 8 BITS/CHAR,ODD PARITY HDX,
612                                     ;9600 BAUD SPEED FOR LINE 4
613 002174 012777 177777 007414 1$:  MOV    #-1,ADHBC         ;TRANSMIT 1 CHARACTER
614 002202 012777 011710 007404      MOV    #TDATA,ADHBA      ;ADDRESS OF TRANSMITTED DATA
615 002210 012777 000020 007404      MOV    #20,ADHBCR        ;SET BREAK FOR LINE 4.
616 002216 012777 000020 007374      MOV    #20,ADHBAR        ;START TRANSMITTER

```



```

617 002224 105777 007356 25: TSTB 3DHSCR ;WAIT FOR CHARACTER TO BE RECEIVED
618 002230 100375 BPL 25 ;
619 002232 017701 007352 MOV 3DHNRC,R1 ;GET RECEIVED CHARACTER
620 002236 020501 CMP R5,R1 ;COMPARE EXPECTED AND RECEIVED DATA
621 002240 001401 BEQ +4 ;
622 002242 104003 HLT 3 ;DATA ERROR
623 002244 104400 45: SCOPE ;CHECK FOR ITERATIONS,LOOP
624
625 ;RECEIVER PARITY LOGIC TEST ON LINE 5.
626 ;TRANSMIT ONE CHARACTER TO THE RECEIVER
627 ;WITH ODD PARITY SET,BREAK SET,HDX SET.
628 ;THE RECEIVER WILL RECEIVE AN ALL 0'S CHARACTER
629 ;WITH EVEN PARITY AND THEREBY
630 ;CAUSE A PARITY ERROR AND OVERRUN
631 ;CHARACTER LENGTH: 8 BITS
632 ;LINE SPEED: 9600 BAUD
633
634 002246 012767 000340 175522 T6: MOV #340,PS ;DISABLE ALL INTERRUPTS
635 002254 012767 000010 007372 MOV #10,ICOUNT ;SET UP FOR 10 ITERATIONS
636 002262 012767 002406 007360 MOV #45,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
637 002270 012767 002336 007354 MOV #15,FREEZ1 ;SET UP TO LOOP WITH DATA
638 002276 012777 004000 007302 MOV #BIT11,3DHSCR ;MASTER CLEAR INTERFACE
639 002304 012703 000005 MOV #5,R3 ;SET UP LINE #
640 002310 012705 132400 MOV #5*400+130000,R5 ;SET EXPECTED LINE NUMBER
641 ;VALID DATA FLAG,
642 ;PARITY ERROR,DATA OVERRUN,
643 ;AND EXPECTED DATA
644 002314 012767 000377 007366 MOV #377,TDATA ;ACTUAL DATA TO BE TRANSMITTED
645 002322 012777 000005 007256 MOV #5,3DHSCR ;SELECT LINE 5
646 002330 012777 073563 007254 MOV #73563,3DHLPR ;SELECT 8 BITS/CHAR,ODD PARITY HDX,
647 ;9600 BAUD SPEED FOR LINE 5
648 002336 012777 177777 007252 15: MOV #-1,3DHBC ;TRANSMIT 1 CHARACTER
649 002344 012777 011710 007242 MOV #TDATA,3DHBA ;ADDRESS OF TRANSMITTED DATA
650 002352 012777 000040 007242 MOV #40,3DHBCR ;SET BREAK FOR LINE 5.
651 002360 012777 000040 007232 MOV #40,3DHBAR ;START TRANSMITTER
652 002366 105777 007214 25: TSTB 3DHSCR ;WAIT FOR CHARACTER TO BE RECEIVED
653 002372 100375 BPL 25 ;
654 002374 017701 007210 MOV 3DHNRC,R1 ;GET RECEIVED CHARACTER
655 002400 020501 CMP R5,R1 ;COMPARE EXPECTED AND RECEIVED DATA
656 002402 001401 BEQ +4 ;
657 002404 104003 HLT 3 ;DATA ERROR
658 002406 104400 45: SCOPE ;CHECK FOR ITERATIONS,LOOP
659
660 ;RECEIVER PARITY LOGIC TEST ON LINE 6.
661 ;TRANSMIT ONE CHARACTER TO THE RECEIVER
662 ;WITH ODD PARITY SET,BREAK SET,HDX SET.
663 ;THE RECEIVER WILL RECEIVE AN ALL 0'S CHARACTER
664 ;WITH EVEN PARITY AND THEREBY
665 ;CAUSE A PARITY ERROR AND OVERRUN
666 ;CHARACTER LENGTH: 8 BITS
667 ;LINE SPEED: 9600 BAUD
668
669 002410 012767 000340 175360 T7: MOV #340,PS ;DISABLE ALL INTERRUPTS
670 002416 012767 000010 007230 MOV #10,ICOUNT ;SET UP FOR 10 ITERATIONS
671 002424 012767 002550 007216 MOV #45,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
672 002432 012767 002500 007212 MOV #15,FREEZ1 ;SET UP TO LOOP WITH DATA

```



```

673 002440 012777 004000 007140      MOV      #BIT11,JDHSCR      ;MASTER CLEAR INTERFACE
674 002446 012703 000006                MOV      #6,R3             ;SET UP LINE #
675 002452 012705 133000                MOV      #6*400+130000,R5 ;SET EXPECTED LINE NUMBER
676                                     ;VALID DATA FLAG,
677                                     ;PARITY ERROR,DATA OVERRUN,
678                                     ;AND EXPECTED DATA
679 002456 012767 000377 007224      MOV      #377,TDATA       ;ACTUAL DATA TO BE TRANSMITTED
680 002464 012777 000006 007114      MOV      #6,JDHSCR        ;SELECT LINE 6
681 002472 012777 073563 007112      MOV      #73563,JDHLPR    ;SELECT 8 BITS/CHAR,ODD PARITY HDX,
682                                     ;9600 BAUD SPEED FOR LINE 6
683 002500 012777 177777 007110 1$:    MOV      #-1,JDHBC        ;TRANSMIT 1 CHARACTER
684 002506 012777 011710 007100      MOV      #TDATA,JDHBA     ;ADDRESS OF TRANSMITTED DATA
685 002514 012777 000100 007100      MOV      #100,JDHBCR      ;SET BREAK FOR LINE 6.
686 002522 012777 000100 007070      MOV      #100,JDHBR       ;START TRANSMITTER
687 002530 105777 007052 2$:    TSTB    JDHSCR            ;WAIT FOR CHARACTER TO BE RECEIVED
688 002534 100375                                     BPL      2$
689 002536 017701 007046                MOV      JDHNR,R1         ;GET RECEIVED CHARACTER
690 002542 020501                CMP      R5,R1           ;COMPARE EXPECTED AND RECEIVED DATA
691 002544 001401                BEQ      +4
692 002546 104003                HLT      3               ;DATA ERROR
693 002550 104400 4$:    SCOPE    ;CHECK FOR ITERATIONS,LOOP
694
695 ;RECEIVER PARITY LOGIC TEST ON LINE 7.
696 ;TRANSMIT ONE CHARACTER TO THE RECEIVER
697 ;WITH ODD PARITY SET,BREAK SET,HDH SET.
698 ;THE RECEIVER WILL RECEIVE AN ALL 0'S CHARACTER
699 ;WITH EVEN PARITY AND THEREBY
700 ;CAUSE A PARITY ERROR AND OVERRUN
701 ;CHARACTER LENGTH: 8 BITS
702 ;LINE SPEED: 9600 BAUD
703
704 002552 012767 000340 175216 T10:  MOV      #340,PS         ;DISABLE ALL INTERRUPTS
705 002560 012767 000010 007066      MOV      #10,I/COUNT     ;SET UP FOR 10 ITERATIONS
706 002566 012767 002712 007054      MOV      #4$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
707 002574 012767 002642 007050      MOV      #1$,FREEZ1      ;SET UP TO LOOP WITH DATA
708 002602 012777 004000 006776      MOV      #BIT11,JDHSCR    ;MASTER CLEAR INTERFACE
709 002610 012703 000007                MOV      #7,R3           ;SET UP LINE #
710 002614 012705 133400                MOV      #7*400+130000,R5 ;SET EXPECTED LINE NUMBER
711                                     ;VALID DATA FLAG,
712                                     ;PARITY ERROR,DATA OVERRUN,
713                                     ;AND EXPECTED DATA
714 002620 012767 000377 007062      MOV      #377,TDATA       ;ACTUAL DATA TO BE TRANSMITTED
715 002626 012777 000007 006752      MOV      #7,JDHSCR        ;SELECT LINE 7
716 002634 012777 073563 006750      MOV      #73563,JDHLPR    ;SELECT 8 BITS/CHAR,ODD PARITY HDX,
717                                     ;9600 BAUD SPEED FOR LINE 7
718 002642 012777 177777 006746 1$:    MOV      #-1,JDHBC        ;TRANSMIT 1 CHARACTER
719 002650 012777 011710 006736      MOV      #TDATA,JDHBA     ;ADDRESS OF TRANSMITTED DATA
720 002656 012777 000200 006736      MOV      #200,JDHBCR      ;SET BREAK FOR LINE 7.
721 002664 012777 000200 006726      MOV      #200,JDHBR       ;START TRANSMITTER
722 002672 105777 006710 2$:    TSTB    JDHSCR            ;WAIT FOR CHARACTER TO BE RECEIVED
723 002676 100375                                     BPL      2$
724 002700 017701 006704                MOV      JDHNR,R1         ;GET RECEIVED CHARACTER
725 002704 020501                CMP      R5,R1           ;COMPARE EXPECTED AND RECEIVED DATA
726 002706 001401                BEQ      +4
727 002710 104003                HLT      3               ;DATA ERROR
728 002712 104400 4$:    SCOPE    ;CHECK FOR ITERATIONS,LOOP

```



```

729
730
731
732
733
734
735
736
737
738
739 002714 012767 000340 175054 T11: MOV #340,PS ;DISABLE ALL INTERRUPTS
740 002722 012767 000010 006724 MOV #10,I,COUNT ;SET UP FOR 10 ITERATIONS
741 002730 012767 003054 006712 MOV #45,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
742 002736 012767 003004 006706 MOV #15,FREEZ1 ;SET UP TO LOOP WITH DATA
743 002744 012777 004000 006634 MOV #BIT11,JDHSCR ;MASTER CLEAR INTERFACE
744 002752 012703 000010 MOV #10,R3 ;SET UP LINE #
745 002756 012705 134000 MOV #10*400+130000,R5 ;SET EXPECTED LINE NUMBER
746 ;VALID DATA FLAG,
747 ;PARITY ERROR,DATA OVERRUN,
748 ;AND EXPECTED DATA
749 002762 012767 000377 006720 MOV #377,TDATA ;ACTUAL DATA TO BE TRANSMITTED
750 002770 012777 000010 006610 MOV #10,JDHSCR ;SELECT LINE 10
751 002776 012777 073563 006606 MOV #73563,JDHLPR ;SELECT 8 BITS/CHAR,ODD PARITY HDX,
752 ;9600 BAUD SPEED FOR LINE 10
753 003004 012777 177777 006604 15: MOV #-1,JDHBC ;TRANSMIT 1 CHARACTER
754 003012 012777 011710 006574 MOV #TDATA,JDHBA ;ADDRESS OF TRANSMITTED DATA
755 003020 012777 000400 006574 MOV #400,JDHBCR ;SET BREAK FOR LINE 10.
756 003026 012777 000400 006564 MOV #400,JDHBAR ;START TRANSMITTER
757 003034 105777 006546 25: TSTB JDHSCR ;WAIT FOR CHARACTER TO BE RECEIVED
758 003040 100375 BPL 25 ;
759 003042 017701 006542 MOV JDHNR, R1 ;GET RECEIVED CHARACTER
760 003046 020501 CMP R5,R1 ;COMPARE EXPECTED AND RECEIVED DATA
761 003050 001401 BEQ +4 ;
762 003052 104003 HLT 3 ;DATA ERROR
763 003054 104400 45: SCOPE ;CHECK FOR ITERATIONS,LOOP
764
765 ;RECEIVER PARITY LOGIC TEST ON LINE 11.
766 ;TRANSMIT ONE CHARACTER TO THE RECEIVER
767 ;WITH ODD PARITY SET,BREAK SET,HDX SET.
768 ;THE RECEIVER WILL RECEIVE AN ALL 0'S CHARACTER
769 ;WITH EVEN PARITY AND THEREBY
770 ;CAUSE A PARITY ERROR AND OVERRUN
771 ;CHARACTER LENGTH: 8 BITS
772 ;LINE SPEED: 9600 BAUD
773
774 003056 012767 000340 174712 T12: MOV #340,PS ;DISABLE ALL INTERRUPTS
775 003064 012767 000010 006562 MOV #10,I,COUNT ;SET UP FOR 10 ITERATIONS
776 003072 012767 003216 006550 MOV #45,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
777 003100 012767 003146 006544 MOV #15,FREEZ1 ;SET UP TO LOOP WITH DATA
778 003106 012777 004000 006472 MOV #BIT11,JDHSCR ;MASTER CLEAR INTERFACE
779 003114 012703 000011 MOV #11,R3 ;SET UP LINE #
780 003120 012705 134400 MOV #11*400+130000,R5 ;SET EXPECTED LINE NUMBER
781 ;VALID DATA FLAG,
782 ;PARITY ERROR,DATA OVERRUN,
783 ;AND EXPECTED DATA
784 003124 012767 000377 006556 MOV #377,TDATA ;ACTUAL DATA TO BE TRANSMITTED

```


F02

DZDHG MACY11 27(732) 28-SEP-76 15:16 PAGE 19
 DZDHGB.P11

```

841                                     ;CHARACTER LENGTH: 8 BITS
842                                     ;LINE SPEED: 9600 BAUD
843
844 003362 012767 000340 174406 T14:  MOV    #340,PS                ;DISABLE ALL INTERRUPTS
845 003370 012767 000010 006256      MOV    #10,I'COUNT          ;SET UP FOR 10 ITERATIONS
846 003376 012767 003522 006244      MOV    #4$,ESCAPE           ;SET UP TO ESCAPE TO NEXT TEST
847 003404 012767 003452 006240      MOV    #1$,FREEZ1          ;SET UP TO LOOP WITH DATA
848 003412 012777 004000 006166      MOV    #BIT11,JDHSCR       ;MASTER CLEAR INTERFACE
849 003420 012703 000013              MOV    #13,R3              ;SET UP LINE #
850 003424 012705 135400              MOV    #13*400+130000,R5   ;SET EXPECTED LINE NUMBER
851                                     ;VALID DATA FLAG,
852                                     ;PARITY ERROR,DATA OVERRUN,
853                                     ;AND EXPECTED DATA
854 003430 012767 000377 006252      MOV    #377,TDATA          ;ACTUAL DATA TO BE TRANSMITTED
855 003436 012777 000013 006142      MOV    #13,JDHSCR          ;SELECT LINE 13
856 003444 012777 073563 006140      MOV    #73563,JDHLPR       ;SELECT 8 BITS/CHAR,ODD PARITY HDX,
857                                     ;9600 BAUD SPEED FOR LINE 13
858 003452 012777 177777 006136 1$:  MOV    #-1,JDHBC           ;TRANSMIT 1 CHARACTER
859 003460 012777 011710 006126      MOV    #TDATA,JDHBA        ;ADDRESS OF TRANSMITTED DATA
860 003466 012777 004000 006126      MOV    #4000,JDHBCR        ;SET BREAK FOR LINE 13.
861 003474 012777 004000 006116      MOV    #4000,JDHBAR        ;START TRANSMITTER
862 003502 105777 006100              TSTB   JDHSCR              ;WAIT FOR CHARACTER TO BE RECEIVED
863 003506 100375                    BPL    2$                  ;
864 003510 017701 006074              MOV    #JDHNR,RI           ;GET RECEIVED CHARACTER
865 003514 020501                    CMP    R5,R1              ;COMPARE EXPECTED AND RECEIVED DATA
866 003516 001401                    BEQ    +4                  ;
867 003520 104003                    HLT    3                  ;DATA ERROR
868 003522 104400 4$:  SCOPE      ;CHECK FOR ITERATIONS,LOOP
869
870                                     ;RECEIVER PARITY LOGIC TEST ON LINE 14.
871                                     ;TRANSMIT ONE CHARACTER TO THE RECEIVER
872                                     ;WITH ODD PARITY SET,BREAK SET,HDX SET.
873                                     ;THE RECEIVER WILL RECEIVE AN ALL 0'S CHARACTER
874                                     ;WITH EVEN PARITY AND THEREBY
875                                     ;CAUSE A PARITY ERROR AND OVERRUN
876                                     ;CHARACTER LENGTH: 8 BITS
877                                     ;LINE SPEED: 9600 BAUD
878
879 003524 012767 000340 174244 T15:  MOV    #340,PS                ;DISABLE ALL INTERRUPTS
880 003532 012767 000010 006114      MOV    #10,I'COUNT          ;SET UP FOR 10 ITERATIONS
881 003540 012767 003664 006102      MOV    #4$,ESCAPE           ;SET UP TO ESCAPE TO NEXT TEST
882 003546 012767 003614 006076      MOV    #1$,FREEZ1          ;SET UP TO LOOP WITH DATA
883 003554 012777 004000 006024      MOV    #BIT11,JDHSCR       ;MASTER CLEAR INTERFACE
884 003562 012703 000014              MOV    #14,R3              ;SET UP LINE #
885 003566 012705 136000              MOV    #14*400+130000,R5   ;SET EXPECTED LINE NUMBER
886                                     ;VALID DATA FLAG,
887                                     ;PARITY ERROR,DATA OVERRUN,
888                                     ;AND EXPECTED DATA
889 003572 012767 000377 006110      MOV    #377,TDATA          ;ACTUAL DATA TO BE TRANSMITTED
890 003600 012777 000014 006000      MOV    #14,JDHSCR          ;SELECT LINE 14
891 003606 012777 073563 005776      MOV    #73563,JDHLPR       ;SELECT 8 BITS/CHAR,ODD PARITY HDX,
892                                     ;9600 BAUD SPEED FOR LINE 14
893 003614 012777 177777 005774 1$:  MOV    #-1,JDHBC           ;TRANSMIT 1 CHARACTER
894 003622 012777 011710 005764      MOV    #TDATA,JDHBA        ;ADDRESS OF TRANSMITTED DATA
895 003630 012777 010000 005764      MOV    #10000,JDHBCR       ;SET BREAK FOR LINE 14.
896 003636 012777 010000 005754      MOV    #10000,JDHBAR       ;START TRANSMITTER
  
```

G02

DZDHG MACY11 27(732) 29-SEP-76 15:16 PAGE 20
 DZDHGB.P11

```

897 003644 105777 005736      2$:  TSTB   @DHSCR ;WAIT FOR CHARACTER TO BE RECEIVED
898 003650 100375                BPL     2$
899 003652 017701 005732      MOV     @DHNRC,R1 ;GET RECEIVED CHARACTER
900 003656 020501                CMP     R5,R1 ;COMPARE EXPECTED AND RECEIVED DATA
901 003660 001401                BEQ     +4
902 003662 104003                HLT     3 ;DATA ERROR
903 003664 104400      4$:  SCOPE   ;CHECK FOR ITERATIONS,LOOP
904
905 ;RECEIVER PARITY LOGIC TEST ON LINE 15.
906 ;TRANSMIT ONE CHARACTER TO THE RECEIVER
907 ;WITH ODD PARITY SET,BREAK SET,HDX SET.
908 ;THE RECEIVER WILL RECEIVE AN ALL 0'S CHARACTER
909 ;WITH EVEN PARITY AND THEREBY
910 ;CAUSE A PARITY ERROR AND OVERRUN
911 ;CHARACTER LENGTH: 8 BITS
912 ;LINE SPEED: 9600 BAUD
913
914 003666 012767 000340 174102 T16:  MOV     #340,PS ;DISABLE ALL INTERRUPTS
915 003674 012767 000010 005752      MOV     #10,ICOUNT ;SET UP FOR 10 ITERATIONS
916 003702 012767 004026 005740      MOV     #4$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
917 003710 012767 003756 005734      MOV     #1$,FREEZ1 ;SET UP TO LOOP WITH DATA
918 003716 012777 004000 005662      MOV     #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
919 003724 012703 000015      MOV     #15,R3 ;SET UP LINE #
920 003730 012705 136400      MOV     #15*400+130000,R5 ;SET EXPECTED LINE NUMBER
921 ;VALID DATA FLAG,
922 ;PARITY ERROR,DATA OVERRUN,
923 ;AND EXPECTED DATA
924 003734 012767 000377 005746      MOV     #377,TDATA ;ACTUAL DATA TO BE TRANSMITTED
925 003742 012777 000015 005636      MOV     #15,@DHSCR ;SELECT LINE 15
926 003750 012777 073563 005634      MOV     #73563,@DHLPR ;SELECT 8 BITS/CHAR,ODD PARITY HDX,
927 ;9600 BAUD SPEED FOR LINE 15
928 003756 012777 177777 005632 1$:  MOV     #-1,@DHBC ;TRANSMIT 1 CHARACTER
929 003764 012777 011710 005622      MOV     #TDATA,@DHBA ;ADDRESS OF TRANSMITTED DATA
930 003772 012777 020000 005622      MOV     #20000,@DHBCR ;SET BREAK FOR LINE 15.
931 004000 012777 020000 005612      MOV     #20000,@DHBAR ;START TRANSMITTER
932 004006 105777 005574      2$:  TSTB   @DHSCR ;WAIT FOR CHARACTER TO BE RECEIVED
933 004012 100375                BPL     2$
934 004014 017701 005570      MOV     @DHNRC,R1 ;GET RECEIVED CHARACTER
935 004020 020501                CMP     R5,R1 ;COMPARE EXPECTED AND RECEIVED DATA
936 004022 001401                BEQ     +4
937 004024 104003                HLT     3 ;DATA ERROR
938 004026 104400      4$:  SCOPE   ;CHECK FOR ITERATIONS,LOOP
939
940 ;RECEIVER PARITY LOGIC TEST ON LINE 16.
941 ;TRANSMIT ONE CHARACTER TO THE RECEIVER
942 ;WITH ODD PARITY SET,BREAK SET,HDX SET.
943 ;THE RECEIVER WILL RECEIVE AN ALL 0'S CHARACTER
944 ;WITH EVEN PARITY AND THEREBY
945 ;CAUSE A PARITY ERROR AND OVERRUN
946 ;CHARACTER LENGTH: 8 BITS
947 ;LINE SPEED: 9600 BAUD
948
949 004030 012767 000340 173740 T17:  MOV     #340,PS ;DISABLE ALL INTERRUPTS
950 004036 012767 000010 005610      MOV     #10,ICOUNT ;SET UP FOR 10 ITERATIONS
951 004044 012767 004170 005576      MOV     #4$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
952 004052 012767 004120 005572      MOV     #1$,FREEZ1 ;SET UP TO LOOP WITH DATA
  
```


H02

```

953 004060 012777 004000 005520      MOV      #BIT11,ADHSCR      ;MASTER CLEAR INTERFACE
954 004066 012703 000016                MOV      #16,R3           ;SET UP LINE #
955 004072 012705 137000                MOV      #16*400+130000,R5 ;SET EXPECTED LINE NUMBER
956                                     ;VALID DATA FLAG,
957                                     ;PARITY ERROR,DATA OVERRUN,
958                                     ;AND EXPECTED DATA
959 004076 012767 000377 005604      MOV      #377,TDATA      ;ACTUAL DATA TO BE TRANSMITTED
960 004104 012777 000016 005474      MOV      #16,ADHSCR      ;SELECT LINE 16
961 004112 012777 073563 005472      MOV      #73563,ADHLPR   ;SELECT 8 BITS/CHAR,ODD PARITY HDX,
962                                     ;9600 BAUD SPEED FOR LINE 16
963 004120 012777 177777 005470 1$:    MOV      #-1,ADHBC       ;TRANSMIT 1 CHARACTER
964 004126 012777 011710 005460      MOV      #TDATA,ADHBA    ;ADDRESS OF TRANSMITTED DATA
965 004134 012777 040000 005460      MOV      #40000,ADHBCR   ;SET BREAK FOR LINE 16.
966 004142 012777 040000 005450      MOV      #40000,ADHBAR   ;START TRANSMITTER
967 004150 105777 005432                TSTB    ADHSCR           ;WAIT FOR CHARACTER TO BE RECEIVED
968 004154 100375                        BPL     2$              ;
969 004156 017701 005426                MOV      ADHNR,R1        ;GET RECEIVED CHARACTER
970 004162 020501                        CMP     R5,R1           ;COMPARE EXPECTED AND RECEIVED DATA
971 004164 001401                        BEQ     +4              ;
972 004166 104003                        HLT     3               ;DATA ERROR
973 004170 104400 4$:                SCOPE   ;CHECK FOR ITERATIONS,LOOP
974
975                                     ;RECEIVER PARITY LOGIC TEST ON LINE 17.
976                                     ;TRANSMIT ONE CHARACTER TO THE RECEIVER
977                                     ;WITH ODD PARITY SET,BREAK SET,HDX SET.
978                                     ;THE RECEIVER WILL RECEIVE AN ALL 0'S CHARACTER
979                                     ;WITH EVEN PARITY AND THEREBY
980                                     ;CAUSE A PARITY ERROR AND OVERRUN
981                                     ;CHARACTER LENGTH: 8 BITS
982                                     ;LINE SPEED: 9600 BAUD
983
984 004172 012767 000340 173576 T20:  MOV      #340,PS         ;DISABLE ALL INTERRUPTS
985 004200 012767 000010 005446      MOV      #10,ICOUNT     ;SET UP FOR 10 ITERATIONS
986 004206 012767 004332 005434      MOV      #4$,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST
987 004214 012767 004262 005430      MOV      #1$,FREEZ1     ;SET UP TO LOOP WITH DATA
988 004222 012777 004000 005356      MOV      #BIT11,ADHSCR  ;MASTER CLEAR INTERFACE
989 004230 012703 000017                MOV      #17,R3         ;SET UP LINE #
990 004234 012705 137400                MOV      #17*400+130000,R5 ;SET EXPECTED LINE NUMBER
991                                     ;VALID DATA FLAG,
992                                     ;PARITY ERROR,DATA OVERRUN,
993                                     ;AND EXPECTED DATA
994 004240 012767 000377 005442      MOV      #377,TDATA     ;ACTUAL DATA TO BE TRANSMITTED
995 004246 012777 000017 005332      MOV      #17,ADHSCR     ;SELECT LINE 17
996 004254 012777 073563 005330      MOV      #73563,ADHLPR ;SELECT 8 BITS/CHAR,ODD PARITY HDX,
997                                     ;9600 BAUD SPEED FOR LINE 17
998 004262 012777 177777 005326 1$:    MOV      #-1,ADHBC       ;TRANSMIT 1 CHARACTER
999 004270 012777 011710 005316      MOV      #TDATA,ADHBA   ;ADDRESS OF TRANSMITTED DATA
1000 004276 012777 100000 005316      MOV      #100000,ADHBCR ;SET BREAK FOR LINE 17.
1001 004304 012777 100000 005306      MOV      #100000,ADHBAR ;START TRANSMITTER
1002 004312 105777 005270 2$:    TSTB    ADHSCR           ;WAIT FOR CHARACTER TO BE RECEIVED
1003 004316 100375                        BPL     2$              ;
1004 004320 017701 005264                MOV      ADHNR,R1        ;GET RECEIVED CHARACTER
1005 004324 020501                        CMP     R5,R1           ;COMPARE EXPECTED AND RECEIVED DATA
1006 004326 001401                        BEQ     +4              ;
1007 004330 104003                        HLT     3               ;DATA ERROR
1008 004332 104400 4$:                SCOPE   ;CHECK FOR ITERATIONS,LOOP

```

```

1009
1010 ;MULTI-LINE PARITY DATA TEST
1011 ;TRANSMIT A BINARY COUNT WITH ODD PARITY
1012 ;PATTERN ON ALL LINES.
1013 ;SILO ALARM LEVEL SET TO 0
1014 ;RECEIVER WILL BE SERVICED ON A PER CHARACTER BASIS
1015 ;IN INTERRUPT MODE.
1016 ;TRANSMITTER INTERRUPT WILL BE DISABLED
1017 ;LINE SPEED: 2400 BAUD
1018 ;CHARACTER LENGTH: 8 + PARITY
1019
1020 004334 012767 000340 173434 T21: MOV #340,PS ;DISABLE ALL INTERRUPTS
1021 004342 012767 000010 005304 MOV #10,ICOUNT ;SET UP FOR 10 ITERATIONS
1022 004350 012767 004602 005272 MOV #11$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
1023 004356 012777 004000 005222 MOV #BIT11,JDHSCR ;MASTER CLEAR INTERFACE
1024 004364 012700 000020 MOV #16.,R0 ;SET UP TO START 16. LINES
1025 004370 005001 CLR R1 ;COUNT AND BUS ADDRESS MEMORIES
1026 ;AND RECEIVED DATA STORAGE
1027 004372 012702 000200 MOV #200,R2 ;SET UP TO LOAD HIGH BYTE
1028 004376 012703 000001 MOV #1,R3 ;OF EXPECTED DATA
1029 004402 012777 011712 005204 1$: MOV #TBUF,JDHBA ;LOAD BUSS ADDRESS
1030 004410 012777 177400 005200 MOV #-400,JDHBC ;LOAD BYTE COUNT
1031 004416 012777 027363 005166 MOV #27363,JDHLPR ;SET LINE SPEED AND
1032 ;CHARACTER LENGTH 8 + ODD PARITY
1033 004424 105061 012312 CLRB RBUF(R1) ;CLEAR NEXT RECEIVED CHARACTER
1034 004430 110263 012312 MOVB R2,RBUF(R3) ;LOAD HIGH BYTE
1035 004434 005277 005146 INC JDHSCR ;ADVANCE TO NEXT LINE
1036 004440 005202 INC R2 ;UPDATE POINTER TO
1037 004442 062701 000002 ADD #2,R1 ;LOW AND HIGH BYTE OF
1038 004446 062703 000002 ADD #2,R3 ;NEXT EXPECTED CHARACTER
1039 004452 005300 DEC R0 ;CONTINUE IF ALL LINES
1040 004454 001352 BNE 1$ ;NOT SET UP
1041 004456 012777 004516 005144 MOV #6$,JDHRVEC ;SET UP POINTER FOR
1042 004464 012777 000240 005140 MOV #240,JDHRLVL ;RECEIVER INTERRUPT
1043 004472 012777 000100 005106 MOV #100,JDHSCR ;ENABLE RECEIVER INTERRUPT
1044 004500 012777 177777 005112 MOV #-1,JDHBAR ;SET ALL BAR BITS
1045 004506 005067 173264 CLR PS ;ENABLE INTERRUPTS
1046 004512 000167 000076 JMP 12$ ;GO TO TRANSMITTER WAITING ROUTINE
1047
1048 ;RECEIVER INTERRUPT SERVICE ROUTINE
1049 ;CHECK FOR RECEIVER DONE AND NO ERRORS
1050 ;CHECK RECEIVED DATA FOR "VALID DATA" FLAG AND NO ERRORS
1051 ;VERIFY THAT RECEIVED DATA IS CORRECT
1052 ;CHECK FOR END OF PASS
1053
1054 004516 105777 005064 6$: TSTB JDHSCR ;IF CHARACTER AVAILABLE NOT SET, ERROR
1055 004522 100026 BPL 10$
1056 004524 032777 040000 005054 BIT #40000,JDHSCR ;IF SILO OVERRUN SET, ERROR
1057 004532 001401 BEQ .+4
1058 004534 104002 HLT 2 ;SILO OVERRUN, ERROR
1059 004536 017701 005046 7$: MOV JDHNR, R1 ;READ CHARACTER FROM SILO
1060 004542 010102 MOV R1,R2 ;EXTRACT LINE NUMBER
1061 004544 000302 SWAB R2
1062 004546 042702 177760 BIC #177760,R2
1063 004552 010203 MOV R2,R3 ;SAVE LINE NUMBER
1064 004554 006302 ASL R2 ;USE LINE NUMBER AS OFFSET

```


1065	004556	026201	012312			CMP	RBUF(R2),R1	;COMPARE EXPECTED AND RECEIVED DATA
1066	004562	001403				BEG	.+10	
1067	004564	016205	012312			MOV	RBUF(R2),R5	;GET EXPECTED DATA
1068	004570	104003				HLT	3	;DATA ERROR
1069	004572	105262	012312			INCB	RBUF(R2)	;UPDATE EXPECTED CHARACTER
1070	004576	000002				RTI		;CONTINUE
1071	004600	104001			10\$:	HLT	1	;CHARACTER AVAILABLE NOT SET, ERROR
1072	004602	022626			11\$:	POP2SP		;RESTORE STACK
1073	004604	012777	004000	004774		MOV	#BIT11,JDHSCR	;MASTER CLEAR INTERFACE
1074	004612	000411				BR	13\$;RESTART TEST
1075								
1076								
1077								
1078								
1079								
1080								
1081								
1082	004614	005777	005000		12\$:	TST	JDHBAR	;WAIT FOR ALL BAR BITS TO CLEAR
1083	004620	001375				BNE	.-4	
1084	004622	105777	005000			TSTB	JDHSLR	;WAIT FOR SILO TO EMPTY
1085	004626	001375				BNE	.-4	
1086	004630	012767	000340	173140		MOV	#340,PS ;PREVENT INTERRUPTS	
1087	004636	104400			13\$:	SCOPE		;CHECK FOR ITERATIONS, LOOP
1088								
1089								
1090								
1091								
1092								
1093								
1094								
1095								
1096								
1097								
1098								
1099	004640	012767	000340	173130	T22:	MOV	#340,PS	;DISABLE ALL INTERRUPTS
1100	004646	012767	000010	005000		MOV	#10,ICOUNT	;SET UP FOR 10 ITERATIONS
1101	004654	012767	005106	004756		MOV	#11\$,ESCAPE	;SET UP TO ESCAPE TO NEXT TEST
1102	004662	012777	004000	004716		MOV	#BIT11,JDHSCR	;MASTER CLEAR INTERFACE
1103	004670	012700	000020			MOV	#16.,R0	;SET UP TO START 16. LINES
1104	004674	005001				CLR	R1	;COUNT AND BUS ADDRESS MEMORIES
1105								;AND RECEIVED DATA STORAGE
1106	004676	012702	000200			MOV	#200,R2	;SET UP TO LOAD HIGH BYTE
1107	004702	012703	000001			MOV	#1,R3	;OF EXPECTED DATA
1108	004706	012777	011712	004700	1\$:	MOV	#TBUF,JDHBA	;LOAD BUSS ADDRESS
1109	004714	012777	177400	004674		MOV	#-400,JDHBC	;LOAD BYTE COUNT
1110	004722	012777	027323	004662		MOV	#27323,JDHLPR	;SET LINE SPEED AND
1111								;CHARACTER LENGTH 8 + EVEN PARITY
1112	004730	105061	012312			CLRB	RBUF(R1)	;CLEAR NEXT RECEIVED CHARACTER
1113	004734	110263	012312			MOVB	R2,RBUF(R3)	;LOAD HIGH BYTE
1114	004740	005277	004642			INC	JDHSCR	;ADVANCE TO NEXT LINE
1115	004744	005202				INC	R2	;UPDATE POINTER TO
1116	004746	062701	000002			ADD	#2,R1	;LOW AND HIGH BYTE OF
1117	004752	062703	000002			ADD	#2,R3	;NEXT EXPECTED CHARACTER
1118	004756	005300				DEC	R0	;CONTINUE IF ALL LINES
1119	004760	001352				BNE	1\$;NOT SET UP
1120	004762	012777	005022	004640		MOV	#6\$,JDHRVEC	;SET UP POINTER FOR

1177										
1178	005144	012767	000340	172624	T23:	MOV	#340,PS		:DISABLE ALL INTERRUPTS	
1179	005152	012767	000010	004474		MOV	#10,ICOUNT		:SET UP FOR 10 ITERATIONS	
1180	005160	012767	005412	004462		MOV	#11\$,ESCAPE		:SET UP TO ESCAPE TO NEXT TEST	
1181	005166	012777	004000	004412		MOV	#BIT11,ADHSCR		:MASTER CLEAR INTERFACE	
1182	005174	012700	000020			MOV	#16.,R0		:SET UP TO START 16. LINES	
1183	005200	005001				CLR	R1		:COUNT AND BUS ADDRESS MEMORIES	
1184									:AND RECEIVED DATA STORAGE	
1185	005202	012702	000200			MOV	#200,R2		:SET UP TO LOAD HIGH BYTE	
1186	005206	012703	000001			MOV	#1,R3		:OF EXPECTED DATA	
1187	005212	012777	011712	004374	1\$:	MOV	#TBUF,ADHBA		:LOAD BUSS ADDRESS	
1188	005220	012777	177600	004370		MOV	#-200,ADHBC		:LOAD BYTE COUNT	
1189	005226	012777	027362	004356		MOV	#27362,ADHLPR		:SET LINE SPEED AND	
1190									:CHARACTER LENGTH 7 + ODD PARITY	
1191	005234	105061	012312			CLRB	RBUF(R1)		:CLEAR NEXT RECEIVED CHARACTER	
1192	005240	110263	012312			MOVB	R2,RBUF(R3)		:LOAD HIGH BYTE	
1193	005244	005277	004336			INC	ADHSCR		:ADVANCE TO NEXT LINE	
1194	005250	005202				INC	R2		:UPDATE POINTER TO	
1195	005252	062701	000002			ADD	#2,R1		:LOW AND HIGH BYTE OF	
1196	005256	062703	000002			ADD	#2,R3		:NEXT EXPECTED CHARACTER	
1197	005262	005300				DEC	R0		:CONTINUE IF ALL LINES	
1198	005264	001352				BNE	1\$:NOT SET UP	
1199	005266	012777	005326	004334		MOV	#6\$,ADHRVEC		:SET UP POINTER FOR	
1200	005274	012777	000240	004330		MOV	#240,ADHRLVL		:RECEIVER INTERRUPT	
1201	005302	012777	000100	004276		MOV	#100,ADHSCR		:ENABLE RECEIVER INTERRUPT	
1202	005310	012777	177777	004302		MOV	#-1,ADHBAR		:SET ALL BAR BITS	
1203	005316	005067	172454			CLR	PS		:ENABLE INTERRUPTS	
1204	005322	000167	000076			JMP	12\$:GO TO TRANSMITTER WAITING ROUTINE	
1205										
1206									:RECEIVER INTERRUPT SERVICE ROUTINE	
1207									:CHECK FOR RECEIVER DONE AND NO ERRORS	
1208									:CHECK RECEIVED DATA FOR "VALID DATA" FLAG AND NO ERRORS	
1209									:VERIFY THAT RECEIVED DATA IS CORRECT	
1210									:CHECK FOR END OF PASS	
1211										
1212	005326	105777	004254		6\$:	TSTB	ADHSCR		:IF CHARACTER AVAILABLE NOT SET, ERROR	
1213	005332	100026				BPL	10\$			
1214	005334	032777	040000	004244		BIT	#40000,ADHSCR		:IF SILO OVERRUN SET, ERROR	
1215	005342	001401				BEQ	.+4			
1216	005344	104002				HLT	2		:SILO OVERRUN, ERROR	
1217	005346	017701	004236		7\$:	MOV	ADHNRC,R1		:READ CHARACTER FROM SILO	
1218	005352	010102				MOV	R1,R2		:EXTRACT LINE NUMBER	
1219	005354	000302				SWAB	R2			
1220	005356	042702	177760			BIC	#177760,R2			
1221	005362	010203				MOV	R2,R3		:SAVE LINE NUMBER	
1222	005364	006302				ASL	R2		:USE LINE NUMBER AS OFFSET	
1223	005366	026201	012312			CMP	RBUF(R2),R1		:COMPARE EXPECTED AND RECEIVED DATA	
1224	005372	001403				BEQ	.+10			
1225	005374	016205	012312			MOV	RBUF(R2),R5		:GET EXPECTED DATA	
1226	005400	104003				HLT	3		:DATA ERROR	
1227	005402	105262	012312			INCB	RBUF(R2)		:UPDATE EXPECTED CHARACTER	
1228	005406	000002				RTI			:CONTINUE	
1229	005410	104001			10\$:	HLT	1		:CHARACTER AVAILABLE NOT SET, ERROR	
1230	005412	022626			11\$:	POP2SP			:RESTORE STACK	
1231	005414	012777	004000	004154		MOV	#BIT11,ADHSCR		:MASTER CLEAR INTERFACE	
1232	005422	000411				BR	13\$:RESTART TEST	

M02

DZDHG MACY11 27(732) 29-SEP-76 15:16 PAGE 26
 DZDHGB.P11

```

1233
1234
1235 ;WAIT FOR ALL BAR BITS TO CLEAR
1236 ;WHEN ALL BAR BITS HAVE CLEARED,
1237 ;WAIT FOR SILO TO EMPTY
1238 ;WHEN SILO IS EMPTY, SCOPE ON TEST
1239
1240 005424 005777 004170 12$: TST  @DHBAR ;WAIT FOR ALL BAR BITS TO CLEAR
1241 005430 001375 BNE  -4
1242 005432 105777 004170 TSTB @DHSLR ;WAIT FOR SILO TO EMPTY
1243 005436 001375 BNE  -4
1244 005440 012767 000340 172330 MOV  #340,PS ;PREVENT INTERRUPTS
1245 005446 104400 13$: SCOPE ;CHECK FOR ITERATIONS, LOOP
1246
1247 ;MULTI-LINE PARITY DATA TEST
1248 ;TRANSMIT A BINARY COUNT WITH EVEN PARITY
1249 ;PATTERN ON ALL LINES.
1250 ;SILO ALARM LEVEL SET TO 0
1251 ;RECEIVER WILL BE SERVICED ON A PER CHARACTER BASIS
1252 ;IN INTERRUPT MODE.
1253 ;TRANSMITTER INTERRUPT WILL BE DISABLED
1254 ;LINE SPEED: 2400 BAUD
1255 ;CHARACTER LENGTH: 7 + PARITY
1256
1257 005450 012767 000340 172320 724: MOV  #340,PS ;DISABLE ALL INTERRUPTS
1258 005456 012767 000010 004170 MOV  #10,I'COUNT ;SET UP FOR 10 ITERATIONS
1259 005464 012767 005716 004156 MOV  #11$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
1260 005472 012777 004000 004106 MOV  #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
1261 005500 012700 000020 MOV  #16.,R0 ;SET UP TO START 16. LINES
1262 005504 005001 CLR  R1 ;COUNT AND BUS ADDRESS MEMORIES
1263 ;AND RECEIVED DATA STORAGE
1264 005506 012702 000200 MOV  #200,R2 ;SET UP TO LOAD HIGH BYTE
1265 005512 012703 000001 MOV  #1,R3 ;OF EXPECTED DATA
1266 005516 012777 011712 004070 1$: MOV  #TBUF,@DHBA ;LOAD BUSS ADDRESS
1267 005524 012777 177600 004064 MOV  #-200,@DHBC ;LOAD BYTE COUNT
1268 005532 012777 027322 004052 MOV  #27322,@DHLPR ;SET LINE SPEED AND
1269 ;CHARACTER LENGTH 7 + EVEN PARITY
1270 005540 105061 012312 CLR  RBUF(R1) ;CLEAR NEXT RECEIVED CHARACTER
1271 005544 110263 012312 MOV  R2,RBUF(R3) ;LOAD HIGH BYTE
1272 005550 005277 004032 INC  @DHSCR ;ADVANCE TO NEXT LINE
1273 005554 005202 INC  R2 ;UPDATE POINTER TO
1274 005556 062701 000002 ADC  #2,R1 ;LOW AND HIGH BYTE OF
1275 005562 062703 000002 ADD  #2,R3 ;NEXT EXPECTED CHARACTER
1276 005566 005300 DEC  R0 ;CONTINUE IF ALL LINES
1277 005570 001352 BNE  1$ ;NOT SET UP
1278 005572 012777 005632 004030 MOV  #6$,@DHAVEC ;SET UP POINTER FOR
1279 005600 012777 000240 004024 MOV  #240,@DHLVL ;RECEIVER INTERRUPT
1280 005606 012777 000100 003772 MOV  #100,@DHSCR ;ENABLE RECEIVER INTERRUPT
1281 005614 012777 177777 003776 MOV  #-1,@DHBAR ;SET ALL BAR BITS
1282 005622 005067 172150 CLR  PS ;ENABLE INTERRUPTS
1283 005626 000167 000076 JMP  12$ ;G' TO TRANSMITTER WAITING ROUTINE
1284
1285 ;RECEIVER INTERRUPT SERVICE ROUTINE
1286 ;CHECK FOR RECEIVER DONE AND NO ERRORS
1287 ;CHECK RECEIVED DATA FOR "VALID DATA" FLAG AND NO ERRORS
1288 ;VERIFY THAT RECEIVED DATA IS CORRECT

```



```

1345 006022 012777 011712 003564 1$: MOV #TBUF, @DHBM ;LOAD BUSS ADDRESS
1346 006030 012777 177700 003560 MOV #-100, @DHBC ;LOAD BYTE COUNT
1347 006036 012777 027361 003546 MOV #27361, @DHLPR ;SET LINE SPEED AND
; CHARACTER LENGTH 6 + ODD PARITY
1348
1349 006044 105061 012312 CLRS RBUF(R1) ;CLEAR NEXT RECEIVED CHARACTER
1350 006050 110263 012312 MOV# R2, RBUF(R3) ;LOAD HIGH BYTE
1351 006054 005277 003526 INC @DHSCR ;ADVANCE TO NEXT LINE
1352 006060 005202 INC R2 ;UPDATE POINTER TO
1353 006062 062701 000002 ADD #2, R1 ;LOW AND HIGH BYTE OF
1354 006066 062703 000002 ADD #2, R3 ;NEXT EXPECTED CHARACTER
1355 006072 005300 DEC R0 ;CONTINUE IF ALL LINES
1356 006074 001352 BNE 1$ ;NOT SET UP
1357 006076 012777 006136 003524 MOV #6$, @DHVEC ;SET UP POINTER FOR
1358 006104 012777 000240 003520 MOV #240, @DHLVL ;RECEIVER INTERRUPT
1359 006112 012777 000100 003466 MOV #100, @DHSCR ;ENABLE RECEIVER INTERRUPT
1360 006120 012777 177777 003472 MOV #-1, @DHBAR ;SET ALL BAR BITS
1361 006126 005067 171644 CLR PS ;ENABLE INTERRUPTS
1362 006132 000167 000076 JMP 12$ ;GO TO TRANSMITTER WAITING ROUTINE
1363
1364 ;RECEIVER INTERRUPT SERVICE ROUTINE
1365 ;CHECK FOR RECEIVER DONE AND NO ERRORS
1366 ;CHECK RECEIVED DATA FOR "VALID DATA" FLAG AND NO ERRORS
1367 ;VERIFY THAT RECEIVED DATA IS CORRECT
1368 ;CHECK FOR END OF PASS
1369
1370 006136 105777 003444 6$: TSTB @DHSCR ;IF CHARACTER AVAILABLE NOT SET, ERROR
1371 006142 100026 BPL 10$
1372 006144 032777 040000 003434 BIT #40000, @DHSCR ;IF SILO OVERRUN SET, ERROR
1373 006152 001401 BEQ .+4
1374 006154 104002 HLT 2 ;SILO OVERRUN, ERROR
1375 006156 017701 003426 7$: MOV @DHNRC, R1 ;READ CHARACTER FROM SILO
1376 006162 010102 MOV R1, R2 ;EXTRACT LINE NUMBER
1377 006164 000302 SWAB R2
1378 006166 042702 177760 BIC #177760, R2
1379 006172 010203 MOV R2, R3 ;SAVE LINE NUMBER
1380 006174 006302 ASL R2 ;USE LINE NUMBER AS OFFSET
1381 006176 026201 012312 CMP RBUF(R2), R1 ;COMPARE EXPECTED AND RECEIVED DATA
1382 006202 001403 BEQ .+10
1383 006204 016205 012312 MOV RBUF(R2), R5 ;GET EXPECTED DATA
1384 006210 104003 HLT 3 ;DATA ERROR
1385 006212 105262 012312 INCB RBUF(R2) ;UPDATE EXPECTED CHARACTER
1386 006216 000002 RTI ;CONTINUE
1387 006220 104001 10$: HLT 1 ;CHARACTER AVAILABLE NOT SET, ERROR
1388 006222 022626 11$: POP2SP ;RESTORE STACK
1389 006224 012777 004000 003354 MOV #BIT11, @DHSCR ;MASTER CLEAR INTERFACE
1390 006232 000411 BR 13$ ;RESTART TEST
1391
1392
1393 ;WAIT FOR ALL BAR BITS TO CLEAR
1394 ;WHEN ALL BAR BITS HAVE CLEARED,
1395 ;WAIT FOR SILO TO EMPTY
1396 ;WHEN SILO IS EMPTY, SCOPE ON TEST
1397
1398 006234 005777 003360 12$: TST @DHBAR ;WAIT FOR ALL BAR BITS TO CLEAR
1399 006240 001375 BNE .-4
1400 006242 105777 003360 TSTB @DHSLR ;WAIT FOR SILO TO EMPTY

```



```

1401 006246 001375          BNE      -4
1402 006250 012767 000340 171520  MOV      #340,PS ;PREVENT INTERRUPTS
1403 006256 104400          13$:    SCOPE      ;CHECK FOR ITERATIONS, LOOP
1404
1405          ;MULTI-LINE PARITY DATA TEST
1406          ;TRANSMIT A BINARY COUNT WITH EVEN PARITY
1407          ;PATTERN ON ALL LINES.
1408          ;SILO ALARM LEVEL SET TO 0
1409          ;RECEIVER WILL BE SERVICED ON A PER CHARACTER BASIS
1410          ;IN INTERRUPT MODE.
1411          ;TRANSMITTER INTERRUPT WILL BE DISABLED
1412          ;LINE SPEED: 2400 BAUD
1413          ;CHARACTER LENGTH: 6 + PARITY
1414
1415 006260 012767 000340 171510 T26:    MOV      #340,PS      ;DISABLE ALL INTERRUPTS
1416 006266 012767 000010 003360      MOV      #10,COUNT   ;SET UP FOR 10 ITERATIONS
1417 006274 012767 006526 003346      MOV      #11$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
1418 006302 012777 004000 003276      MOV      #BIT11,ADHSCR ;MASTER CLEAR INTERFACE
1419 006310 012700 000020          MOV      #16.,R0     ;SET UP TO START 16. LINES
1420 006314 005001          CLR      R1          ;COUNT AND BUS ADDRESS MEMORIES
1421          ;AND RECEIVED DATA STORAGE
1422 006316 012702 000200          MOV      #200,R2    ;SET UP TO LOAD HIGH BYTE
1423 006322 012703 000001          MOV      #1,R3      ;OF EXPECTED DATA
1424 006326 012777 011712 003260 15:    MOV      #TBUF,ADHBA ;LOAD BUSS ADDRESS
1425 006334 012777 177700 003254      MOV      #-100,ADHBC ;LOAD BYTE COUNT
1426 006342 012777 027321 003242      MOV      #27321,ADHLPR ;SET LINE SPEED AND
1427          ;CHARACTER LENGTH 6 + EVEN PARITY
1428 006350 105061 012312          CLRB    RBUF(R1)    ;CLEAR NEXT RECEIVED CHARACTER
1429 006354 110263 012312          MOV     R2,RBUF(R3) ;LOAD HIGH BYTE
1430 006360 005277 003222          INC     ADHSCR      ;ADVANCE TO NEXT LINE
1431 006364 005202          INC     R2          ;UPDATE POINTER TO
1432 006366 062701 000002          ADD     #2,R1       ;LOW AND HIGH BYTE OF
1433 006372 062703 000002          ADD     #2,R3       ;NEXT EXPECTED CHARACTER
1434 006376 005300          DEC     R0          ;CONTINUE IF ALL LINES
1435 006400 001352          BNE     1$          ;NOT SET UP
1436 006402 012777 006442 003220      MOV     #6$,ADHRVEC ;SET UP POINTER FOR
1437 006410 012777 000240 003214      MOV     #240,ADHRLVL ;RECEIVER INTERRUPT
1438 006416 012777 000100 003162      MOV     #100,ADHSCR ;ENABLE RECEIVER INTERRUPT
1439 006424 012777 177777 003166      MOV     #-1,ADHBAR  ;SET ALL BAR BITS
1440 006432 005067 171340          CLR     PS          ;ENABLE INTERRUPTS
1441 006436 000167 000076          JMP     12$         ;GO TO TRANSMITTER WAITING ROUTINE
1442
1443          ;RECEIVER INTERRUPT SERVICE ROUTINE
1444          ;CHECK FOR RECEIVER DONE AND NO ERRORS
1445          ;CHECK RECEIVED DATA FOR "VALID DATA" FLAG AND NO ERRORS
1446          ;VERIFY THAT RECEIVED DATA IS CORRECT
1447          ;CHECK FOR END OF PASS
1448
1449 006442 105777 003140          6$:    TSTB    ADHSCR      ;IF CHARACTER AVAILABLE NOT SET, ERROR
1450 006446 100026          BPL     10$
1451 006450 032777 040000 003130      BIT     #40000,ADHSCR ;IF SILO OVERRUN SET, ERROR
1452 006456 001401          BEQ     +4
1453 006460 104002          HLT     2
1454 006462 017701 003122          7$:    MOV     ADHNR, R1   ;SILO OVERRUN, ERROR
1455 006466 010102          MOV     R1,R2      ;READ CHARACTER FROM SILO
1456 006470 000302          SWAB   R2          ;EXTRACT LINE NUMBER

```

1457	006472	042702	177760		BIC	#177760,R2	
1458	006476	010203			MOV	R2,R3	;SAVE LINE NUMBER
1459	006500	006302			ASL	R2	;USE LINE NUMBER AS OFFSET
1460	006502	026201	012312		CMF	RBUF(R2),R1	;COMPARE EXPECTED AND RECEIVED DATA
1461	006506	001403			BEQ	+10	
1462	006510	016205	012312		MOV	RBUF(R2),R5	;GET EXPECTED DATA
1463	006514	104003			HLT	3	;DATA ERROR
1464	006516	105262	012312		INCB	RBUF(R2)	;UPDATE EXPECTED CHARACTER
1465	006522	000002			RTI		;CONTINUE
1466	006524	104001			HLT	1	;CHARACTER AVAILABLE NOT SET, ERROR
1467	006526	022626		10\$:	POP2SP		;RESTORE STACK
1468	006530	012777	004000	003050	MOV	#BIT11,JDHSCR	;MASTER CLEAR INTERFACE
1469	006536	000411			BR	13\$;RESTART TEST
1470							
1471							
1472							
1473							
1474							
1475							
1476							
1477	006540	005777	003054	12\$:	TST	JDHBAR	;WAIT FOR ALL BAR BITS TO CLEAR
1478	006544	001375			BNE	-4	
1479	006546	105777	003054		TSTB	JDHSLR	;WAIT FOR SILO TO EMPTY
1480	006552	001375			BNE	-4	
1481	006554	012767	000340	171214	MOV	#340,PS ;PREVENT INTERRUPTS	
1482	006562	104400		13\$:	SCOPE		;CHECK FOR ITERATIONS, LOOP
1483							
1484							
1485							
1486							
1487							
1488							
1489							
1490							
1491							
1492							
1493							
1494	006564	012767	000340	171204	T27:	MOV	#340,PS ;DISABLE ALL INTERRUPTS
1495	006572	012767	000010	003054		MOV	#10,ICOUNT ;SET UP FOR 10 ITERATIONS
1496	006600	012767	007032	003042		MOV	#11\$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
1497	006606	012777	004000	002772		MOV	#BIT11,JDHSCR ;MASTER CLEAR INTERFACE
1498	006614	012700	000020			MOV	#16.,R0 ;SET UP TO START 16. LINES
1499	006620	005001				CLR	R1 ;COUNT AND BUS ADDRESS MEMORIES
1500							;AND RECEIVED DATA STORAGE
1501	006622	012702	000200			MOV	#200,R2 ;SET UP TO LOAD HIGH BYTE
1502	006626	012703	000001			MOV	#1,R3 ;OF EXPECTED DATA
1503	006632	012777	011712	002754	1\$:	MOV	#TBUF,JDHBA ;LOAD BUSS ADDRESS
1504	006640	012777	177740	002750		MOV	#-40,JDHBC ;LOAD BYTE COUNT
1505	006646	012777	027360	002736		MOV	#27360,JDHLPR ;SET LINE SPEED AND
1506							;CHARACTER LENGTH 5 + ODD PARITY
1507	006654	105061	012312			CLRB	RBUF(R1) ;CLEAR NEXT RECEIVED CHARACTER
1508	006660	110263	012312			MOV	R2,RBUF(R3) ;LOAD HIGH BYTE
1509	006664	005277	002716			INC	JDHSCR ;ADVANCE TO NEXT LINE
1510	006670	005202				INC	R2 ;UPDATE POINTER TO
1511	006672	062701	000002			ADD	#2,R1 ;LOW AND HIGH BYTE OF
1512	006676	062703	000002			ADD	#2,R3 ;NEXT EXPECTED CHARACTER

E03

DZDHG MACY11 27(732) 29-SEP-76 15:16 PAGE 31
DZDHGB.P11

```

1513 006702 005300          DEC      RD          ;CONTINUE IF ALL LINES
1514 006704 001352          BNE      1$          ;NOT SET UP
1515 006706 012777 006746 002714  MOV      #6$,DHRVEC ;SET UP POINTER FOR
1516 006714 012777 000240 002710  MOV      #240,DHRLVL ;RECEIVER INTERRUPT
1517 006722 012777 000100 002656  MOV      #100,DHSCR  ;ENABLE RECEIVER INTERRUPT
1518 006730 012777 177777 002662  MOV      #-1,DHBAR  ;SET ALL BAR BITS
1519 006736 005067 171034          CLR      PS          ;ENABLE INTERRUPTS
1520 006742 000167 000076          JMP      12$         ;GO TO TRANSMITTER WAITING ROUTINE
1521
1522          ;RECEIVER INTERRUPT SERVICE ROUTINE
1523          ;CHECK FOR RECEIVER DONE AND NO ERRORS
1524          ;CHECK RECEIVED DATA FOR "VALID DATA" FLAG AND NO ERRORS
1525          ;VERIFY THAT RECEIVED DATA IS CORRECT
1526          ;CHECK FOR END OF PASS
1527
1528 006746 105777 002634      6$:  TSTB     DHSCR          ;IF CHARACTER AVAILABLE NOT SET, ERROR
1529 006752 100026          BPL      10$         ;
1530 006754 032777 040000 002624  BIT      #40000,DHSCR ;IF SILO OVERRUN SET, ERROR
1531 006762 001401          BEQ      +4          ;
1532 006764 104002          HLT      2           ;SILO OVERRUN, ERROR
1533 006766 017701 002616      7$:  MOV      DHNRC,R1    ;READ CHARACTER FROM SILO
1534 006772 010102          MOV      R1,R2      ;EXTRACT LINE NUMBER
1535 006774 000302          SWAB    R2          ;
1536 006776 042702 177760  BIC      #177760,R2 ;
1537 007002 010203          MOV      R2,R3      ;SAVE LINE NUMBER
1538 007004 006302          ASL     R2          ;USE LINE NUMBER AS OFFSET
1539 007006 026201 012312  CMP      RBUF(R2),R1 ;COMPARE EXPECTED AND RECEIVED DATA
1540 007012 001403          BEQ     +10         ;
1541 007014 016205 012312  MOV      RBUF(R2),R5 ;GET EXPECTED DATA
1542 007020 104003          HLT     3           ;DATA ERROR
1543 007022 105262 012312  INCB    RBUF(R2)    ;UPDATE EXPECTED CHARACTER
1544 007026 000002          RTI     ;CONTINUE
1545 007030 104001          10$:  HLT     1           ;CHARACTER AVAILABLE NOT SET, ERROR
1546 007032 022626          11$:  POP2SP ;RESTORE STACK
1547 007034 012777 004000 002544  MOV      #BIT11,DHSCR ;MASTER CLEAR INTERFACE
1548 007042 000411          BR      13$         ;RESTART TEST
1549
1550
1551          ;WAIT FOR ALL BAR BITS TO CLEAR
1552          ;WHEN ALL BAR BITS HAVE CLEARED,
1553          ;WAIT FOR SILO TO EMPTY
1554          ;WHEN SILO IS EMPTY, SCOPE ON TEST
1555
1556 007044 005777 002550      12$: TST      DHBAR          ;WAIT FOR ALL BAR BITS TO CLEAR
1557 007050 001375          BNE     -4          ;
1558 007052 105777 002550      TSTB    DHSLR          ;WAIT FOR SILO TO EMPTY
1559 007056 001375          BNE     -4          ;
1560 007060 012767 000340 170710  MOV      #340,PS ;PREVENT INTERRUPTS
1561 007066 104400          13$:  SCOPE          ;CHECK FOR ITERATIONS, LOOP
1562
1563          ;MULTI-LINE PARITY DATA TEST
1564          ;TRANSMIT A BINARY COUNT WITH EVEN PARITY
1565          ;PATTERN ON ALL LINES.
1566          ;SILO ALARM LEVEL SET TO 0
1567          ;RECEIVER WILL BE SERVICED ON A PER CHARACTER BASIS
1568          ;IN INTERRUPT MODE.

```

F03

DZDHG MACY11 27(732) 28-SEP-76 15:16 PAGE 32
 DZDHGB.P11

```

1569                                     ; TRANSMITTER INTERRUPT WILL BE DISABLED
1570                                     ; LINE SPEED: 2400 BAUD
1571                                     ; CHARACTER LENGTH: 5 + PARITY
1572
1573 007070 012767 000340 170700 T30:  MOV    #340,PS          ; DISABLE ALL INTERRUPTS
1574 007076 012767 000010 002550      MOV    #10,I,COUNT      ; SET UP FOR 10 ITERATIONS
1575 007104 012767 007336 002536      MOV    #11$,ESCAPE     ; SET UP TO ESCAPE TO NEXT TEST
1576 007112 012777 004000 002466      MOV    #BIT11,JDHSCR   ; MASTER CLEAR INTERFACE
1577 007120 012700 000020              MOV    #16.,R0         ; SET UP TO START 16. LINES
1578 007124 005001                    CLR    R1              ; COUNT AND BUS ADDRESS MEMORIES
1579                                     ; AND RECEIVED DATA STORAGE
1580 007126 012702 000200              MOV    #200,R2         ; SET UP TO LOAD HIGH BYTE
1581 007132 012703 000001              MOV    #1,R3          ; OF EXPECTED DATA
1582 007136 012777 011712 002450 1$:  MOV    #TBUF,JDHBA     ; LOAD BUSS ADDRESS
1583 007144 012777 177740 002444      MOV    #-40,JDHBC     ; LOAD BYTE COUNT
1584 007152 012777 027320 002432      MOV    #27320,JDHLPR  ; SET LINE SPEED AND
1585                                     ; CHARACTER LENGTH 5 + EVEN PARITY
1586 007160 105061 012312              CLRB   RBUF(R1)       ; CLEAR NEXT RECEIVED CHARACTER
1587 007164 110263 012312              MOVB  R2,RBUF(R3)    ; LOAD HIGH BYTE
1588 007170 005277 002412              INC    JDHSCR         ; ADVANCE TO NEXT LINE
1589 007174 005202                    INC    R2             ; UPDATE POINTER TO
1590 007176 062701 000002              ADD    #2,R1         ; LOW AND HIGH BYTE OF
1591 007202 062703 000002              ADD    #2,R3         ; NEXT EXPECTED CHARACTER
1592 007206 005300                    DEC    R0             ; CONTINUE IF ALL LINES
1593 007210 001352                    BNE   1$             ; NOT SET UP
1594 007212 012777 007252 002410      MOV    #6$,JDHRVEC    ; SET UP POINTER FOR
1595 007220 012777 000240 002404      MOV    #240,JDHRLVL  ; RECEIVER INTERRUPT
1596 007226 012777 000100 002352      MOV    #100,JDHSCR   ; ENABLE RECEIVER INTERRUPT
1597 007234 012777 177777 002356      MOV    #-1,JDHBAR    ; SET ALL BAR BITS
1598 007242 005067 170530              CLR    PS            ; ENABLE INTERRUPTS
1599 007246 000167 000076              JMP    12$          ; GO TO TRANSMITTER WAITING ROUTINE
1600
1601                                     ; RECEIVER INTERRUPT SERVICE ROUTINE
1602                                     ; CHECK FOR RECEIVER DONE AND NO ERRORS
1603                                     ; CHECK RECEIVED DATA FOR "VALID DATA" FLAG AND NO ERRORS
1604                                     ; VERIFY THAT RECEIVED DATA IS CORRECT
1605                                     ; CHECK FOR END OF PASS
1606
1607 007252 105777 002330 6$:          TSTB   JDHSCR         ; IF CHARACTER AVAILABLE NOT SET, ERROR
1608 007256 100026                    BPL   10$           ;
1609 007260 032777 040000 002320      BIT    #40000,JDHSCR ; IF SILO OVERRUN SET, ERROR
1610 007266 001401                    BEQ   .+4           ;
1611 007270 104002                    HLT   2             ; SILO OVERRUN, ERROR
1612 007272 017701 002312 7$:        MOV    JDHNR,C,R1    ; READ CHARACTER FROM SILO
1613 007276 010102                    MOV    R1,R2        ; EXTRACT LINE NUMBER
1614 007300 000302                    SWAB  R2            ;
1615 007302 042702 177760              BIC   #177760,R2    ;
1616 007306 010203                    MOV    R2,R3        ;
1617 007310 006302                    ASL   R2            ; SAVE LINE NUMBER
1618 007312 026201 012312              CMP    RBUF(R2),R1  ; USE LINE NUMBER AS OFFSET
1619 007316 001403                    BEQ   .+10          ; COMPARE EXPECTED AND RECEIVED DATA
1620 007320 016205 012312              MOV    RBUF(R2),R5  ; GET EXPECTED DATA
1621 007324 104003                    HLT   3             ; DATA ERROR
1622 007326 105262 012312              INCB  RBUF(R2)      ; UPDATE EXPECTED CHARACTER
1623 007332 000002                    RTI                    ; CONTINUE
1624 007334 104001 10$:              HLT    1            ; CHARACTER AVAILABLE NOT SET, ERROR

```


G03

DZDHG MACY11 27(732) 29-SEP-76 15:16 PAGE 33
 DZDHGB.P11

1625	007336	022626		11\$:	POP2SP		;RESTORE STACK
1626	007340	012777	004000 002240		MOV	#BIT11, @DHSCR	;MASTER CLEAR INTERFACE
1627	007346	000411			BR	13\$;RESTART TEST
1628							
1629							
1630							
1631							
1632							
1633							
1634							
1635	007350	005777	002244	12\$:	TST	@DHBAR	;WAIT FOR ALL BAR BITS TO CLEAR
1636	007354	001375			BNE	-4	
1637	007356	105777	002244		TSTB	@DHSLR	;WAIT FOR SILO TO EMPTY
1638	007362	001375			BNE	-4	
1639	007364	012767	000340 170404		MOV	#340, PS ;PREVENT INTERRUPTS	
1640	007372	104400		13\$:	SCOPE		;CHECK FOR ITERATIONS, LOOP

H03

DZDHG MACY11 27(732) 29-SEP-76 15:16 PAGE 34
 DZDHGB.P11

```

1641
1642 ;MULTI LINE DATA TEST
1643 ;SILO ALARM LEVEL SET TO 0
1644 ;RECEIVER WILL BE SERVICED ON A PER CHARACTER BASIS
1645 ;IN INTERRUPT MODE
1646 ;TRANSMITTER INTERRUPTS WILL BE DISABLED
1647 ;TRANSMIT A BINARY COUNT PATTERN ON ALL LINES
1648 ;LINE SPEED IS 4800 BAUD FOR ALL LINES
1649 ;CHARACTER LENGTH IS 8 BITS FOR ALL LINES
1650
1651 007374 012767 000340 170374 T31: MOV #340,PS ;DISABLE ALL INTERRUPTS
1652 007402 012767 000010 002244 MOV #10,I,COUNT ;SET UP FOR 10 ITERATIONS
1653 007410 012767 007642 002232 MOV #11$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
1654 007416 012777 004000 002162 MOV #BIT11,JDHSCR ;MASTER CLEAR INTERFACE
1655 007424 012700 000020 MOV #20,R0 ;SET UP TO LOAD BYTE
1656 007430 005001 CLR R1 ;COUNT AND BUS ADDRESS MEMORIES
1657 ;AND RECEIVED DATA STORAGE
1658 007432 012702 000200 MOV #200,R2 ;SET UP TO LOAD HIGH BYTE
1659 007436 012703 000001 MOV #1,R3 ;OF EXPECTED DATA
1660 007442 012777 011712 002144 1$: MOV #TBUF,JDHBA ;LOAD BUSS ADDRESS
1661 007450 012777 177400 002140 MOV #-400,JDHBC ;LOAD BYTE COUNT
1662 007456 012777 031403 002126 MOV #31403,JDHLPR ;SET LINE SPEED AND
1663 ;CHARACTER LENGTH
1664 007464 105061 012312 CLRB RBUF(R1) ;CLEAR NEXT RECEIVED CHARACTER
1665 007470 110263 012312 MOV R2,RBUF(R3) ;LOAD HIGH BYTE
1666 007474 005277 002106 INC JDHSCR ;ADVANCE TO NEXT LINE
1667 007500 005202 INC R2 ;UPDATE POINTER TO
1668 007502 062701 000002 ADD #2,R1 ;LOW AND HIGH BYTE OF
1669 007506 062703 000002 ADD #2,R3 ;NEXT EXPECTED CHARACTER
1670 007512 005300 DEC R0 ;CONTINUE IF ALL LINES
1671 007514 001352 BNE 1$ ;NOT SET UP
1672 007516 012777 007556 002104 MOV #6$,JDHRVEC ;SET UP POINTER FOR
1673 007524 012777 000240 002100 MOV #240,JDHRLVL ;RECEIVER INTERRUPT
1674 007532 012777 000100 002046 MOV #100,JDHSCR ;ENABLE INTERFACE
1675 007540 012777 177777 002052 MOV #-1,JDHBAR ;SET ALL BAR BITS
1676 007546 005067 170224 CLR PS ;ENABLE INTERRUPTS
1677 007552 000167 000076 JMP 12$ ;GO TO TRANSMITTER WAITING ROUTINE
1678
1679 ;RECEIVER INTERRUPT SERVICE ROUTINE
1680 ;CHECK FOR RECEIVER DONE AND NO ERRORS
1681 ;CHECK RECEIVED DATA FOR "VALID DATA" FLAG AND NO ERRORS
1682 ;VERIFY THAT RECEIVED DATA IS CORRECT
1683 ;CHECK FOR END OF PASS
1684
1685 007556 105777 002024 6$: TSTB JDHSCR ;IF CHARACTER AVAILABLE NOT SET, ERROR
1686 007562 100026 BPL 10$
1687 007564 032777 040000 002014 BIT #40000,JDHSCR ;IF SILO OVERRUN SET, ERROR
1688 007572 001401 BEQ .+4
1689 007574 104002 HLT 2 ;SILO OVERRUN, ERROR
1690 007576 017701 002006 7$: MOV JDHNRC,R1 ;READ CHARACTER FORM SILO
1691 007602 010102 MOV R1,R2 ;EXTRACT LINE NUMBER
1692 007604 000302 SWAB R2
1693 007606 042702 177760 BIC #177760,R2
1694 007612 010203 MOV R2,R3 ;SAVE LINE NUMBER
1695 007614 006302 ASL R2 ;USE LINE NUMBER AS OFFSET
1696 007616 026201 012312 CMP RBUF(R2),R1 ;COMPARE EXPECTED AND RECEIVED DATA

```


1697	007622	001403			BEO	.+10		
1698	007624	016205	012312		MOV	RBUF(R2),R5		;GET EXPECTED DATA
1699	007630	104003			HLT	3		;DATA ERROR
1700	007632	105262	012312		INCB	RBUF(R2)		;UPDATE EXPECTED CHARACTER
1701	007636	000002			RTI			;CONTINUE
1702	007640	104001		10\$:	HLT	1		;CHARACTER AVAILABLE NOT SET, ERROR
1703	007642	022626		11\$:	POP2SP			;RESTORE STACK
1704	007644	012777	004000	001734	MOV	#BIT11,@DHSCR		;MASTER CLEAR INTERFACE
1705	007652	000406			BR	13\$;RESTART TEST
1706								
1707								
1708								
1709								
1710								
1711								
1712								
1713	007654	005777	001740	12\$:	TST	@DHBAR		;WAIT FOR ALL BAR BITS TO CLEAR
1714	007660	001375			BNE	.-4		
1715	007662	105777	001740		TSTB	@DHSLR		;WAIT FOR SILO TO EMPTY
1716	007666	001375			BNE	.-4		
1717	007670	104400		13\$:	SCOPE			;CHECK FOR ITERATIONS, LOOP

;WAIT FOR ALL BAR BITS TO CLEAR
 ;WHEN ALL BAR BITS HAVE CLEARED,
 ;WAIT FOR SILO TO EMPTY
 ;WHEN SILO IS EMPTY, SCOPE ON TEST

```

1718
1719
1720
1721
1722
1723
1724
1725
1726
1727 007672 012767 000340 170076 T32:
1728 007700 012767 000012 001746
1729 007706 012767 010240 001734
1730 007714 012777 004000 001664
1731 007722 012700 000020
1732 007726 005001
1733
1734 007730 012702 000200
1735 007734 012703 000001
1736 007740 012777 011712 001646 1$:
1737 007746 012777 177400 001642
1738 007754 012777 033503 001630
1739
1740 007762 105061 012312
1741 007766 110263 012312
1742 007772 005277 001610
1743 007776 005202
1744 010000 062701 000002
1745 010004 062703 000002
1746 010010 005300
1747 010012 001352
1748 010014 012777 010104 001612
1749 010022 012777 000240 001606
1750 010030 012777 010116 001572
1751 010036 012777 000240 001566
1752 010044 012700 000012
1753 010050 005067 001632
1754
1755 010054 012777 000077 001542
1756 010062 012777 030000 001516
1757 010070 012777 177777 001522
1758 010076 005067 167674
1759 010102 000421
1760
1761
1762
1763
1764 010104 005777 001476 2$:
1765 010110 100401
1766 010112 104004
1767 010114 000002
1768
1769
1770
1771 010116 105777 001464 6$:
1772 010122 100001
1773 010124 104005

```

```

;MULTI-LINE DATA TEST
;TRANSMIT A BINARY COUNT COUNT PATTERN ON ALL LINES
;LINE SPEED IS 9600 BAUD
;SILO ALARM LEVEL IS SET TO 63 (NO RECEIVER DONE
;INTERRUPTS WILL OCCUR)
;CONTINUOUSLY ATTEMPT TO READ DATA FROM SILO
;IF "VALID DATA" FLAG IS DETECTED,
;THE DATA WILL BE CHECKED
MOV #340,PS ;DISABLE ALL INTERRUPTS
MOV #12,I,COUNT ;SET UP FOR 12 ITERATIONS
MOV #8$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
MOV #BIT11,JDHSCR ;MASTER CLEAR INTERFACE
MOV #20,R0 ;SET UP TO LOAD BYTE
CLR R1 ;COUNT AND BUS ADDRESS MEMORIES
;AND RECEIVED DATA STORAGE
MOV #200,R2 ;SET UP TO LOAD HIGH BYTE
MOV #1,R3 ;OF EXPECTED DATA
MOV #TBUF,JDHBA ;LOAD BUSS ADDRESS
MOV #-400,JDHBC ;LOAD BYTE COUNT
MOV #33503,JDHLPR ;SET LINE SPEED AND
;CHARACTER LENGTH
CLRB RBUF(R1) ;CLEAR NEXT RECEIVED CHARACTER
MOVB R2,RBUF(R3) ;LOAD HIGH BYTE
INC JDHSCR ;ADVANCE TO NEXT LINE
INC R2 ;UPDATE POINTER TO
ADD #2,R1 ;LOW AND HIGH BYTE OF
ADD #2,R3 ;NEXT EXPECTED CHARACTER
DEC R0 ;CONTINUE IF ALL LINES
BNE 1$ ;NOT SET UP
MOV #2$,JDHTVEC ;SET UP POINTER FOR
MOV #240,JDHTLVL ;TRANSMITTER INTERRUPT
MOV #6$,JDHRVEC ;SET UP POINTER FOR
MOV #240,JDHRLVL ;RECEIVER INTERRUPT
MOV #12,R0 ;SET UP TO TRANSMIT
CLR ENDFLG ;10 (DEDIMAL) BLOCKS OF DATA
;ON EACH LINE
MOV #77,JDHSSR
MOV #30000,JDHSCR ;ENABLE INTERFACE
MOV #-1,JDHBAR ;SET ALL BAR BITS
CLR PS ;ENABLE INTERRUPTS
BR 7$ ;TRANSFER TO RECEIVER SERVICE

;TRANSMITTER INTERRUPT SERVICE ROUTINE
;CHECK FOR TRANSMIT DONE AND NO ERRORS
TST JDHSCR ;IS TRANSMITTER DONE SET
BMI .+4
HLT 4 ;TRANSMIT DONE NOT SET, ERROR
RTI ;RETURN

;RECEIVER INTERRUPT SERVICE ROUTINE
TSTB JDHSCR ;IS CHARACTER AVAILABLE SET
BPL .+4
HLT 5 ;CHARACTER AVAILABLE SET, ERROR

```


K03

DZDHG MACY11 27(732) 28-SEP-76 15:16 PAGE 37
 DZDHGB.P11

1774	010126	032777	040000	001452		BIT	#40000,JDHSCR		;IS SILO OVERRUN SET
1775	010134	001402				BEQ	.+6		
1776	010136	104002				HLT	2		;SILO OVERRUN, ERROR
1777	010140	000401				BR	3\$		
1778	010142	104000				HLT	0		;SPURIOUS INTERRUPT
1779									;TEST WILL RESTART
1780	010144	000002			3\$:	RTI			
1781									
1782									;RECEIVER DATA CHECK
1783									;GET A CHARACTER FROM THE SILO
1784									;IF IT IS VALID DATA, CHECK IT
1785									;OTHERWISE, TRY AGAIN
1786									
1787	010146	017701	001436		7\$:	MOV	JDHNR, R1		;READ NEXT RECEIVED CHARACTER REGISTER
1788	010152	005701				TST	R1		;IF VALID DATA BIT NOT SET
1789	010154	100374				BPL	7\$;TRY AGAIN
1790	010156	010102				MOV	R1, R2		;EXTRACT LINE NUMBER
1791	010160	000302				SWAB	R2		
1792	010162	042702	177760			BIC	#177760, R2		
1793	010166	010203				MOV	R2, R3		
1794	010170	006302				ASL	R2		;DOUBLE LINE NUMBER FOR OFFSET
1795	010172	126201	012312			CMPB	RBUF(R2), R1		;COMPARE RECEIVED AND EXPECTED RESULTS
1796	010176	001403				BEQ	.+10		
1797	010200	016205	012312			MOV	RBUF(R2), R5		;GET EXPECTED DATA
1798	010204	104003				HLT	3		;DATA ERROR
1799	010206	105262	012312			INCB	RBUF(R2)		;UPDATE EXPECTED DATA
1800	010212	001355				BNE	7\$;CONTINUE IF NOT 0
1801	010214	005767	001466			TST	ENDFLG		;IF ALL LINES HAVE FINISHED
1802	010220	001007				BNE	8\$;10 BLOCKS OF DATA, CLEAN UP
1803	010222	005703				TST	R3		;IF THE LINE IS 0
1804	010224	001014				BNE	9\$		
1805	010226	005300				DEC	R0		;UPDATE BLOCK COUNT
1806	010230	001012				BNE	9\$;IF LINE 0 HAS TRANSMITTED 10 BLOCKS
1807	010232	012767	000001	001446		MOV	#1, ENDFLG		;SET END FLAG
1808	010240	005777	001354		8\$:	TST	JDHBAR		;IF ALL LINES NOT DONE
1809	010244	001340				BNE	7\$;GET MORE CHARACTERS
1810	010246	105777	001354			TSTB	JDHSLR		;IF SILO IS NOT EMPTY
1811	010252	001335				BNE	7\$;GET REST OF DATA
1812	010254	000417				BR	10\$;WHEN EMPTY, SCOPE ON TEST
1813	010256	042777	000017	001322	9\$:	BIC	#17, JDHSCR		;ADDRESS LINE THAT JUST FINISHED
1814	010264	050377	001316			BIS	R3, JDHSCR		
1815	010270	012777	011712	001316		MOV	#TBUF, JDHBA		;SET UP BUS ADDRESS AND BYTE COUNT
1816	010276	012777	177400	001312		MOV	#-400, JDHBC		;FOR THAT LINE
1817	010304	056277	010322	001306		BIS	BARBIT(R2), JDHBAR		;SET BAR BIT FOR THAT LINE
1818	010312	000715				BR	7\$;CONTINUE
1819	010314	104400			10\$:	SCOPE			;CHECK FOR ITERATIONS, LOOP
1820	010316	000167	000040			JMP	EOP		;GO TO END OF PASS ROUTINE
1821	010322	000001			BARBIT:	1			
1822	010324	000002				2			
1823	010326	000004				4			
1824	010330	000010				10			
1825	010332	000020				20			
1826	010334	000040				40			
1827	010336	000100				100			
1828	010340	000200				200			
1829	010342	000400				400			

L03

DZDHG MACY11 27(732) 29-SEP-76 15:16 PAGE 38
DZDHGB.P11

1830	010344	001000	1000
1831	010346	002000	2000
1832	010350	004000	4000
1833	010352	010000	10000
1834	010354	020000	20000
1835	010356	040000	40000
1836	010360	100000	100000


```

1837
1838
1839
1840
1841
1842
1843
1844
1845 010362 104401
1846 010364 012765
1847 010366 005067 001312
1848 010372 005067 001242
1849 010376 005267 001240
1850 010402 016767 001234 167160
1851 010410 013701 000042
1852 010414 001405
1853 010416 000005
1854 010420 004711
1855 010422 000240
1856 010424 000240
1857 010426 000240
1858 010430 000167 170546
1859
1860
1861
1862
1863 010434 032767 002000 167126
1864 010442 001030
1865 010444 032767 040000 167116 1$:
1866 010452 001021
1867 010454 032767 004000 167106
1868 010462 001006
1869 010464 005267 001166
1870 010470 026767 001162 001156
1871 010476 001007
1872 010500 005067 001152 2$:
1873 010504 005067 001130
1874 010510 011667 001132
1875 010514 000002
1876 010516 016716 001124 3$:
1877 010522 000002
1878 010524 005767 001110 4$:
1879 010530 001745
1880 010532 000762
1881
1882
1883
1884 010534 032767 001000 167026 SCOP1R:
1885 010542 001402
1886 010544 016716 001102
1887 010550 000002 1$:
1888
1889
1890
1891 010552 032767 020000 167010 ERRORS:
1892 010560 001051

```

```

;END OF PASS
;TYPE NAME OF TEST
;UPDATE PASS COUNT
;CHECK FOR EXIT TO ACT-
;RESTART TEST

```

```

;TYPE NAME OF TEST
;CLEAR LAST ERROR PC
;CLEAR ERROR FLAG
;UPDATE PASS COUNT
;DISPLAY PASS COUNT
;CHECK FOR ACT-11 OR DDP
;IF NOT, CONTINUE TESTING

```

```

EOP: TYPE
MEPASS
CLR LAST
CLR ERRFLG
INC PASCNT
MOV PASCNT,LIGHTS
MOV 2#42,R1
BEQ RESTRT
RESET
LOGICAL: JSR PC,(R1)
NOP
NOP
NOP
RESTRT: JMP BEGIN

```

```

;CHECK FOR LOOP ON CURRENT TEST
;CHECK FOR ITERATION SUPPRESSION

```

```

SCOPER: BIT #SW10,SWR
BNE 4$
1$: BIT #SW14,SWR
BNE 3$
BIT #SW11,SWR
BNE 2$
INC LPCNT
CMP LPCNT,ICOUNT
BNE 3$
2$: CLR LPCNT
CLR ERRFLG
MOV (SP),RETURN
RTI
3$: MOV RETURN,(SP)
RTI
4$: TST ERRFLG
1$: BEQ 1$
2$: BR 2$

```

```

;CHECK FOR FREEZE ON CURRENT DATA

```

```

SCOP1R: BIT #SW09,SWR
BEQ 1$
MOV FREEZ1,(SP)
1$: RTI

```

```

;ERROR HANDLER

```

```

ERRORS: BIT #SW13,SWR
BNE HALTS

```

1893	010562	021667	001116		CMP	(SP),LAST		
1894	010566	001404			BEQ	1\$		
1895	010570	011667	001110		MOV	(SP),LAST		
1896	010574	005067	001040		CLR	ERRFLG		
1897	010600	104406		1\$:	SAVOSP			
1898	010602	011605			MOV	(SP),R5		
1899	010604	162705	000002		SUB	#2,R5		
1900	010610	011504			MOV	(R5),R4		
1901	010612	006304			ASL	R4		
1902	010614	006304			ASL	R4		
1903	010616	042704	177001		BIC	#177001,R4		
1904	010622	062704	013074		ADD	#ERRTAB,R4		
1905	010626	012467	000034		MOV	(R4)+,ERRMSG		
1906	010632	011467	000042		MOV	(R4),DATABP		
1907	010636	005767	000776		TST	ERRFLG		
1908	010642	001403			BEQ	TYPMSG		
1909	010644	005767	000030		TST	DATABP		
1910	010650	001007			BNE	TYPDAT		
1911	010652	104402			TYPMSG:	OCTASC		
1912	010654	010746				ERTABO		
1913	010656	012767	000001	000754	MOV	#1,ERRFLG		
1914	010664	104401			TYPE			
1915	010666	000000			ERRMSG:	0		
1916	010670	005767	000004		TYPDAT:	TST	DATABP	
1917	010674	001402				BEQ	RESREG	
1918	010676	104402				OCTASC		
1919	010700	000000			DATABP:	0		
1920	010702	104407			RESREG:	RESOS		
1921	010704	005767	166660		HALTS:	TST	SWR	
1922	010710	100005				BPL	EXITER	
1923	010712	010046				PUSHRO		
1924	010714	016600	000002		MOV	2(SP),P0		
1925	010720	000000			HALT			
1926	010722	012600			POPPO			
1927	010724	005267	000714		EXITER:	INC	ERRCNT	
1928	010730	032767	002000	166632		BIT	#SW10,SWR	
1929	010736	001402				BEQ	1\$	
1930	010740	016716	000704			MOV	ESCAPE,(SP)	
1931	010744	000002			1\$:	RTI		
1932	010746	000001			ERTABO:	1		
1933	010750	006	002			.BYTE	6,2	
1934	010752	011676			SAVPC			
1935						;TRAP DISPATCH SERVICE		
1936						;ARGUMENT OF TRAP IS EXTRACTED		
1937						;AND USED AS OFFSET TO OBTAIN POINTER		
1938						;TO SELECTED SUBROUTINE		
1939								
1940	010754	011646			TRPSRV:	MOV	(SP),-(SP)	;GET PC OF RETURN
1941	010756	162716	000002			SUB	#2,(SP)	;=PC OF TRAP
1942	010762	017616	000000			MOV	2(SP),(SP)	;GET TRP
1943	010766	006316			TRPOK:	ASL	(SP)	;MULTIPLY TRAF ARG BY 2
1944	010770	042716	177001			BIC	#177001,(SP)	;CLEAR UNWANTED BITS
1945	010774	062716	013014			ADD	#TRPTAB,(SP)	;POINTER TO SUBROUTINE ADDRESS
1946	011000	017616	000000			MOV	2(SP),(SP)	;SUBROUTINE ADDRESS
1947	011004	000136				JMP	2(SP)+	;GO TO SUBROUTINE
1948								


```

1949                                     ;TELETYPE OUTPUT ROUTINE
1950
1951 011006 017605 000000          TYPBR: MOV      2(SP),R5
1952 011012 062716 000002          ADD      #2,(SP)
1953 011016 105777 000560          1$:  TSTB   2TPCSR
1954 011022 100375                BPL      1$
1955 011024 105715                TSTB   (R5)
1956 011026 001001                BNE     2$
1957 011030 000002                RTI
1958 011032 112577 000546          2$:  MOVB   (R5)+,2TPDBR
1959 011036 000767                BR      1$
  
```

```

1960                                     ;ASCII STRING INPUT ROUTINE
1961
1962
1963 011040 017667 000000 000006  INSTRG: MOV      2(SP),MSG
1964 011046 062716 000002          ADD      #2,(SP)
1965 011052 104401                INSTR1: TYPE
1966 011054 000000                MSG:    0
1967 011056 012704 013036          MOV      #INBUF,R4
1968 011062 012703 000007          MOV      #7,R3
1969 011066 105777 000504          1$:  TSTB   2TKCSR
1970 011072 100375                BPL      1$
1971 011074 117714 000500          MOVB   2TKDBR,(R4)
1972 011100 142714 000200          BICB   #200,(R4)
1973 011104 122427 000015          CMPB   (R4)+,#15
1974 011110 001413                BEQ     INSTR2
1975 011112 117777 000462 000464  2$:  MOVB   2TKDBR,2TPDBR
1976 011120 105777 000456          TSTB   2TPCSR
1977 011124 100375                BPL      2$
1978 011126 005303                DEC     R3
1979 011130 001356                BNE     1$
1980 011132 104401                INSTRE: TYPE
1981 011134 012671                MQM
1982 011136 000745                BR      INSTR1
1983 011140 000002                INSTR2: RTI
  
```

```

1984                                     ;CONVERT ASCII STRING TO OCTAL
1985
1986
1987 011142 011605                PARAMS: MOV      (SP),R5
1988 011144 012567 000146          MOV      (R5)+,LOLIM
1989 011150 012567 000144          MOV      (R5)+,HILIM
1990 011154 012567 000142          MOV      (R5)+,DEVADR
1991 011160 112567 000140          MOV      (R5)+,LOBITS
1992 011164 112567 000135          MOV      (R5)+,ADRCNT
1993 011170 010516                MOV      R5,(SP)
1994 011172 005005                PARAM1: CLR     R5
1995 011174 012704 013036          MOV      #INBUF,R4
1996 011200 122714 000015          CMPB   #15,(R4)
1997 011204 001420                BEQ     PARERR
1998 011206 121427 000060          1$:  CMPB   (R4),#60
1999 011212 002415                BLT     PARERR
2000 011214 121427 000067          CMPB   (R4),#67
2001 011220 003012                BGT     PARERR
2002 011222 142714 000060          BICB   #60,(R4)
2003 011226 152405                BISB   (R4)+,R5
2004 011230 122714 000015          CMPB   #15,(R4)
  
```

2005	011234	001406		BEQ	LIMITS
2006	011236	006305		ASL	R5
2007	011240	006305		ASL	R5
2008	011242	006305		ASL	R5
2009	011244	000750		BR	1\$
2010	011246	104404		PARERR: INSTER	
2011	011250	000750		BR	PARAM1
2012					
2013					;TEST TO SEE IF NUMBER IS WITHIN LIMITS
2014					
2015	011252	020567	000042	LIMITS: CMP	R5, HILIM
2016	011256	101373		BHI	PARERR
2017	011260	020567	000032	CMP	R5, LOLIM
2018	011264	103770		BLO	PARERR
2019	011266	136705	000032	BITB	LOBITS, R5
2020	011272	001365		BNE	PARERR
2021					
2022					;STORE NUMBER AT SPECIFIED ADDRESS
2023					
2024	011274	016704	000022	1\$: MOV	DEVADR, R4
2025	011300	010524		MOV	R5, (R4)+
2026	011302	062705	000002	ADD	#2, R5
2027	011306	105367	000013	DECB	ADRCNT
2028	011312	001372		BNE	1\$
2029	011314	000002		RTI	
2030	011316	000000		LOLIM:	0
2031	011320	000000		HILIM:	0
2032	011322	000000		DEVADR:	0
2033	011324	000000		LOBITS:	0
2034		011325		ADRCNT=	LOBITS+1
2035					
2036					;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
2037					
2038	011326	104401		OCTASN: TYPE	
2039	011330	012675		MCRLF	
2040	011332	017601	000000	MOV	2(SP), R1
2041	011336	062716	000002	ADD	#2, (SP)
2042	011342	012167	000130	MOV	(R1)+, WRDCNT
2043	011346	112167	000126	1\$: MOV	(R1)+, CHRCNT
2044	011352	112167	000123	MOV	(R1)+, SPACNT
2045	011356	013167	000120	MOV	2(R1)+, BINWRD
2046	011362	016704	000114	2\$: MOV	BINWRD, R4
2047	011366	116705	000106	MOV	CHRCNT, R5
2048	011372	012700	013050	MOV	#TEMP, R0
2049	011376	010403		3\$: MOV	R4, R3
2050	011400	042703	177770	BIC	#177770, R3
2051	011404	062703	000260	ADD	#260, R3
2052	011410	110320		MOV	R3, (R0)+
2053	011412	006204		ASR	R4
2054	011414	006204		ASR	R4
2055	011416	006204		ASR	R4
2056	011420	005305		DEC	R5
2057	011422	001365		BNE	3\$
2058	011424	012703	013062	4\$: MOV	#MDATA, R3
2059	011430	114023		MOV	-(R0), (R3)+
2060	011432	105367	000042	DECB	CHRCNT

2061	011436	001374				BNE	4\$
2062	011440	105767	000035			TSTB	SPACNT
2063	011444	001405				BEQ	6\$
2064	011446	112723	000240		5\$:	MOVB	#240,(R3)+
2065	011452	105367	000023			DECB	SPACNT
2066	011456	001373				BNE	5\$
2067	011460	105013			6\$:	CLRB	(R3)
2068	011462	104401				TYPE	
2069	011464	013062				MDATA	
2070	011466	005367	000004			DEC	WRDCNT
2071	011472	001325				BNE	1\$
2072	011474	000002				RTI	
2073	011476	000000				WRDCNT:	0
2074	011500	000000				CHRCNT:	0
2075		011501				SPACNT=CHRCNT+1	
2076	011502	000000				BINWRD:	0
2077							
2078							;SAVE PC OF TEST THAT FAILED AND RO-R5
2079							
2080	011504	016667	000004	000164	SV05P:	MOV	4(SP),SAVPC
2081							
2082							;SAVE RO-R5
2083							
2084	011512	010567	000154		SV05:	MOV	R5,SAVR5
2085	011516	010467	000146			MOV	R4,SAVR4
2086	011522	010367	000140			MOV	R3,SAVR3
2087	011526	010267	000132			MOV	R2,SAVR2
2088	011532	010167	000124			MOV	R1,SAVR1
2089	011536	010067	000116			MOV	RO,SAVRC
2090	011542	000002				RTI	
2091							;RESTORE RO-R5
2092							
2093	011544	016700	000110		RS05:	MOV	SAVR0,RO
2094	011550	016701	000106			MOV	SAVR1,R1
2095	011554	016702	000104			MOV	SAVR2,R2
2096	011560	016703	000102			MOV	SAVR3,R3
2097	011564	016704	000100			MOV	SAVR4,R4
2098	011570	016705	000076			MOV	SAVR5,R5
2099	011574	000002				RTI	
2100							;INDIRECT POINTERS
2101							
2102	011576	177560			TKCSR:	177560	
2103	011600	177562			TKDBR:	177562	
2104	011602	177564			TPCSR:	177564	
2105	011604	177566			TPDBR:	177566	
2106	011606	000000			DHSCR:	0	
2107	011610	000000			DHNRC:	0	
2108	011612	000000			DHLPR:	0	
2109	011614	000000			DHBA:	0	
2110	011616	000000			DHBC:	0	
2111	011620	000000			DHBAR:	0	
2112	011622	000000			DHBCR:	0	
2113	011624	000000			DHSSR:	0	
2114	011626	000000			DHSLR:	0	
2115	011630	000000			DHRVEC:	0	
2116	011632	000000			DHRLVL:	0	

2173	011751	037	.BYTE	TDAT
2174	011752	040	.BYTE	TDAT
2175	011753	041	.BYTE	TDAT
2176	011754	042	.BYTE	TDAT
2177	011755	043	.BYTE	TDAT
2178	011756	044	.BYTE	TDAT
2179	011757	045	.BYTE	TDAT
2180	011760	046	.BYTE	TDAT
2181	011761	047	.BYTE	TDAT
2182	011762	050	.BYTE	TDAT
2183	011763	051	.BYTE	TDAT
2184	011764	052	.BYTE	TDAT
2185	011765	053	.BYTE	TDAT
2186	011766	054	.BYTE	TDAT
2187	011767	055	.BYTE	TDAT
2188	011770	056	.BYTE	TDAT
2189	011771	057	.BYTE	TDAT
2190	011772	060	.BYTE	TDAT
2191	011773	061	.BYTE	TDAT
2192	011774	062	.BYTE	TDAT
2193	011775	063	.BYTE	TDAT
2194	011776	064	.BYTE	TDAT
2195	011777	065	.BYTE	TDAT
2196	012000	066	.BYTE	TDAT
2197	012001	067	.BYTE	TDAT
2198	012002	070	.BYTE	TDAT
2199	012003	071	.BYTE	TDAT
2200	012004	072	.BYTE	TDAT
2201	012005	073	.BYTE	TDAT
2202	012006	074	.BYTE	TDAT
2203	012007	075	.BYTE	TDAT
2204	012010	076	.BYTE	TDAT
2205	012011	077	.BYTE	TDAT
2206	012012	100	.BYTE	TDAT
2207	012013	101	.BYTE	TDAT
2208	012014	102	.BYTE	TDAT
2209	012015	103	.BYTE	TDAT
2210	012016	104	.BYTE	TDAT
2211	012017	105	.BYTE	TDAT
2212	012020	106	.BYTE	TDAT
2213	012021	107	.BYTE	TDAT
2214	012022	110	.BYTE	TDAT
2215	012023	111	.BYTE	TDAT
2216	012024	112	.BYTE	TDAT
2217	012025	113	.BYTE	TDAT
2218	012026	114	.BYTE	TDAT
2219	012027	115	.BYTE	TDAT
2220	012030	116	.BYTE	TDAT
2221	012031	117	.BYTE	TDAT
2222	012032	120	.BYTE	TDAT
2223	012033	121	.BYTE	TDAT
2224	012034	122	.BYTE	TDAT
2225	012035	123	.BYTE	TDAT
2226	012036	124	.BYTE	TDAT
2227	012037	125	.BYTE	TDAT
2228	012040	126	.BYTE	TDAT

2229	012041	127	.BYTE	TDAT
2230	012042	130	.BYTE	TDAT
2231	012043	131	.BYTE	TDAT
2232	012044	132	.BYTE	TDAT
2233	012045	133	.BYTE	TDAT
2234	012046	134	.BYTE	TDAT
2235	012047	135	.BYTE	TDAT
2236	012050	136	.BYTE	TDAT
2237	012051	137	.BYTE	TDAT
2238	012052	140	.BYTE	TDAT
2239	012053	141	.BYTE	TDAT
2240	012054	142	.BYTE	TDAT
2241	012055	143	.BYTE	TDAT
2242	012056	144	.BYTE	TDAT
2243	012057	145	.BYTE	TDAT
2244	012060	146	.BYTE	TDAT
2245	012061	147	.BYTE	TDAT
2246	012062	150	.BYTE	TDAT
2247	012063	151	.BYTE	TDAT
2248	012064	152	.BYTE	TDAT
2249	012065	153	.BYTE	TDAT
2250	012066	154	.BYTE	TDAT
2251	012067	155	.BYTE	TDAT
2252	012070	156	.BYTE	TDAT
2253	012071	157	.BYTE	TDAT
2254	012072	160	.BYTE	TDAT
2255	012073	161	.BYTE	TDAT
2256	012074	162	.BYTE	TDAT
2257	012075	163	.BYTE	TDAT
2258	012076	164	.BYTE	TDAT
2259	012077	165	.BYTE	TDAT
2260	012100	166	.BYTE	TDAT
2261	012101	167	.BYTE	TDAT
2262	012102	170	.BYTE	TDAT
2263	012103	171	.BYTE	TDAT
2264	012104	172	.BYTE	TDAT
2265	012105	173	.BYTE	TDAT
2266	012106	174	.BYTE	TDAT
2267	012107	175	.BYTE	TDAT
2268	012110	176	.BYTE	TDAT
2269	012111	177	.BYTE	TDAT
2270	012112	200	.BYTE	TDAT
2271	012113	201	.BYTE	TDAT
2272	012114	202	.BYTE	TDAT
2273	012115	203	.BYTE	TDAT
2274	012116	204	.BYTE	TDAT
2275	012117	205	.BYTE	TDAT
2276	012120	206	.BYTE	TDAT
2277	012121	207	.BYTE	TDAT
2278	012122	210	.BYTE	TDAT
2279	012123	211	.BYTE	TDAT
2280	012124	212	.BYTE	TDAT
2281	012125	213	.BYTE	TDAT
2282	012126	214	.BYTE	TDAT
2283	012127	215	.BYTE	TDAT
2284	012130	216	.BYTE	TDAT

2285	012131	217	.BYTE	TDAT
2286	012132	220	.BYTE	TDAT
2287	012133	221	.BYTE	TDAT
2288	012134	222	.BYTE	TDAT
2289	012135	223	.BYTE	TDAT
2290	012136	224	.BYTE	TDAT
2291	012137	225	.BYTE	TDAT
2292	012140	226	.BYTE	TDAT
2293	012141	227	.BYTE	TDAT
2294	012142	230	.BYTE	TDAT
2295	012143	231	.BYTE	TDAT
2296	012144	232	.BYTE	TDAT
2297	012145	233	.BYTE	TDAT
2298	012146	234	.BYTE	TDAT
2299	012147	235	.BYTE	TDAT
2300	012150	236	.BYTE	TDAT
2301	012151	237	.BYTE	TDAT
2302	012152	240	.BYTE	TDAT
2303	012153	241	.BYTE	TDAT
2304	012154	242	.BYTE	TDAT
2305	012155	243	.BYTE	TDAT
2306	012156	244	.BYTE	TDAT
2307	012157	245	.BYTE	TDAT
2308	012160	246	.BYTE	TDAT
2309	012161	247	.BYTE	TDAT
2310	012162	250	.BYTE	TDAT
2311	012163	251	.BYTE	TDAT
2312	012164	252	.BYTE	TDAT
2313	012165	253	.BYTE	TDAT
2314	012166	254	.BYTE	TDAT
2315	012167	255	.BYTE	TDAT
2316	012170	256	.BYTE	TDAT
2317	012171	257	.BYTE	TDAT
2318	012172	260	.BYTE	TDAT
2319	012173	261	.BYTE	TDAT
2320	012174	262	.BYTE	TDAT
2321	012175	263	.BYTE	TDAT
2322	012176	264	.BYTE	TDAT
2323	012177	265	.BYTE	TDAT
2324	012200	266	.BYTE	TDAT
2325	012201	267	.BYTE	TDAT
2326	012202	270	.BYTE	TDAT
2327	012203	271	.BYTE	TDAT
2328	012204	272	.BYTE	TDAT
2329	012205	273	.BYTE	TDAT
2330	012206	274	.BYTE	TDAT
2331	012207	275	.BYTE	TDAT
2332	012210	276	.BYTE	TDAT
2333	012211	277	.BYTE	TDAT
2334	012212	300	.BYTE	TDAT
2335	012213	301	.BYTE	TDAT
2336	012214	302	.BYTE	TDAT
2337	012215	303	.BYTE	TDAT
2338	012216	304	.BYTE	TDAT
2339	012217	305	.BYTE	TDAT
2340	012220	306	.BYTE	TDAT

DZDHG MACY11 27(732) 28-SEP-76 15:16 PAGE 48
DZDHGB.P11

2341	012221	307	.BYTE	TDAT
2342	012222	310	.BYTE	TDAT
2343	012223	311	.BYTE	TDAT
2344	012224	312	.BYTE	TDAT
2345	012225	313	.BYTE	TDAT
2346	012226	314	.BYTE	TDAT
2347	012227	315	.BYTE	TDAT
2348	012230	316	.BYTE	TDAT
2349	012231	317	.BYTE	TDAT
2350	012232	320	.BYTE	TDAT
2351	012233	321	.BYTE	TDAT
2352	012234	322	.BYTE	TDAT
2353	012235	323	.BYTE	TDAT
2354	012236	324	.BYTE	TDAT
2355	012237	325	.BYTE	TDAT
2356	012240	326	.BYTE	TDAT
2357	012241	327	.BYTE	TDAT
2358	012242	330	.BYTE	TDAT
2359	012243	331	.BYTE	TDAT
2360	012244	332	.BYTE	TDAT
2361	012245	333	.BYTE	TDAT
2362	012246	334	.BYTE	TDAT
2363	012247	335	.BYTE	TDAT
2364	012250	336	.BYTE	TDAT
2365	012251	337	.BYTE	TDAT
2366	012252	340	.BYTE	TDAT
2367	012253	341	.BYTE	TDAT
2368	012254	342	.BYTE	TDAT
2369	012255	343	.BYTE	TDAT
2370	012256	344	.BYTE	TDAT
2371	012257	345	.BYTE	TDAT
2372	012260	346	.BYTE	TDAT
2373	012261	347	.BYTE	TDAT
2374	012262	350	.BYTE	TDAT
2375	012263	351	.BYTE	TDAT
2376	012264	352	.BYTE	TDAT
2377	012265	353	.BYTE	TDAT
2378	012266	354	.BYTE	TDAT
2379	012267	355	.BYTE	TDAT
2380	012270	356	.BYTE	TDAT
2381	012271	357	.BYTE	TDAT
2382	012272	360	.BYTE	TDAT
2383	012273	361	.BYTE	TDAT
2384	012274	362	.BYTE	TDAT
2385	012275	363	.BYTE	TDAT
2386	012276	364	.BYTE	TDAT
2387	012277	365	.BYTE	TDAT
2388	012300	366	.BYTE	TDAT
2389	012301	367	.BYTE	TDAT
2390	012302	370	.BYTE	TDAT
2391	012303	371	.BYTE	TDAT
2392	012304	372	.BYTE	TDAT
2393	012305	373	.BYTE	TDAT
2394	012306	374	.BYTE	TDAT
2395	012307	375	.BYTE	TDAT
2396	012310	376	.BYTE	TDAT


```

2397 012311 377 .BYTE TUAT
2398 .EVEN
2399 012312 000000 RBUF: 0
2400 012354 .=. +40
2401 ;ENTER HERE ON POWER FAILURE
2402
2403
2404 012354 010046 PFAIL: MOV R0, -(SP) ;SAVE R0-R5 ON PROCESSOR STACK
2405 012356 010146 MOV R1, -(SP)
2406 012360 010246 MOV R2, -(SP)
2407 012362 010346 MOV R3, -(SP)
2408 012364 010446 MOV R4, -(SP)
2409 012366 010546 MOV R5, -(SP)
2410 012370 016746 165430 MOV 24, -(SP)
2411 012374 010667 177274 MOV SP, SAVSP ;SAVE STACK POINTER
2412 012400 012767 012412 165416 MOV #RESTART, 24 ;SET UP FOR POWER UP TRAP
2413 012406 000000 HALT ;HALT ON POWER DOWN NORMAL
2414 012410 000777 BR .
2415
2416 ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
2417
2418 012412 016706 177256 RESTAR: MOV SAVSP, SP ;RESTORE STACK POINTER
2419 012416 012605 MOV (SP)+, R5 ;RESTORE R0-R5
2420 012420 012604 MOV (SP)+, R4
2421 012422 012603 MOV (SP)+, R3
2422 012424 012602 MOV (SP)+, R2
2423 012426 012601 MOV (SP)+, R1
2424 012430 012600 MOV (SP)+, R0
2425 012432 012767 012354 165364 MOV #PFAIL, 24 ;SET UP FOR POWER FAILURE
2426 012440 012767 000340 165330 MOV #340, PS
2427 012446 012706 013564 MOV #STACK, SP
2428 012452 005067 000372 CLR TEMP
2429 012456 005267 000366 INC TEMP
2430 012462 001375 BNE .-4
2431 012464 104402 OCTASC
2432 012466 012510 PFTAB
2433 012470 104401 TYPE
2434 012472 012700 MPFAIL
2435 012474 005067 177140 CLR ERRFLG
2436 012500 005067 177200 CLR LAST
2437 012504 000177 177136 JMP @RETURN
2438 012510 000001 PFTAB: 1
2439 012512 000006 000002 6, 2
2440 012516 000207 RETURN
2441 012520 005015 042012 030510 MTITLE: .ASCIZ <15><12><12>/DH11 SINGLE LINE PARITY CHECK & MULTI-LINE DATA TEST /<15><
2442 012526 020061 044523 043516
2443 012534 042514 046040 047111
2444 012542 020105 040520 044522
2445 012550 054524 041440 042510
2446 012556 045503 023040 046440
2447 012564 046125 044524 046055
2448 012572 047111 020105 040504
2449 012600 040524 052040 051505
2450 012606 020124 005015 000
2451 012613 015 053012 041505 MVECTO: .ASCIZ <15><12>/VECTOR ADDRESS-/
2452 012620 047524 020122 042101
  
```

2453	012626	051104	051505	026523
2454	012634	000		
2455	012635	015	041412	047117
2456	012642	051124	046117	051040
2457	012650	043505	051511	042524
2458	012656	020122	042101	051104
2459	012664	051505	026523	000
2460	012671	040	037440	000
2461	012675	015	000012	
2462	012700	020040	047520	042527
2463	012706	020122	040506	046111
2464	012714	051125	026105	050040
2465	012722	047522	051107	046501
2466	012730	051040	051505	040524
2467	012736	052122	040440	020124
2468	012744	042524	052123	044440
2469	012752	020116	051120	043517
2470	012760	042522	051523	000
2471	012765	015	042012	042132
2472	012772	043510	000	
2473	012775	015	051012	000
2474	013001	015	052012	051505
2475	013006	020124	041520	000055
2476				
2477				
2478				
2479				
2480	013014	010434		
2481	013016	011006		
2482	013020	011326		
2483	013022	011040		
2484	013024	011132		
2485	013026	011142		
2486	013030	011504		
2487	013032	011544		
2488	013034	010534		
2489				
2490				
2491				
2492	013036	000000		
2493		013050		
2494	013050	000000		
2495		013062		
2496	013062	000000		
2497		013074		
2498				
2499				
2500				
2501	013074			
2502	013074	013124		
2503	013076	000000		
2504	013100	013151		
2505	013102	000000		
2506	013104	013205		
2507	013106	000000		
2508	013110	013226		

MREGAD: .ASCIZ <15><12>/CONTROL REGISTER ADDRESS-/
 MQM: .ASCIZ / ?/
 MCRLF: .ASCIZ <15><12>
 MPFAIL: .ASCIZ / POWER FAILURE, PROGRAM RESTART AT TEST IN PROGRESS/
 MEPASS: .ASCIZ <15><12>/DZDHG/
 MR: .ASCIZ <15><12>/R/
 MTSTPC: .ASCIZ <15><12>/TEST PC-/
 .EVEN
 ;TABLE OF POINTERS FOR TRAP DECODING
 TRPTAB: SCOPER
 TYPER
 OCTASN
 INSTRG
 INSTRE
 PARAMS
 SVOSP
 RSOS
 SCOP1R
 ;BUFFERS FOR INPUT-OUTPUT
 INBUF: 0
 .=. +10
 TEMP: 0
 .=. +10
 MDATA: 0
 .=. +10
 ;TABLE OF POINTERS TO ERROR MESSAGES AND DATA
 ERRTAB:
 EMO
 0
 EM1
 0
 EM2
 0
 EM3

2509	013112	013346				DT1
2510	013114	013267				EM4
2511	013116	000000				0
2512	013120	013315				EMS
2513	013122	000000				0
2514	013124	047125	054105	042520	EMO:	.ASCIZ /UNEXPECTED INTERRUPT/
2515	013132	052103	042105	044440		
2516	013140	052116	051105	052522		
2517	013146	052120	000			
2518	013151	103	040510	040522	EM1:	.ASCIZ /CHARACTER AVAILABLE NOT SET/
2519	013156	052103	051105	040440		
2520	013164	040526	046111	041101		
2521	013172	042514	047040	052117		
2522	013200	051440	052105	000		
2523	013205	123	046111	020117	EM2:	.ASCIZ /SILO OVERRUN SET/
2524	013212	053117	051105	052522		
2525	013220	020116	042523	000124		
2526	013226	040504	040524	042440	EM3:	.ASCII /DATA ERROR/
2527	013234	051122	051117			
2528	013240	005015	054105	020120		.ASCIZ <15><12>/EXP REC LINE/
2529	013246	020040	020040	042522		
2530	013254	020103	020040	020040		
2531	013262	044514	042516	000		
2532	013267	124	040522	051516	EM4:	.ASCIZ /TRANSMIT DONE NOT SET/
2533	013274	044515	020124	047504		
2534	013302	042516	047040	052117		
2535	013310	051440	052105	000		
2536	013315	103	040510	040522	EMS:	.ASCIZ /CHARACTER AVAILABLE SET/
2537	013322	052103	051105	040440		
2538	013330	040526	046111	041101		
2539	013336	042514	051440	052105		
2540	013344	000				
2541		013346				.EVEN
2542						
2543						;DATA TABLES FOR ERRORS
2544						
2545	013346	000003			DT1:	3
2546	013350	006	002		.BYTE	6.2
2547	013352	011672				SAVR5
2548	013354	006	002		.BYTE	6.2
2549	013356	011662				SAVR1
2550	013360	002	000		.BYTE	2.0
2551	013362	011666				SAVR3
2552	013364	000000			ENDCOD:	0
2553		000001			.END	

M04

DZDHG MACY11 27(732) 28-SEP-76 15:16 PAGE 53
 DZDHGB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

ADRCNT=	011325	1992*	2027*	2034#														
BARBIT	010322	1817	1821#															
BEGIN	001202	395	424	430#	1858													
BINWRD	011502	2045*	2046	2076#														
BITX =	000000	449#	484#	519#	554#	589#	624#	659#	694#	729#	764#	799#	834#	869#				
		904#	939#	974#	1009#													
BIT00 =	000001	85#																
BIT01 =	000002	84#																
BIT02 =	000004	83#																
BIT03 =	000010	82#																
BIT04 =	000020	81#																
BIT05 =	000040	80#																
BIT06 =	000100	79#																
BIT07 =	000200	78#																
BIT08 =	000400	77#																
BIT09 =	001000	76#																
BIT10 =	002000	75#																
BIT11 =	004000	74#	463	498	533	568	603	638	673	708	743	778	813	848				
		883	918	953	988	1023	1073	1102	1152	1181	1231	1260	1310	1339				
		1389	1418	1468	1497	1547	1576	1626	1654	1704	1730							
BIT12 =	010000	73#																
BIT13 =	020000	72#																
BIT14 =	040000	71#																
BIT15 =	100000	70#																
CHRCNT	011500	2043*	2047	2060*	2074#	2075												
DATABP	010700	1906*	1909	1916	1919#													
DEVADR	011322	1990*	2024	2032#														
DHBA	011614	474*	509*	544*	579*	614*	649*	684*	719*	754*	789*	824*	859*	894*				
		929*	964*	999*	1029*	1108*	1187*	1266*	1345*	1424*	1503*	1582*	1660*	1736*				
		1815*	2109#															
DHBAR	011620	476*	511*	546*	581*	616*	651*	686*	721*	756*	791*	826*	861*	896*				
		931*	966*	1001*	1044*	1082	1123*	1161	1202*	1240	1281*	1319	1360*	1398				
		1439*	1477	1518*	1556	1597*	1635	1675*	1713	1757*	1808	1817*	2111#					
DHBC	011616	473*	508*	543*	578*	613*	648*	683*	718*	753*	788*	823*	858*	893*				
		928*	963*	998*	1030*	1109*	1188*	1267*	1346*	1425*	1504*	1583*	1661*	1737*				
		1816*	2110#															
DHBCR	011622	475*	510*	545*	580*	615*	650*	685*	720*	755*	790*	825*	860*	895*				
		930*	965*	1000*	2112#													
DHLPR	011612	471*	506*	541*	576*	611*	646*	681*	716*	751*	786*	821*	856*	891*				
		926*	961*	996*	1031*	1110*	1189*	1268*	1347*	1426*	1505*	1584*	1662*	1738*				
		2108#																
DHNRC	011610	479	514	549	584	619	654	689	724	759	794	829	864	899				
		934	969	1004	1059	1138	1217	1296	1375	1454	1533	1612	1690	1787				
		2107#																
DHRLVL	011632	1042*	1121*	1200*	1279*	1358*	1437*	1516*	1595*	1673*	1751*	2116#						
DHREVC	011630	410	1041*	1120*	1199*	1278*	1357*	1436*	1515*	1594*	1672*	1750*	2115#					
DHSCR	011606	418	463*	470*	477	498*	505*	512	533*	540*	547	568*	575*	582				
		603*	610*	617	638*	645*	652	673*	680*	687	708*	715*	722	743*				
		750*	757	778*	785*	792	813*	820*	827	848*	855*	862	883*	890*				
		897	918*	925*	932	953*	960*	967	988*	995*	1002	1023*	1035*	1043*				
		1054	1056	1073*	1102*	1114*	1122*	1133	1135	1152*	1181*	1193*	1201*	1212				
		1214	1231*	1260*	1272*	1280*	1291	1293	1310*	1339*	1351*	1359*	1370	1372				
		1389*	1418*	1430*	1438*	1449	1451	1468*	1497*	1509*	1517*	1528	1530	1547*				
		1576*	1588*	1596*	1607	1609	1626*	1654*	1666*	1674*	1685	1687	1704*	1730*				
		1742*	1756*	1764	1771	1774	1813*	1814*	2106#									
DHSLR	011626	421*	422*	1084	1163	1242	1321	1400	1479	1558	1637	1715	1810	2114#				

		779*	814*	849*	884*	919*	954*	989*	1028*	1034*	1038*	1063*	1107*	1113*
		1117*	1142*	1186*	1192*	1196*	1221*	1265*	1271*	1275*	1300*	1344*	1350*	1354*
		1379*	1423*	1429*	1433*	1458*	1502*	1508*	1512*	1537*	1581*	1587*	1591*	1616*
		1659*	1665*	1669*	1694*	1735*	1741*	1745*	1793*	1803	1814	1968*	1978*	2049*
R4	=%000004	2050*	2051*	2052	2058*	2059*	2064*	2067*	2086	2096*	2407	2421*	1973	1995*
		47#	1900*	1901*	1902*	1903*	1904*	1905	1906	1967*	1971*	1972*	2054*	2055*
		1996	1998	2000	2002*	2003	2004	2024*	2025*	2046*	2049	2053*		
R5	=%000005	2085	2097*	2408	2420*									
		48#	465*	480	500*	515	535*	550	570*	585	605*	620	640*	655
		675*	690	710*	725	745*	760	780*	795	815*	830	850*	865	885*
		900	920*	935	955*	970	990*	1005	1067*	1146*	1225*	1304*	1383*	1462*
		1541*	1620*	1698*	1797*	1898*	1899*	1900	1951*	1955	1958	1987*	1988	1989
		1990	1991	1992	1993	1994*	2003*	2006*	2007*	2008*	2015	2017	2019	2025
		2026*	2047*	2056*	2084	2098*	2409	2419*						
SAVPC	011676	1934	2080*	2136#										
SAVRO	011660	2089*	2093	2129#										
SAVR1	011662	2088*	2094	2130#	2549									
SAVR2	011664	2087*	2095	2131#										
SAVR3	011666	2086*	2096	2132#	2551									
SAVR4	011670	2085*	2097	2133#										
SAVRS	011672	2084*	2098	2134#	2547									
SAVSP	011674	2135#	2411*	2418										
SAVOSP=	104406	369#	1897											
SCOPE =	104400	363#	483	518	553	588	623	658	693	728	763	798	833	868
		903	938	973	1008	1087	1166	1245	1324	1403	1482	1561	1640	1717
		1819												
SCOPER	010434	1863#	2480											
SCOPE1=	104410	371#												
SCOPIR	010534	1884#	2488											
SP	=%000006	49#	382*	431*	1874	1876*	1886*	1893	1895	1898	1924	1930*	1940*	1941*
		1942*	1943*	1944*	1945*	1946*	1947	1951	1952*	1963	1964*	1987	1993*	2040
		2041*	2080	2404*	2405*	2406*	2407*	2408*	2409*	2410*	2411	2418*	2419	2420
		2421	2422	2423	2424	2427*								
		2044*	2062	2065*	2075#									
SPACNT=	011501	57#	382	431	2427									
STACK =	013564													
START	001000	355	381#											
STFLG	011702	384*	444	446*	2138#									
SV05	011512	2084#												
SV05P	011504	2080#	2486											
SWR =	177570	54#	394	432	1863	1865	1867	1884	1891	1921	1928			
SW00 =	000001	37#	394											
SW01 =	000002	36#	432											
SW02 =	000004	35#												
SW03 =	000010	34#												
SW04 =	000020	33#												
SW05 =	000040	32#												
SW06 =	000100	31#												
SW08 =	000400	30#												
SW09 =	001000	29#	1884											
SW10 =	002000	28#	1863	1928										
SW11 =	004000	27#	1867											
SW12 =	010000	26#												
SW13 =	020000	25#	1891											
SW14 =	040000	24#	1865											
SW15 =	100000	23#												
TBUF	011712	1029	1108	1187	1266	1345	1424	1503	1582	1660	1736	1815	2142#	

T32	007672	1727#												
T4	001742	564#												
T5	002104	599#												
T6	002246	634#												
T7	002410	669#												
VEC1	001060	391	394#											
VEC2	001070	393	396#											
WRDCNT	011476	2042*	2070*	2073#										
X	= 000000	1#												
XBIT	= 000001	449#												
XLINE	= 000000	449#												
XN	= 000033	1#	459	463#	494	498#	529	533#	564	568#	599	603#	634	638#
		669	673#	704	708#	739	743#	774	778#	809	813#	844	848#	879
		883#	914	918#	949	953#	984	988#	1020	1023#	1099	1102#	1178	1181#
		1257	1260#	1336	1339#	1415	1418#	1494	1497#	1573	1576#	1651	1654#	1727
		1730#												
Y	= 000011	1#	363	364#	365#	366#	367#	368#	369#	370#	371#	372#		
.	= 013366	87#	88	90	92	94	96	98	100	102	104	106	108	110
		112	114	116	118	120	122	124	126	128	130	132	134	136
		138	140	142	144	146	148	150	152	154	156	158	160	162
		164	166	168	170	172	174	176	178	180	182	184	186	188
		190	192	194	196	198	200	202	204	206	208	210	212	214
		216	218	220	222	224	226	228	230	232	234	236	238	240
		242	244	246	248	250	252	254	256	258	260	262	264	266
		268	270	272	274	276	278	280	282	284	286	288	290	292
		294	296	298	300	302	304	306	308	310	312	314	316	318
		320	322	324	326	328	330	332	334	336	338	340	342	347#
		354#	372#	481	516	551	586	621	656	691	726	761	796	831
		866	901	936	971	1006	1057	1066	1083	1085	1136	1145	1162	1164
		1215	1224	1241	1243	1294	1303	1320	1322	1373	1382	1399	1401	1452
		1461	1478	1480	1531	1540	1557	1559	1610	1619	1636	1638	1688	1697
		1714	1716	1765	1772	1775	1796	2400#	2414	2430	2493#	2495#	2497#	2541#

G05

DZDHG MACY11 27(732) 29-SEP-76 15:16 PAGE 62
 DZDHGB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

ADD	401	402	1037	1038	1116	1117	1195	1196	1274	1275	1353	1354	1432	1433	1511
ASL	1512	1590	1591	1668	1669	1744	1745	1904	1945	1952	1964	2026	2041	2051	
ASR	1064	1143	1222	1301	1380	1459	1538	1617	1695	1794	1901	1902	1943	2006	2007
BEQ	2008	2054	2055												
BGT	395	433	481	516	551	586	621	656	691	726	761	796	831	866	901
BHI	936	971	1006	1057	1066	1136	1145	1215	1224	1294	1303	1373	1382	1452	1461
BIC	1531	1540	1610	1619	1688	1697	1775	1796	1852	1879	1885	1894	1908	1917	1929
BICB	1974	1997	2005	2063											
BIS	2001														
BISB	2016	1141	1220	1299	1378	1457	1536	1615	1693	1792	1813	1903	1944	2050	
BIT	1062	2002													
BITB	1972	1817													
BLO	1814														
BLT	2003														
BMI	394	432	1056	1135	1214	1293	1372	1451	1530	1609	1687	1774	1863	1865	1867
BNE	1884	1891	1928												
BPL	2019														
BR	2018														
CLR	1999														
CLRB	1765														
CMP	391	404	424	445	1040	1083	1085	1119	1162	1164	1198	1241	1243	1277	1320
CMPB	1322	1356	1399	1401	1435	1478	1480	1514	1557	1559	1593	1636	1638	1671	1714
COM	1716	1747	1800	1802	1804	1806	1809	1811	1864	1866	1868	1871	1892	1910	1956
DEC	1979	2020	2028	2057	2061	2066	2071	2430							
DECB	478	513	548	583	618	653	688	723	758	793	828	863	898	933	968
EMT	1003	1055	1134	1213	1292	1371	1450	1529	1608	1686	1772	1789	1922	1954	1970
HALT	1977														
INC	393	442	1074	1153	1232	1311	1390	1469	1548	1627	1705	1759	1777	1812	1818
INCB	1680	1959	1982	2009	2011	2414									
JMP	384	385	386	387	388	400	1025	1045	1104	1124	1183	1203	1262	1282	1341
JSR	1361	1420	1440	1499	1519	1578	1598	1656	1676	1732	1753	1758	1847	1848	1872
MOV	1873	1896	1994	2428	2435	2436									
	1033	1112	1191	1270	1349	1428	1507	1586	1664	1740	2067				
	403	480	515	550	585	620	655	690	725	760	795	830	865	900	935
	970	1005	1065	1144	1223	1302	1381	1460	1539	1618	1696	1870	1893	2015	2017
	1795	1973	1996	1998	2000	2004									
	425	446													
	1039	1118	1197	1276	1355	1434	1513	1592	1670	1746	1805	1978	2056	2070	
	2027	2060	2065												
	67														
	89	91	93	95	97	99	101	103	105	107	109	111	113	115	117
	119	121	123	125	127	129	131	133	135	137	139	141	143	145	147
	149	151	153	155	157	159	161	163	165	167	169	171	173	175	177
	179	181	183	185	187	189	191	193	195	197	199	201	203	205	207
	209	211	213	215	217	219	221	223	225	227	229	231	233	235	237
	239	241	243	245	247	249	251	253	255	257	259	261	263	265	267
	269	271	273	275	277	279	281	283	285	287	289	291	293	295	297
	299	301	303	305	307	309	311	313	315	317	319	321	323	325	327
	329	331	333	335	337	339	341	343	1925	2413					
	422	1035	1036	1114	1115	1193	1194	1272	1273	1351	1352	1430	1431	1509	1510
	1588	1589	1666	1667	1742	1743	1849	1869	1927	2429					
	1069	1148	1227	1306	1385	1464	1543	1622	1700	1799					
	355	448	1046	1125	1204	1283	1362	1441	1520	1599	1677	1820	1858	1947	2437
	1854														
	381	382	383	396	397	398	399	421	430	431	443	459	460	461	462

	463	464	465	469	470	471	473	474	475	476	479	494	495	496	497
	498	499	500	504	505	506	508	509	510	511	514	529	530	531	532
	533	534	535	539	540	541	543	544	545	546	549	564	565	566	567
	568	569	570	574	575	576	578	579	580	581	584	599	600	601	602
	603	604	605	609	610	611	613	614	615	616	619	634	635	636	637
	638	639	640	644	645	646	648	649	650	651	654	659	670	671	672
	673	674	675	679	680	681	683	684	685	686	689	704	705	706	707
	708	709	710	714	715	716	718	719	720	721	724	739	740	741	742
	743	744	745	749	750	751	753	754	755	756	759	774	775	776	777
	778	779	780	784	785	786	788	789	790	791	794	809	810	811	812
	813	814	815	819	820	821	823	824	825	826	829	844	845	846	847
	848	849	850	854	855	856	858	859	860	861	864	879	880	881	882
	883	884	885	889	890	891	893	894	895	896	899	914	915	916	917
	918	919	920	924	925	926	928	929	930	931	934	949	950	951	952
	953	954	955	959	960	961	963	964	965	966	969	984	985	986	987
	988	989	990	994	995	996	998	999	1000	1001	1004	1020	1021	1022	1023
	1024	1027	1028	1029	1030	1031	1041	1042	1043	1044	1059	1060	1063	1067	1073
	1086	1099	1100	1101	1102	1103	1106	1107	1108	1109	1110	1120	1121	1122	1123
	1138	1139	1142	1146	1152	1165	1178	1179	1180	1181	1182	1185	1186	1187	1188
	1189	1199	1200	1201	1202	1217	1218	1221	1225	1231	1244	1257	1258	1259	1260
	1261	1264	1265	1266	1267	1268	1278	1279	1280	1281	1296	1297	1300	1304	1310
	1323	1336	1337	1338	1339	1340	1343	1344	1345	1346	1347	1357	1358	1359	1360
	1375	1376	1379	1383	1389	1402	1415	1416	1417	1418	1419	1422	1423	1424	1425
	1426	1436	1437	1438	1439	1454	1455	1458	1462	1468	1481	1494	1495	1496	1497
	1498	1501	1502	1503	1504	1505	1515	1516	1517	1518	1533	1534	1537	1541	1547
	1560	1573	1574	1575	1576	1577	1580	1581	1582	1583	1584	1594	1595	1596	1597
	1612	1613	1616	1620	1626	1639	1651	1652	1653	1654	1655	1658	1659	1660	1661
	1662	1672	1673	1674	1675	1690	1691	1694	1698	1704	1727	1728	1729	1730	1731
	1734	1735	1736	1737	1738	1748	1749	1750	1751	1752	1755	1756	1757	1787	1790
	1793	1797	1807	1815	1816	1850	1851	1874	1876	1886	1895	1898	1900	1905	1906
	1913	1924	1930	1940	1942	1946	1951	1963	1967	1968	1987	1988	1989	1990	1993
	1995	2024	2025	2040	2042	2045	2046	2048	2049	2058	2080	2084	2085	2086	2087
	2088	2089	2093	2094	2095	2096	2097	2098	2404	2405	2406	2407	2408	2409	2410
	2411	2412	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427			
MOV B	1034	1113	1192	1271	1350	1429	1508	1587	1665	1741	1958	1971	1975	1991	1992
	2043	2044	2047	2052	2059	2064									
NOP	1855	1856	1857												
RESET	1853														
RETURN	439	2440													
RTI	1070	1149	1228	1307	1386	1465	1544	1623	1701	1767	1780	1875	1877	1887	1931
	1957	1983	2029	2072	2090	2099									
SUB	1899	1941													
SWAB	1061	1140	1219	1298	1377	1456	1535	1614	1692	1791					
TRAP	363	364	365	366	367	368	369	370	371						
TST	390	423	444	1082	1161	1240	1319	1398	1477	1556	1635	1713	1764	1788	1801
	1803	1808	1878	1907	1909	1916	1921								
TSTB	477	512	547	582	617	652	687	722	757	792	827	862	897	932	967
	1002	1054	1084	1133	1163	1212	1242	1291	1321	1370	1400	1449	1479	1528	1558
	1607	1637	1685	1715	1771	1810	1953	1955	1969	1976	2062				
.ASCII	2526														
.ASCIZ	2441	2451	2455	2460	2461	2462	2471	2473	2474	2514	2518	2523	2528	2532	2536
.BYTE	411	412	419	420	440	441	1933	2142	2143	2144	2145	2146	2147	2148	2149
	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159	2160	2161	2162	2163	2164
	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179
	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191	2192	2193	2194
	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207	2208	2209

	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223	2224
	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239
	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254
	2255	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269
	2270	2271	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284
	2285	2286	2287	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299
	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314
	2315	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329
	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344
	2345	2346	2347	2348	2349	2350	2351	2352	2353	2354	2355	2356	2357	2358	2359
	2360	2361	2362	2363	2364	2365	2366	2367	2368	2369	2370	2371	2372	2373	2374
	2375	2376	2377	2378	2379	2380	2381	2382	2383	2384	2385	2386	2387	2388	2389
	2390	2391	2392	2393	2394	2395	2396	2397	2546	2548	2550				
.ENABL	20														
.END	2553														
.ENDC	393	394	421	423	463	498	533	568	603	638	673	708	743	778	813
	848	883	918	953	988	1023	1031	1102	1110	1181	1188	1189	1260	1267	1268
	1339	1346	1347	1418	1425	1426	1497	1504	1505	1576	1583	1584	1654	1730	
.EQUIV	67														
.EVEN	2398	2476	2541												
.IF	391	393	421	462	497	532	567	602	637	672	707	742	777	812	847
	882	917	952	987	1023	1030	1031	1102	1109	1110	1181	1188	1189	1260	1267
	1268	1339	1346	1347	1418	1425	1426	1497	1504	1576	1583	1654	1730		
.IFF	393	394													
.IIF	380														
.IRP	2106	2140													
.LIST	1	20	364	365	366	367	368	369	370	371	372	449	463	484	498
	519	533	554	568	589	603	624	638	659	673	694	708	729	743	764
	778	799	813	834	848	869	883	904	918	939	953	974	988	1009	1023
	1102	1181	1260	1339	1418	1497	1576	1654	1730	2142	2144	2145	2146	2147	2148
	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159	2160	2161	2162	2163
	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178
	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191	2192	2193
	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207	2208
	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223
	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238
	2239	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253
	2254	2255	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268
	2269	2270	2271	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283
	2284	2285	2286	2287	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298
	2299	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313
	2314	2315	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327	2328
	2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343
	2344	2345	2346	2347	2348	2349	2350	2351	2352	2353	2354	2355	2356	2357	2358
	2359	2360	2361	2362	2363	2364	2365	2366	2367	2368	2369	2370	2371	2372	2373
	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383	2384	2385	2386	2387	2388
	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398					
.MACRO	1	372	449												
.NLIST	1	20	364	365	366	367	368	369	370	371	372	449	463	484	498
	519	533	554	568	589	603	624	638	659	673	694	708	729	743	764
	778	799	813	834	848	869	883	904	918	939	953	974	988	1009	1023
	1102	1181	1260	1339	1418	1497	1576	1654	1730	2142	2144	2145	2146	2147	2148
	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159	2160	2161	2162	2163
	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178
	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191	2192	2193
	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207	2208

2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223
2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238
2239	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253
2254	2255	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268
2269	2270	2271	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283
2284	2285	2286	2287	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298
2299	2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313
2314	2315	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327	2328
2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343
2344	2345	2346	2347	2348	2349	2350	2351	2352	2353	2354	2355	2356	2357	2358
2359	2360	2361	2362	2363	2364	2365	2366	2367	2368	2369	2370	2371	2372	2373
2374	2375	2376	2377	2378	2379	2380	2381	2382	2383	2384	2385	2386	2387	2388
2389	2390	2391	2392	2393	2394	2395	2396	2397						
.PAGE	39	86	344	372										
.REPT	88	449	2143											
.TITLE	20													

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

*.DZDHGB.SEQ+UTIL1.DZDHGB
 RUN-TIME: 13 22 4 SECONDS
 RUN-TIME RATIO: 166/40=4.1
 CORE USED: 11K (21 PAGES)

