

RK611

DISKLESS CONT NO. 5
MD-11-DZR6E-A

EP-DZR6E-A-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

DEC 1976
digital
MADE IN USA

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500
501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600
601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700
701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800
801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900
901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000

RAK11 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 1
DZR6EA.P11 29-SEP-76 15:31

RKE11 DISKLESS CONTROLLER DIA
DZ06EA.P11 29-SEP-76 15:31

IDENTIFICATION

PRODUCT NAME: RK611 DISKLESS CONTROLLER DIAGNOSTIC: PART 5

DATE: DECEMBER, 1976

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: MARY TEGROTENHUIS

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A
LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE
TERMS OF SUCH LICENSE.

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

COPYRIGHT (C) 1976 BY DIGITAL EQUIPMENT CORPORATION

[illegible]

AK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 2
DZR6EA.F11 28-SEP-76 15:31

RK611 DISKLESS CONTROLLER DIA
DIR6EA.F11 28-SEP-76 15:31

အမည်အကျဉ်းချုပ်

2.0 REQUIREMENTS

2.1 EQUIPMENT

2.2 PRELIMINARY PROGRAMS

3.1 LOADING PROCEDURE

3.1 LOADING PROCEDURE

3.2 STARTING PROCEDURE

3.3 OPTIONAL SWITCH SETTING

3.4 RUN TIME

4.1 'SOFTWARE' SWITCH REGISTER

4.2 CONTROL C (IC) OPERATION

4.3 CONTROL S (↑S) OPERATION

4.4 CONTROL Q (IQ) OPERATION

6.0 ERROR REPORTING

001

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 3
DZR6EA.P11 28-SEP-76 15:31

SEQ 0003

1.0 ABSTRACT

THE RK611 DISKLESS CONTROLLER DIAGNOSTIC: PART 3

- A. TESTS MULTI-SECTOR DATA TRANSFERS.
- B. TESTS MID-TRANSFER SEEKS.
- C. TESTS CYLINDER OVERFLOW CHECKING.
- D. TESTS NPR TRANSFERS TO MEMORY.
- E. TESTS ECC ERROR DETECTION AND CORRECTION, BOTH 16 AND 18 BIT MODE.
- F. TESTS WRITE CHECK BOTH 16 AND 18 BIT MODES AND FORCES WRITE CHECK ERRORS.

NO RK06 DRIVE IS REQUIRED FOR PROGRAM EXECUTION.

2.0 REQUIREMENTS

2.1 EQUIPMENT

PDP-11 SYSTEM (16K CORE MEMORY)
CONSOLE TERMINAL
DECTAPE, PAPER TAPE READER, OR DECDISK
RK611 CONTROLLER

2.2 PRELIMINARY PROGRAMS

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PART 1 (DZR6A)
RK611 DISKLESS CONTROLLER DIAGNOSTIC: PART 2 (DZR6B)
RK611 DISKLESS CONTROLLER DIAGNOSTIC: PART 3 (DZR6C)
RK611 DISKLESS CONTROLLER DIAGNOSTIC: PART 4 (DZR6D)

3.0 OPERATING PROCEDURES

3.1 LOADING PROCEDURE

THE PROGRAM CAN BE LOADED FROM PAPER TAPE USING ABSOLUTE LOADER
OR FROM XXDP MEDIA SUPPORTED BY XXDP.

3.2 STARTING PROCEDURE

LOCATION 200 - START PROGRAM
LOCATION 204 - RESTART PROGRAM
LOCATION 214 - REQUEST BUS ADDRESS, VECTOR ADDRESS, AND
PRIORITY MODIFICATION

55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102

E01

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 4
DZR6EA.P11 28-SEP-76 15:31

SEQ 0004

3.3 OPTIONAL SWITCH SETTINGS

SW15 - HALT PROGRAM
SW14 - LOOP ON TEST
SW13 - INHIBIT ERROR TYPE OUT
SW12 - ABORT AFTER 20 ERRORS
SW11 - INHIBIT ITERATION COUNT
SW10 - BELL ON ERROR
SW9 - LOOP ON ERROR
SW8 - LOOP ON TEST IN SWITCHES 0-7

3.5 RUN TIME

FIRST PASS 1:40 MINUTES
SUBSEQUENT PASSES 3:40 MINUTES

4.0 OPERATING PROCEDURES

THE PROGRAM IS EXECUTED BY STARTING AT THE APPROPRIATE ADDRESS.

4.1 'SOFTWARE' SWITCH REGISTER

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR (I.E., AN 11/04 OR 11/34) THE PROGRAM WILL DETERMINE THAT THE HARDWARE SWITCH REGISTER IS NOT PRESENT AND WILL USE A 'SOFTWARE' SWITCH REGISTER. THE SETTINGS OF THE 'SOFTWARE' SWITCHES ARE CONTROLLED THROUGH A KEYBOARD ROUTINE WHICH IS CALLED BY TYPING 'CONTROL G'. THE PROGRAM WILL RECOGNIZE THE 'CONTROL G' AT ANY TIME EXCEPT WHEN THE PROGRAM IS AT A HIGHER PRIORITY PROCESSING AN RK06 INTERRUPT. THE 'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

SWR = NNNNNN NEW =

EACH TIME SWITCH SETTINGS ARE ENTERED, THE ENTIRE SWITCH REGISTER IMAGE MUST BE ENTERED. LEADING ZEROES ARE NOT REQUIRED. 'FUBOUT' AND 'CONTROL U' FUNCTIONS MAY BE USED TO CORRECT TYPING ERRORS DURING SWITCH ENTRY.

ON PROCESSORS WITH HARDWARE SWITCH REGISTERS, THE 'SOFTWARE' SWITCH REGISTER MAY BE USED. IF THE PROGRAM FINDS ALL 16 SWITCHES IN THE 'UP' POSITION, ALL SWITCH REGISTER REFERENCES WILL BE TO THE 'SOFTWARE' REGISTER AND THE PROCEDURES DESCRIBED ABOVE MUST BE FOLLOWED.

4.2 CONTROL C (IC) OPERATION

IF IC IS TYPED AT ANY TIME DURING THE PROGRAM EXECUTION THE PROGRAM IS HALTED IMMEDIATELY. IF A MONITOR IS PRESENT (XXDP CHAIN, ACT, APT) THE PROGRAM RETURNS CONTROL TO THE MONITOR. IF NO MONITOR IS PRESENT, THE CPU IS HALTED. DEPRESSING THE CONTINUE KEY WILL DO A PROGRAM RESTART.

F01

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 5
DZR6EA.P11 28-SEP-76 15:31

SEQ 0005

159
160
161
162
163
164
165
166
167

4.3 CONTROL S (↑S) OPERATION

IF ↑S IS TYPED AT ANY TIME THE PROGRAM WILL GO INTO A STALL LOOP
UNTIL A CONTROL Q (↑Q) IS TYPED.

4.4 CONTROL G (↑Q) OPERATION

IF A ↑S HAS BEEN TYPED, TYPING THE ↑Q CANCELS THE STALL
INITIATED BY THE ↑S.

GO1

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 6
DZR6EA.P11 28-SEP-76 15:31

SEQ 0006

5.0 PROGRAM DESCRIPTION

**MULTI-SECTOR WRITE OPERATIONS

TEST 1 MULTI-SECTOR WRITE (PART 1)

CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT THE CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
DATA OF 401 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE
AND A GOOD HEADER. CLOCK THROUGH 400 WORDS AND THE
TWO ECC WORDS. MAKE SURE THE NEXT HEADER RECOGNITION
CYCLE STARTS AND THE PROPER HEADER IS SEARCHED FOR.
PROVIDE A SECOND GOOD HEADER AND CHECK RECOGNITION
OF HEADER.

TEST 2 MULTI-SECTOR WRITE (PART 2)

CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT THE CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
DATA OF 401 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
CYLINDER 0, HEAD 0, SECTOR 23. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE
AND A GOOD HEADER. CLOCK THROUGH 400 WORDS AND THE
TWO ECC WORDS. MAKE SURE THE NEXT HEADER RECOGNITION
CYCLE STARTS AND THE PROPER HEADER IS SEARCHED FOR.
PROVIDE A SECOND GOOD HEADER AND CHECK RECOGNITION
OF HEADER.

TEST 3 MID-TRANSFER SEEK (PART 1)

CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT THE CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
DATA OF 401 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
CYLINDER 0, HEAD 0, SECTOR 25. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE
AND A GOOD HEADER. CLOCK THROUGH 400 WORDS AND THE
TWO WORD ECC. MAKE SURE THE CONTROLLER ISSUES
AN IMPLIED SEEK TO THE RIGHT HEAD AND CYLINDER.
MAKE SURE THE NEXT HEADER RECOGNITION PROCEEDS PROPERLY.

TEST 4 MID-TRANSFER SEEK (PART 2)

CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT THE CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
DATA OF 401 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
CYLINDER 0, HEAD 2, SECTOR 25. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE
AND A GOOD HEADER. CLOCK THROUGH 400 WORDS AND THE
TWO WORD ECC. MAKE SURE THE CONTROLLER ISSUES

168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221

H01

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 7
DZR6EA.P11 28-SEP-76 15:31

SEQ 0007

AN IMPLIED SEEK TO THE RIGHT HEAD AND CYLINDER.
MAKE SURE THE NEXT HEADER RECOGNITION PROCEEDS PROPERLY.

TEST 5 MID-TRANSFER SEEK (PART 3)

CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT THE CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
DATA OF 401 WORDS IN 26 SECTOR FORMAT WITH CDT SET,
CYLINDER 0, HEAD 2, SECTOR 25. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE
AND A GOOD HEADER. CLOCK THROUGH 400 WORDS AND THE
TWO WORD ECC. MAKE SURE THE CONTROLLER ISSUES
AN IMPLIED SEEK TO THE RIGHT HEAD AND CYLINDER.
MAKE SURE THE NEXT HEADER RECOGNITION PROCEEDS PROPERLY.

TEST 6 MID-TRANSFER SEEK (PART 4)

CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT THE CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
DATA OF 401 WORDS IN 26 SECTOR FORMAT WITH CDT SET,
CYLINDER 0, HEAD 4, SECTOR 25. CLOCK THROUGH SEEK AND
DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE
AND A GOOD HEADER. CLOCK THROUGH 400 WORDS AND THE
TWO WORD ECC. MAKE SURE THE CONTROLLER ISSUES
IMPLIED SEEK TO THE RIGHT HEAD AND CYLINDER.
MAKE SURE THE NEXT HEADER RECOGNITION PROCEEDS PROPERLY.

TEST 7 MID-TRANSFER SEEK AND 24 SECTOR FORMAT

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE DATA
DATA OF 401 WORDS TO AN RK06 IN 24 SECTOR FORMAT,
CYLINDER 0, HEAD 0, SECTOR 23. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE
AND A GOOD HEADER. CLOCK THROUGH 400 18 BIT WORDS
AND THE 32 BIT ECC. CHECK THAT IMPLIED SEEK IS
ISSUE TO NEXT TRACK. CHECK FOR PROPER HEADER RECOGNITION
OCCURS.

TEST 10 CYLINDER OVERFLOW (PART 1)

CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT THE CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
DATA OF 400 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
CYLINDER 632, HEAD 2, SECTOR 25. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE
AND A GOOD HEADER. CLOCK THROUGH 400 WORDS AND THE
TWO WORD ECC. MAKE SURE CYLINDER OVERFLOW ERROR DOES
NOT SET.

222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275

101

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 8
DZR6EA.P11 28-SEP-76 15:31

SEG 0008

276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323

TEST 11 CYLINDER OVERFLOW (PART 2)

CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT THE CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
DATA OF 401 WORDS TO AN RK06 IN 26 SECTOR FORMAT
CYLINDER 632, HEAD 2, SECTOR 25. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE
AND A GOOD HEADER. CLOCK THROUGH 400 WORDS AND
TWO ECC WORDS. MAKE SURE CYLINDER OVERFLOW ERROR SETS.

TEST 12 CYLINDER OVERFLOW (PART 3)

CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT THE CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
DATA OF 401 WORDS IN 26 SECTOR FORMAT WITH CDT SET
CYLINDER 632, HEAD 4, SECTOR 25. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE
AND A GOOD HEADER. CLOCK THROUGH 400 WORDS AND
TWO ECC WORDS. MAKE SURE CYLINDER OVERFLOW.
ERROR DOES NOT SET.

TEST 13 CYLINDER OVERFLOW (PART 4)

CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT THE CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
DATA OF 401 WORDS IN 26 SECTOR FORMAT WITH CDT SET
CYLINDER 1456, HEAD 4, SECTOR 25. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE
AND A GOOD HEADER. CLOCK THROUGH 400 WORDS AND
TWO ECC WORDS. MAKE SURE CYLINDER OVERFLOW ERROR SETS.

**NPR WRITING OF MEMORY

TEST 14 NPR WRITING OF MEMORY

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ DATA
OF 400 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE,
A GOOD HEADER, 40 DATA WORDS. VERIFY THAT THE WORDS
ARE WRITTEN PROPERLY IN MEMORY.

J01

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 9
DZR6EA.P11 28-SEP-76 15:31

SEQ 0009

**ECC ERROR DETECTION/CORRECTION TESTS

TEST 15 READ DATA WITH NO ERROR

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ DATA
OF 400 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE
A GOOD HEADER, 400 DATA WORDS, AND A GOOD 32 BIT ECC.
VERIFY THAT COMMAND COMPLETES WITHOUT ERROR.

TEST 16 READ DATA WITH ONE BIT BURST ERROR (PART 1)

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ DATA
OF 400 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE,
A GOOD HEADER, 400 DATA WORDS, AND AN ECC INDICATING
A ONE BIT BURST ERROR OF A 0 THAT SHOULD BE A 1.
VERIFY THE COUNTING OF THE POSITION
REGISTER AND THE FINAL PATTERN AND POSITION INFORMATION.
MAKE SURE DATA CHECK AND CONTROLLER ERROR SETS.

TEST 17 READ DATA WITH ONE BIT BURST ERROR (PART 2)

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ DATA
OF 400 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE,
A GOOD HEADER, 400 DATA WORDS, AND AN ECC INDICATING
A ONE BIT BURST ERROR OF A 1 THAT SHOULD BE A 0.
VERIFY THE COUNTING OF THE POSITION
REGISTER AND THE FINAL PATTERN AND POSITION INFORMATION.
MAKE SURE DATA CHECK AND CONTROLLER ERROR SETS.

TEST 20 READ DATA WITH 11 BIT BURST ERROR

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ DATA
OF 400 WORDS TO AN RK06 IN 26 SECTOR FORMAT,

324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370

K01

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 10
DZR6EA.P11 28-SEP-76 15:31

SEQ 0010

CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE, A GOOD HEADER, 400 DATA WORDS, AND AN ECC INDICATING AN 11 BIT BURST ERROR. VERIFY THE COUNTING OF THE POSITION REGISTER AND THE FINAL PATTERN AND POSITION INFORMATION. MAKE SURE DATA CHECK AND CONTROLLER ERROR SETS.

TEST 21 READ DATA WITH 12 BIT BURST ERROR
CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ DATA OF 400 WORDS TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE, A GOOD HEADER, 400 DATA WORDS AND AN ECC INDICATING A 12 BIT BURST ERROR. VERIFY THE COUNTING OF THE POSITION REGISTER. MAKE SURE DATA CHECK, ECC HARD, AND CONTROLLER ERROR SETS.

TEST 22 READ DATA WITH 23 BIT BURST ERROR
CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ DATA OF 400 WORDS TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE, A GOOD HEADER, 400 DATA WORDS AND AN ECC INDICATING A 23 BIT BURST ERROR. VERIFY THE COUNTING OF THE POSITION REGISTER. MAKE SURE DATA CHECK, ECC HARD, AND CONTROLLER ERROR SETS.

TEST 23 READ DATA WITH 1 BIT ECC WORD ERROR
CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
OF 400 WORDS TO AN RK06 IN 26 SECTOR FORMAT, CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE, A GOOD HEADER, 400 DATA WORDS AND AN ECC INDICATING A 1 BIT ERROR IN ECC WORD 2, BIT 6. VERIFY THE COUNTING OF THE POSITION REGISTER. MAKE SURE DATA CHECK AND CONTROLLER ERROR SETS.

TEST 24 18 BIT NPR WRITING OF MEMORY
CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ DATA OF 400 WORDS TO AN RK06 IN 24 SECTOR FORMAT, CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK

371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421

L01

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 11
DZR6EA.P11 28-SEP-76 15:31

SEG 0011

AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE,
A GOOD HEADER, 40 DATA WORDS. VERIFY THAT THE WORDS
ARE WRITTEN PROPERLY IN MEMORY.

TEST 25 18 BIT READ DATA WITH NO ERROR
CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ DATA
OF 400 WORDS TO AN RK06 IN 24 SECTOR FORMAT,
CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE,
A GOOD HEADER, 400 DATA WORDS, AND A GOOD 32 BIT ECC.
VERIFY THAT COMMAND COMPLETES WITHOUT ERROR.

TEST 26 18 BIT READ DATA WITH DCK

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ DATA
OF 400 WORDS TO AN RK06 IN 24 SECTOR FORMAT,
CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE,
A GOOD HEADER, 400 DATA WORDS, AND AN ECC INDICATING
A ONE BIT BURST ERROR BEYOND THE 4128TH BIT OF THE SECTO
VERIFY THE COUNTING OF THE POSITION REGISTER AND THE
FINAL PATTERN AND POSITION. MAKE SURE DATA CHECK AND
CONTROLLER ERROR SETS.

3491 TEST 27 18 BIT READ DATA WITH ECH

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ DATA
OF 400 WORDS TO AN RK06 IN 24 SECTOR FORMAT,
CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE,
A GOOD HEADER, 400 DATA WORDS AND AND ECC INDICATING
A 12 BIT BURST ERROR. VERIFY THE COUNTING OF THE
POSITION REGISTER. MAKE SURE DATE CHECK, ECC HARD,
AND CONTROLLER ERROR SETS.

422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463

MO1

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 12
DZR6EA.P11 28-SEP-76 15:31

SEQ 0012

**SPECIAL READ TESTS

TEST 30 PARTIAL SECTOR READ

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ DATA
OF 103 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE,
A GOOD HEADER, 400 DATA WORDS, AND A GOOD 32 BIT ECC.
MAKE SURE ONLY 103 WORDS GOT TRANSFERRED TO MEMORY.

TEST 31 SILO UNLOAD BEFORE HEADER ERROR

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ DATA
OF 401 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE,
A GOOD HEADER, 400 DATA WORDS, AND A GOOD 32 BIT ECC.
AFTER ECC SIMULATE A SECTOR PULSE AND A BAD SECTOR ERROR
MAKE SURE THE ENTIRE PREVIOUS SECTOR GETS LOADED INTO
MEMORY BEFORE BAD SECTOR ERROR AND CONTROLLER ERROR SETS

**WRITE CHECK TESTS

TEST 32 WRITE CHECK OF 400 WORDS

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
CHECK OF 400 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE,
A GOOD HEADER, THE 400 WORDS TO BE WRITE CHECKED,
AND A GOOD 32 BIT ECC. MAKE SURE NO ERRORS SET.

TEST 33 WRITE CHECK WITH DCK

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
CHECK OF 400 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE,
A GOOD HEADER, THE 400 WORDS TO BE WRITE CHECKED,
AND A 32 BIT ECC INDICATING A 1 BIT BURST ERROR.
VERIFY THAT THE POSITION REGISTER INCREMENTS PROPERLY
AND THAT THE FINAL PATTERN AND POSITION ARE CORRECT.

464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515

NO1

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 13
DZR6EA.P11 28-SEP-76 15:31

SEQ 0013

CHECK THAT DATA CHECK AND CONTROLLER ERROR ARE SET.

TEST 34 WRITE CHECK OF 18 BIT WORDS

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
CHECK OF 400 WORDS TO AN RK06 IN 24 SECTOR FORMAT,
CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK,
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE,
A GOOD HEADER, THE 400 WORDS TO BE WRITE CHECKED,
AND A GOOD 32 BIT ECC. MAKE SURE NO ERRORS SET.

TEST 35 WRITE CHECK OF A PARTIAL SECTOR

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
CHECK OF 1 WORD TO AN RK06 IN 26 SECTOR FORMAT,
CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE,
A GOOD HEADER, TAKE ONE WORD TO BE WRITE-CHECKED, 377 WO
NOT TO BE WRITE CHECKED, AND A GOOD 32 BIT ECC.
MAKE SURE NO ERRORS SET.

TEST 36 WRITE CHECK ERROR (PART 1)

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
CHECK OF 1 WORD TO AN RK06 IN 26 SECTOR FORMAT,
CLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE,
A GOOD HEADER, AND ONE WORD OF DATA CONSISTING OF 000000
WRITE CHECKED DATA IS 000001. MAKE SURE WRITE CHECK ERR
REPEAT USING EACH OF THE FOLLOWING CONFIGURATIONS
AS WRITE CHECKED DATA:

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

TEST 37 WRITE CHECK ERROR (PART 2)

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
CHECK OF 1 WORD TO AN RK06 IN 26 SECTOR FORMAT,
CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE
A GOOD HEADER, AND ONE WORD OF DATA CONSISTING OF 177777
WRITE CHECKED DATA IS 177776. MAKE SURE WRITE
CHECK ERROR SETS. REPEAT USING EACH OF THE FOLLOWING

516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571

B02

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 14
DZR6EA.P1: 28-SEP-76 15:31

SEQ 0014

572

C02

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 15
DZR6EA.P11 28-SEP-76 15:31

SEQ 0015

CONFIGURATION AS WRITE CHECKED DATA:

177775	177737	176777	157777
177773	177677	175777	137777
177767	177577	173777	077777
177757	177377	167777	

TEST 40 WRITE CHECK ERROR (PART 3)

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
CHECK OF 1 WORD TO AN RK06 IN 24 SECTOR FORMAT.
CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE.
A GOOD HEADER, AND ONE WORD OF DATA CONSISTING
OF 200000. WRITE CHECK DATA IS 000000. MAKE SURE
WRITE CHECK ERROR SETS. REPEAT USING 400000 AS
SIMULATED DATA.

TEST 41 WRITE CHECK ERROR (PART 4)

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
CHECK OF 400 WORDS TO AN RK06 IN 26 SECTOR FORMAT.
CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE.
A GOOD HEADER, 125 GOOD WORDS, AND A BAD WORD.
MAKE SURE WRITE CHECK ERROR SET. CHECK WORD COUNT
AND BUS ADDRESS.

573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606

002

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 16
DZR6EA.P11 28-SEP-76 15:31

SEQ 0016

DIAGNOSTIC MODE SECTOR PULSE

THE SECTOR PULSE IS PROVIDED BY THIS ROUTINE

DIAGNOSTIC MODE IMPLIED SEEK

THE CONTROLLER IS CLOCKED THROUGH THE SEEK MESSAGES,
GETTING STATUS, AND CLEARING THE DRIVE. SINCE THIS REQUI
DIFFERENT COUNT FOR WRITE DATA AND READ DATA/WRITE CHECK
THE ROUTINE LOOKS AT THE COMMAND THAT WAS
STARTED (L.CS1) AND SETS THE CLOCK COUNT ACCORDINGLY.

AT THE END OF THE IMPLIED SEEK THE CONTENTS OF CS1, MR1,
AND MR3 ARE CHECKED TO VERIFY THE OPERATION. ALL DISCREP
ARE REPORTED IN THE ORDER LISTED.

CALL:
JSR R4,DISEEK
ERROR 4 ;CS1 MISCOMPARE ERROR
ERROR 5 ;MR1 MISCOMPARE ERROR
ERROR 6 ;MR2 MISCOMPARE ERROR
ERROR 7 ;MR3 MISCOMPARE ERRON

DIAGNOSTIC MODE READ SYNC PATTERN (PREAMBLE)

THE 255 "0" BITS AND 1 "1" BIT ARE CLOCKED THROUTH THE R
THE CONTENTS OF MR1 ARE MONITORED TO WATCH FOR AN ERROR.
AT THE END OF THE SYNC READ, CS1 IS CHECKED TO INSURE IT
DID NOT GET CHANGED.

CALL:
JSR R4,DRSYNC
ERROR 10 ;MR1 CONTENTS IN ERROR
ERROR 11 ;CS1 CONTENTS IN ERROR

IF AN ERROR IS DETECTED, THE SUBROUTINE WILL SET UP TO A
THE PRESENT TEST UNLESS LOOP ON ERROR IS SET.

607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652

 DIAGNOSTIC MODE HEADER READ AND COMPARE

THIS ROUTINE SIMULATES THE THREE HEADER WORDS READ AND COMPARE THAT IS PART OF ALL DATA TRANSFER OPERATIONS.

THE EXPECTED HEADER IS CALCULATED FROM THE CYLINDER, TRA AND SECTOR SPECIFIED IN THE "L" REGISTERS. THE READING OF 3 WORDS IS DONE AND MR1 IS MONITORED TO INSURE IT DOES NOT CHANGE. AT THE END OF THE READ/COMPARE, THE CONTENTS OF RKCS1 IS CHECKED AND THE INCREMENT OF RKDCYL AND RKDA VERIFIED. IF THE INCREMENT IS GOOD, THE "L" REGISTERS ARE UPDATED TO THE NEW PACK ADDRESS IN CASE OF MULTI-SECTOR OPERATIONS. THE BAD SECTOR ERROR DESIRED (BSEDES) SWITCH IS TESTED AND IF IT IS NON-ZERO, THE 2ND AND 3RD HEADER WORDS ARE CHANGED ACCORDING TO BSEDES CONTENTS. THIS CAN BE USED TO FORCE HEADER TYPE ERROR.

CALL:
 JSR R4,DHDCMP
 ERROR 12 ;ERROR IN MR1
 ERROR 13 ;ERROR IN CS1
 ERROR 14 ;ERROR IN PACK ADDRESS

 DIAGNOSTIC MODE GAP READ AND SYNC WRITE

THE READING OF THE GAP AND WRITING OF THE SYNC (OR PREAM) IS HANDLED IN THIS ROUTINE.

THE FIRST GAP BITS ARE PROCESSED IN THE HEADER COMPARE ROUTINE AND THE REMAINDER ARE PROCESSED HERE. THEN A CHECK IS MADE THAT WRITE GATE CAME ON AND 255 "0" BITS AND A SINGLE "1" BIT IS WRITTEN.

CALL:
 JSR R4,DWGPSN
 ERROR 15 ;MR1 IN ERROR IN READ GAP
 ERROR 16 ;CS1 IN ERROR AFTER READ GAP
 ERROR 17 ;MR1 IN ERROR IN WRITE SYNC
 ERROR 20 ;CS1 IN ERROR AFTER SYNC WRITE

ROUTINE CALLED:
 JSR PC,WRTBIT
 RETURN POINT IF ERROR IN WRTBIT
 NO ERROR RETURN

653
 654
 655
 656
 657
 658
 659
 660
 661
 662
 663
 664
 665
 666
 667
 668
 669
 670
 671
 672
 673
 674
 675
 676
 677
 678
 679
 680
 681
 682
 683
 684
 685
 686
 687
 688
 689
 690
 691
 692
 693
 694
 695
 696
 697
 698
 699
 700
 701
 702
 703

F02

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 18
DZR6EA.P11 28-SEP-76 15:31

SEQ 0018

DIAGNOSTIC WRITE DATA AND ECC ROUTINE

THE DATA IN THE OUPUT BUFFER IS WRITE SIMULATED, ECC IS
GENERATED AND WRITTEN, AND THE POSTAMBLE IS WRITTEN.

CALL:

JSR R4,DWRITE
ERROR 17 ;MR1 IN ERROR IN WRITING
ERROR 21 ;ECC ERROR IN WRITING
ERROR 22 ;CS1 ERROR AFTER WRITE
NO ERROR RETURN

ROUTINE CALLED:

JSR PC,WRTBIT
RETURN POINT IF ERROR IN WRTBIT
NO ERROR RETURN FROM WRTBIT

DIAGNOSTIC MODE READ GAP AND DATA SYNC

THE READING OF THE GAP AND READING OF THE SYNC (OR PREAM
IS HANDLED IN THIS ROUTINE.

THE FIRST GAP BITS ARE PROCESSED IN THE HEADER COMPARE
ROUTINE AND THE REMAINING GAP BITS ARE PROCESSED HERE.
THEN A CHECK IS MADE THAT READ GATE CAME ON
AT THE 129TH BIT OF SYNC.:

CALL:

JSR R4,DRGPSN
ERROR 15 ;MR1 IN ERROR IN READ GAP
ERROR 16 ;CS1 IN ERROR AFTER READ GAP

ERROR 32 ;MR1 IN ERROR IN READ SYNC
ERROR 33 ;CS1 IN ERROR AFTER SYNC READ

ROUTINE CALLED:

JSR PC,RDBIT

704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745

G02

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 19
CZR6EA.P11 28-SEP-76 15:31

SEQ 0019

 DIAGNOSTIC READ DATA OR DIAGNOSTIC WRITE CHECK ROUTINE

THIS ROUTINE IS USED TO SIMULATE THE READ OR WRITE CHECK
 OF DATA. THE ROUTINE WILL TRANSFER A FULL OR PARTIAL SEC

THE FIRST WORD AFTER THE CALL SPECIFIES HOW MANY WORDS
 ARE TO BE TRANSFERRED.

IF LESS THAN A FULL SECTOR, THE ECC AT THE END
 OF THE TRANSFER IS NOT CHECKED. THE ECC IS CHECKED AT TH
 EACH BIT IS TRANSFERED.

EITHER 16 OR 18 BIT WORDS CAN BE SIMULATED
 DEPENDING ON THE STATE OF CFMT BIT IN L.CS1. IF 18 BITS
 ARE REQUIRED, BITS 16 AND 17 ARE ASSUMED TO BE ZERO.

THE CONTENTS OF LOCATION ECPOSX MUST BE LOADED WITH THE
 COUNT EXPECTED AT THE END OF THE DATA TRANSFER. THIS
 LOADING MUST OCCUR BEFORE THE ROUTINE IS CALLED.

ANOTHER LOCATION, ECPATX, IS SET TO 0 IF THE ECC IS EXPE
 TO BE CORRECT AND SET TO 1 IF AN ERROR IS BEING FORCED.

A 3RD LOCATION, ECCSRC, SPECIFIES THE SOURCE OF THE ECC
 WORDS TO BE USED IN CHECKING THE OPERATION. IF ECCSRC IS
 1, THE ECC WORDS FED INTO THE DECODER ARE TAKEN FROM THE
 LAST TWO LOCATIONS OF OBUFF. IF ECCSRC IS 0, THE COMPUTE
 ECC WORDS FROM ECCHI AND ECCLO ARE USED.

CALL:

JSR R4,DREAD
 LENGTH OF TRANSFER
 ERROR 34 ;MR1 IN ERROR READING DATA OR ECC
 ERROR 35 ;ECC ERROR READING DATA
 ERROR 36 ;CS1 ERROR AFTER READING DATA OR ECC
 ERROR 41 ;ECC REGISTER INCORRECT AFTER READ ECC
 ERROR 42 ;ERR IN ECC PAT CALCULATION
 ERROR 43 ;ERP IN ECC POS COUNTING

OR

JSR R4,DWRTCK
 LENGTH OF TRANSFER
 ERROR 53 ;MR1 IN ERROR WRITE CHK OR ECC READ
 ERROR 54 ;ECC ERROR IN WRITE CHECK
 ERROR 55 ;CS1 ERROR AFTER IN WRT CHK OR ECC READ
 ERROR 41 ;ECC REGISTER INCORRECT AFTER READ ECC
 ERROR 42 ;ERR IN ECC PAT CALCULATION
 ERROR 43 ;ERR IN ECC POS COUNTING

ROUTINES CALLED:

RDBIT
 ECCGEN

746
 747
 748
 749
 750
 751
 752
 753
 754
 755
 756
 757
 758
 759
 760
 761
 762
 763
 764
 765
 766
 767
 768
 769
 770
 771
 772
 773
 774
 775
 776
 777
 778
 779
 780
 781
 782
 783
 784
 785
 786
 787
 788
 789
 790
 791
 792
 793
 794
 795
 796
 797
 798
 799
 800
 801

H02

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 20
DZR6EA.P11 28-SEP-76 15:31

SEQ 0020

802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844

GENERATE ECC WORD

EACH TIME THIS ROUTINE IS CALLED THE ECC IS CALCULATED F
NEXT BIT TO BE READ OR WRITTEN AND THE BITS PREVIOUSLY
READ OR WRITTEN (STORED IN ECCHI AND ECCLO). THE RESULTS
THE CALCULATION ARE PLACED IN ECCHI AND ECCLO. THE LOW 0
11 BITS OF ECCHI ARE PLACED IN E.ECPT FOR EASY COMPARI
TO RKECPT.

CALL: JSR PC,ECCGEN
RETURN: RTS R4

SIMULATE ONE BIT OF WRITE DATA IN MAINTENANCE MODE

5835

ONE BIT OF DATA IS WRITE SIMULATED. THE CONTENTS OF
RKMRI IS COMPUTED (PRECOMP ADVANCE AND DELAY, MAINTENANC
ENCODED WRITE DATA) FOR THE PROPER TRANSITION OF THE
MAINTENANCE CLOCK AND CHECKED AT EACH TRANSITION. IF MRI
IS INCORRECT AT ANY TRANSITION, AN ERROR IS SET UP TO
CONTAIN A MESSAGE THAT IDENTIFIES THE TRANSITION WHEN TH
ERROR OCCURRED.

CALL: JSR PC,WRTBIT
RETURN: RTS PC+2 FOR NO ERROR RETURN
RTS PC FOR ERROR IN MRI

SIMULATE ONE BIT OR READ DATA IN MAINTENANCE MODE

ONE BIT OF DATA IS READ SIMULATED. NO CHECKING IS
DONE BY THIS ROUTINE.

CALL: JSR PC,RDBIT
RETURN: RTS PC

845
 846
 847
 848
 849
 850
 851
 852
 853
 854
 855
 856
 857
 858
 859
 860
 861
 862
 863
 864
 865
 866
 867
 868
 869
 870
 871
 872
 873
 874
 875
 876
 877
 878
 879
 880
 881
 882
 883
 884

 MEMORY CHECK ENABLE TRAP

IF THE PROCESSOR IN USE HAS MEMORY PARITY OPTION,
 IT WILL BE ENABLED. THIS ROUTINE WILL PROCESS TRAPS CAUS
 BY MEMORY PARITY ERRORS.

 CLEAR INPUT BUFFER

 BUILD DATA BUFFER

THE PATTERN SPECIFIED IN THE CALL IS LOADED INTO THE DAT
 BUFFER. THE ENTIRE BUFFER IS ALWAYS LOADED (400(8) WORDS

 LOAD "L" REGISTERS

THE "L" REGISTERS ARE LOADED FORM THE PARAMETERS FOLLOWI
 THE SUBROUTINE CALL. DIAGNOSTIC MODE IS SET IN L.MR1
 AND L.CS1 IS LOADED WITH THE COMMAND.

CALL:
 JSR R4,LOADRK
 .WORD CYLINDER
 .BYTE ;SECTOR
 .BYTE ;TRACK
 .WORD ;BUFFER ADDRESS
 .WORD ;WORD COUNT
 .WORD ;COMMAND,FORMAT,DRIVE TYPE,&UPPER BUS AD

RETURN:
 RTS R4

J02

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 22
DZR6EA.P11 28-SEP-76 15:31

SEQ 0022

885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918

START THE OPERATION

THE INFORMATION IN THE "L" REGISTERS ARE LOADED INTO THE
RK611 REGISTERS IN A STARAIGHT TRANSFER.

GET THE RK611 REGISTERS

ALL THE RK611 REGISTERS EXCEPT THE DATA BUFFER ARE
STORED IN THE "T" REGISTERS.

"FIRST SHALL BE LAST, LAST SHALL BE FIRST" SUBROUTINE

THE CONTENTS OF R3 IS SWAPPED END FOR END, I.E. BIT 15
BECOMES BIT 0 AND VICE VERSA, BIT 14 BECOMES BIT
1 AND VICE VERSA, ETC.

CALL:
JSR R4,FSBLVV
WITH R3 LOADED WITH THE WORD TO BE SWAPPED

RETURN:
RTS R4
WITH R3 SWAPPED.

CHECK FOR MEMORY CHECK ENABLE

K02

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 23
DZR6EA.P11 28-SEP-76 15:31

SEG 0023

```

919                                     %
920 .NLIST CND,MD,MC,TOC
921 .LIST ME
922 .ENABL ABS,AMA
923 167400 $SWR= 167400
924 000001 $TN= 1
925 .TITLE RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A
926 ;*COPYRIGHT (C) 1976
927 ;*DIGITAL EQUIPMENT CORP.
928 ;*MAYNARD, MASS. 01754
929 ;*
930 ;*PROGRAM BY MARV TEGROTHENHUIS
931 ;*
932 ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
933 ;*PACKAGE (MAINDEC-11-DZQAC-C2), SEPT 14, 1976.
934 ;*
935 .SBTTL OPERATIONAL SWITCH SETTINGS
936 ;*
937 ;* SWITCH USE
938 ;* -----
939 ;* 15 HALT ON ERROR
940 ;* 14 LOOP ON TEST
941 ;* 13 INHIBIT ERROR TYPEOUTS
942 ;* 12 ABORT PROGRAM AFTER 20 ERRORS
943 ;* 11 INHIBIT ITERATIONS
944 ;* 10 BELL ON ERROR
945 ;* 9 LOOP ON ERROR
946 ;* 8 LOOP ON TEST IN SWR<7:0>
947 .SBTTL BASIC DEFINITIONS
948 ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
949 001100 $STACK= 1100
950 .EQUIV EMT,ERROR ;;BASIC DEFINITION OF ERROR CALL
951 .EQUIV IOT,SCOPE ;;BASIC DEFINITION OF SCOPE CALL
952
953 ;*MISCELLANEOUS DEFINITIONS
954 000011 $HT= 11 ;;CODE FOR HORIZONTAL TAB
955 000012 $LF= 12 ;;CODE FOR LINE FEED
956 000015 $CR= 15 ;;CODE FOR CARRIAGE RETURN
957 000200 $CRLF= 200 ;;CODE FOR CARRIAGE RETURN-LINE FEED
958 177776 $PS= 177776 ;;PROCESSOR STATUS WORD
959 .EQUIV PS,PSW
960 $STKLMT= 177774 ;;STACK LIMIT REGISTER
961 177772 $PIRQ= 177772 ;;PROGRAM INTERRUPT REQUEST REGISTER
962 177570 $DSWR= 177570 ;;HARDWARE SWITCH REGISTER
963 177570 $DISP= 177570 ;;HARDWARE DISPLAY REGISTER
964
965 ;*GENERAL PURPOSE REGISTER DEFINITIONS
966 000000 $R0= %0 ;;GENERAL REGISTER
967 000001 $R1= %1 ;;GENERAL REGISTER
968 000002 $R2= %2 ;;GENERAL REGISTER
969 000003 $R3= %3 ;;GENERAL REGISTER
970 000004 $R4= %4 ;;GENERAL REGISTER
971 000005 $R5= %5 ;;GENERAL REGISTER
972 000006 $R6= %6 ;;GENERAL REGISTER
973 000007 $R7= %7 ;;GENERAL REGISTER
974

```

L02

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 24
DZR6EA.P11 28-SEP-76 15:31 BASIC DEFINITIONS

SEQ 0024

975	000006	SP=	%6	::STACK POINTER
976	000007	PC=	%7	::PROGRAM COUNTER
977				
978		:*PRIORITY LEVEL DEFINITIONS		
979	000000	PR0=	0	::PRIORITY LEVEL 0
980	000040	PR1=	40	::PRIORITY LEVEL 1
981	000100	PR2=	100	::PRIORITY LEVEL 2
982	000140	PR3=	140	::PRIORITY LEVEL 3
983	000200	PR4=	200	::PRIORITY LEVEL 4
984	000240	PR5=	240	::PRIORITY LEVEL 5
985	000300	PR6=	300	::PRIORITY LEVEL 6
986	000340	PR7=	340	::PRIORITY LEVEL 7
987				
988		:*"SWITCH REGISTER" SWITCH DEFINITIONS		
989	100000	SW15=	100000	
990	040000	SW14=	40000	
991	020000	SW13=	20000	
992	010000	SW12=	10000	
993	004000	SW11=	4000	
994	002000	SW10=	2000	
995	001000	SW09=	1000	
996	000400	SW08=	400	
997	000200	SW07=	200	
998	000100	SW06=	100	
999	000040	SW05=	40	
1000	000020	SW04=	20	
1001	000010	SW03=	10	
1002	000004	SW02=	4	
1003	000002	SW01=	2	
1004	000001	SW00=	1	
1005		.EQUIV	SW09,SW9	
1006		.EQUIV	SW08,SW8	
1007		.EQUIV	SW07,SW7	
1008		.EQUIV	SW06,SW6	
1009		.EQUIV	SW05,SW5	
1010		.EQUIV	SW04,SW4	
1011		.EQUIV	SW03,SW3	
1012		.EQUIV	SW02,SW2	
1013		.EQUIV	SW01,SW1	
1014		.EQUIV	SW00,SW0	
1015				
1016		:*DATA BIT DEFINITIONS (BIT00 TO BIT15)		
1017	100000	BIT15=	100000	
1018	040000	BIT14=	40000	
1019	020000	BIT13=	20000	
1020	010000	BIT12=	10000	
1021	004000	BIT11=	4000	
1022	002000	BIT10=	2000	
1023	001000	BIT09=	1000	
1024	000400	BIT08=	400	
1025	000200	BIT07=	200	
1026	000100	BIT06=	100	
1027	000040	BIT05=	40	
1028	000020	BIT04=	20	
1029	000010	BIT03=	10	
1030	000004	BIT02=	4	

```

1031      000002      BIT01= 2
1032      000001      BIT00= 1
1033      .EQUIV      BIT09,BIT9
1034      .EQUIV      BIT08,BIT8
1035      .EQUIV      BIT07,BIT7
1036      .EQUIV      BIT06,BIT6
1037      .EQUIV      BIT05,BIT5
1038      .EQUIV      BIT04,BIT4
1039      .EQUIV      BIT03,BIT3
1040      .EQUIV      BIT02,BIT2
1041      .EQUIV      BIT01,BIT1
1042      .EQUIV      BIT00,BIT0
1043
1044      ;*BASIC "CPU" TRAP VECTOR ADDRESSES
1045      000004      ERRVEC= 4      ;; TIME OUT AND OTHER ERRORS
1046      000010      RESVEC= 10     ;; RESERVED AND ILLEGAL INSTRUCTIONS
1047      000014      TBITVEC=14     ;; "T" BIT
1048      000014      TRIVEC= 14     ;; TRACE TRAP
1049      000014      BPTVEC= 14     ;; BREAKPOINT TRAP (BPT)
1050      000020      IOTVEC= 20     ;; INPUT/OUTPUT TRAP (IOT) **SCOPE**
1051      000024      PWRVEC= 24     ;; POWER FAIL
1052      000030      EMTVEC= 30     ;; EMULATOR TRAP (EMT) **ERROR**
1053      000034      TRAPVEC=34     ;; "TRAP" TRAP
1054      000060      TKVEC= 60      ;; TTY KEYBOARD VECTOR
1055      000064      TPVEC= 64      ;; TTY PRINTER VECTOR
1056      000240      PIRQVEC=240    ;; PROGRAM INTERRUPT REQUEST VECTOR
1057      .SBTTL      MEMORY MANAGEMENT DEFINITIONS
1058
1059      ;*KT11 VECTOR ADDRESS
1060
1061      000250      MMVEC= 250
1062
1063      ;*KT11 STATUS REGISTER ADDRESSES
1064
1065      177572      SR0= 177572
1066      177574      SR1= 177574
1067      177576      SR2= 177576
1068      172516      SR3= 172516
1069
1070      ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
1071
1072      172300      KIPDR0= 172300
1073      172302      KIPDR1= 172302
1074      172304      KIPDR2= 172304
1075      172306      KIPDR3= 172306
1076      172310      KIPDR4= 172310
1077      172312      KIPDR5= 172312
1078      172314      KIPDR6= 172314
1079      172316      KIPDR7= 172316
1080
1081      ;*KERNEL "I" PAGE ADDRESS REGISTERS
1082
1083      172340      KIPAR0= 172340
1084      172342      KIPAR1= 172342
1085      172344      KIPAR2= 172344
1086      172346      KIPAR3= 172346

```

N02

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 26
DZR6EA.P11 28-SEP-76 15:31 MEMORY MANAGEMENT DEFINITIONS

SEQ 0026

1087	172350	KIPAR4= 172350	
1088	172352	KIPAR5= 172352	
1089	172354	KIPAR6= 172354	
1090	172356	KIPAR7= 172356	
1091			
1092	000114	MEMVEC= 114	; MEMORY PARITY VECTOR
1093	172100	MEMBAS= 172100	; MEM PARITY OPTION
1094	000004	WR.PAR= 4	; WRITE BAD PARITY
1095	000001	PAR.EN= 1	; ENABLE PARITY ENABLE
1096	120210	AVECT1= 120210	; DEFINE RK611 VECTOR ADDRESS
1097	000005	APRIOR= 5	; DEFINE RK611 PRIORITY
1098	177440	ABASE= 177440	; DEFINE BASE OF RK611 REGISTERS
1099			
1100		.SBTTL RK611 CONTROLLER REGISTER DEFINITION	
1101			
1102	000000	RKCS1= 0	; CONTROL AND STATUS REGISTER 1
1103	000002	RKWC= 2	; WORD COUNT REGISTER
1104	000004	RKBA= 4	; BUS ADDRESS REGISTER
1105	000006	RKDA= 6	; DESIRED TRACK SECTOR REGISTER
1106	000010	RKCS2= 10	; CONTROL AND STATUS REGISTER 2
1107	000012	RKDS= 12	; DRIVE STATUS REGISTER
1108	000014	RKER= 14	; ERROR REGISTER
1109	000016	RKASOF= 16	; ATTENTION SUMMARY AND OFFSET REGISTER
1110	000020	RKDCYL= 20	; DESIRED CYLINDER REGISTER
1111	000024	RKDB= 24	; DATA BUFFER
1112	000026	RKMR1= 26	; MAINTENANCE REGISTER 1
1113	000034	RKMR2= 34	; MAINTENANCE REGISTER 2
1114	000036	RKMR3= 36	; MAINTENANCE REGISTER 3
1115	000030	RKECPS= 30	; ECC POSITION INFORMATION
1116	000032	RKECPT= 32	; ECC PATTERN INFORMATION
1117	000022	RKSPAR= 22	; SPARE REGISTER
1118			
1119		.SBTTL DRIVE COMMANDS	
1120			
1121	000001	SELDRV= 01	; SELECT DRIVE
1122	000003	PACK= 03	; PACK ACKNOWLEDGE
1123	000005	CLEAR= 05	; DRIVE CLEAR
1124	000007	UNLOAD= 07	; UNLOAD
1125	000011	SRTSPL= 11	; START SPINDLE
1126	000013	RECAL= 13	; RECALIBRATE
1127	000015	OFFSET= 15	; OFFSET
1128	000017	SEEK= 17	; SEEK
1129	000021	RDDATA= 21	; READ DATA
1130	000023	WRDATA= 23	; WRITE DATA
1131	000025	RDHEAD= 25	; READ HEADER
1132	000027	WRHEAD= 27	; WRITE HEADER AND DATA
1133	000031	WRTCHK= 31	; WRITE CHECK
1134	000300	INTR= 300	; GENERATE INTERRUPT TO CPU
1135			
1136		.SBTTL CONTROL AND STATUS REGISTER 1 BITS	
1137			
1138	000001	GO= BIT0	; GO BIT
1139	000100	IE= BIT6	; INTERRUPT ENABLE
1140	000200	RDY= BIT7	; CONTROLLER READY
1141	000400	BA16= BIT8	; BUS ADDRESS BIT 16
1142	001000	BA17= BIT9	; BUS ADDRESS BIT 17

B03

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZ6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 27
 DZ6EA.P11 28-SEP-76 15:31 CONTROL AND STATUS REGISTER 1 BITS

SEQ 0027

1143	002000	CDT=	BIT10	: CONTROLLER DRIVE TYPE (0=RK06,
1144	004000	CTC=	BIT11	: CONTROLLER TIMED OUT WAITING FOR
1145				: DRIVE RESPONSE
1146	010000	CFMT=	BIT12	: CONTROLLER DRIVE FORMAT (0=26 SECTOR, 1=24 SECTOR,
1147	020000	SPAR=	BIT13	: DRIVE BUS PARITY ERROR DETECTED BY CONTROLLER
1148	040000	DI=	BIT14	: DRIVE INTERRUPT
1149	100000	CERR=	BIT15	: CONTROLLER ERROR
1150	100000	CCLR=	BIT15	: CONTROLLER CLEAR

.SBTTL CONTROL AND STATUS REGISTER 2 BITS

1151	000007	DRVMSK=	7	: MASK FOR DRIVE SELECTION CODE
1152	000010	RLS=	BIT3	: DESELECT OR RELEASE DRIVE IN BITS 0-2
1153	000020	BAI=	BIT4	: BUS ADDRESS INCREMENT INHIBIT
1154	000040	SCLR=	BIT5	: CLEAR CONTROLLER AND ALL DRIVES
1155	000100	IR=	BIT6	: INPUT READY
1156	000200	OR=	BIT7	: OUTPUT READY
1157	000400	UFE=	BIT8	: UNIT FIELD ERROR
1158	001000	MDS=	BIT9	: MULTIPLE DRIVE SELECT
1159	002000	PGE=	BIT10	: PROGRAMMING ERROR
1160	004000	NEM=	BIT11	: NON-EXISTENT MEMORY
1161	010000	NED=	BIT12	: NON-EXISTENT DRIVE
1162	020000	UPE=	BIT13	: UNIBUS PARITY ERROR
1163	040000	WCE=	BIT14	: WRITE CHECK ERROR
1164	100000	DLT=	BIT15	: DATA LATE ERROR

.SBTTL ERROR REGISTER BIT DEFINITION

1170	000001	ILF=	BIT0	: ILLEGAL FUNCTION CODE
1171	000002	SKI=	BIT1	: SEEK INCOMPLETE
1172	000004	NXF=	BIT2	: NON-EXECUTABLE DRIVE FUNCTION
1173	000010	DRPAR=	BIT3	: DRIVE DETECTED DRIVE BUS PARITY ERROR
1174	000020	FMTE=	BIT4	: FORMAT ERROR
1175	000040	DTYE=	BIT5	: DRIVE TYPE ERROR
1176	000100	ECH=	BIT6	: ECC HARD
1177	000200	BSE=	BIT7	: BAD SECTOR ERROR
1178	000400	HVRC=	BIT8	: HEADER VRC ERROR
1179	001000	COE=	BIT9	: CYLINDER ADDRESS OVERFLOW ERROR
1180	002000	IDAE=	BIT10	: INVALID DISK ADDRESS ERROR
1181	004000	WLE=	BIT11	: WRITE LOCK ERROR
1182	010000	DTE=	BIT12	: DRIVE TIMING ERROR
1183	020000	OPI=	BIT13	: OPERATION (SEARCH) INCOMPLETE
1184	040000	UNS=	BIT14	: DRIVE UNSAFE
1185	100000	DCK=	BIT15	: DATA CHECK

.SBTTL STATUS REGISTER BIT DEFINITION

1188				
1189				
1190	000001	DRA=	BIT0	: DRIVE AVAILABLE (CONTROLLER IS SET IF
1191				: THIS BIT IS RESET)
1192	000004	OFST=	BIT2	: DRIVE OFFSET
1193	000010	ACLO=	BIT3	: AC LOW
1194	000020	SPDLSS=	BIT4	: SPEED LOSS
1195	000040	DROT=	BIT5	: DRIVE OFF TRACK
1196	000100	VV=	BIT6	: VOLUME VALID
1197	000200	DRDY=	BIT7	: DRIVE READY
1198	000400	DOT=	BIT8	: DRIVE TYPE (0=RK06)

C03

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZ6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 28
 DZ6EA.P11 28-SEP-76 15:31 STATUS REGISTER BIT DEFINITION

SEQ 0028

```

1199      004000      WRL=      BIT11      ;WRITE LOCK
1200      020000      PIP=      BIT13      ;POSITIONING IN PROGRESS
1201      040000      DSC=      BIT14      ;DRIVE STATUS CHANGE
1202      100000      SVAL=     BIT15      ;STATUS VALID
1203
1204      .SBTTL  MAINTENANCE REGISTER 1 BIT DEFINITION
1205
1206      000017      MESMSK= 17      ;MESSAGE MASK
1207
1208      000020      PAT=      BIT4       ;FORCE EVEN PARITY ON DRIVE MESSAGE LINES
1209      000040      DMD=      BIT5       ;DIAGNOSTIC MODE
1210      000100      MSP=      BIT6       ;MAINTENANCE SECTOR PULSE
1211      000200      MIND=     BIT7       ;MAINTENANCE INDEX
1212      000400      MCLK=     BIT8       ;MAINTENANCE CLOCK
1213      001000      MERD=     BIT9       ;MAINTENANCE ENCODED READ DATA
1214      002000      MEWD=     BIT10      ;MAINTENANCE ENCODED WRITE DATA
1215      004000      PCA=      BIT11      ;PRECOMPENSATION ADVANCE
1216      010000      PCD=      BIT12      ;PRECOMPENSATION DELAY
1217      020000      ECCW=     BIT13      ;ECC WORD IS BEING READ OR WRITTEN
1218      040000      WRTGAT=   BIT14      ;WRITE GATE
1219      100000      RDGATE=   BIT15      ;READ GATE
1220
1221      .SBTTL  TRANSMITTED MESSAGE A
1222
1223      000020      S.SEEK=    BIT4       ;SEEK COMMAND
1224      000040      S.RECL=    BIT5       ;RECALIBRATE COMMAND
1225      000100      S.STSP=    BIT6       ;START SPINDLE COMMAND
1226      000200      S.RTC=     BIT7       ;DRIVE RETURN TO CENTERLINE COMMAND
1227      000400      S.CLR=     BIT8       ;CLEAR ERROR AND DSC
1228      001000      S.FMT=     BIT9       ;FORMAT
1229      002000      S.UNLD=     BIT10      ;UNLOAD
1230      004000      S.PACK=    BIT11      ;SET VOLUME VALID (PACK ACKNOWLEDGE)
1231
1232      .SBTTL  TRAP CATCHER
1233
1234      .=0
1235      ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
1236      ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
1237      ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
1238      .=174
1239      DISPREG: .WORD 0      ;:SOFTWARE DISPLAY REGISTER
1240      SWREG:   .WORD 0      ;:SOFTWARE SWITCH REGISTER
1241      .SBTTL  STARTING ADDRESS(ES)
1242      JMP      2*START ;:JUMP TO STARTING ADDRESS OF PROGRAM
1243      JMP      RESTRT  ;:JUMP TO RESTART ROUTINE
1244      .=214
1245      JMP      PARM    ;:JUMP TO OPERATOR ASSIGNED PARAMETERS
1246
1247      .SBTTL  ACT11 HOOKS
1248
1249      ;*****
1250      ;HOOKS REQUIRED BY ACT11
1251      $SVPC=.      ;SAVE PC
1252      .=46
1253      $ENDAD      ;:1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
1254      .=52
1255      .WORD 0      ;:2)SET LOC.52 TO ZERO
1256      .=$SVPC      ;: RESTORE PC

```

D03

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:10 PAGE 29
DZR6EA.P11 28-SEP-76 15:31 ACT11 HOOKS

SEQ 0029

1255		000114	. = MEMVEC
1256	000114	033524	MEMERR
1257	000116	000340	PR7
1258		001000	. = 1000
1259			.SBTTL APT PARAMETER BLOCK
1260			
1261			::*****
1262			::SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
1263			::*****
1264		001000	.SX= . ::SAVE CURRENT LOCATION
1265		000024	. = 24 ::SET POWER FAIL TO POINT TO START OF PROGRAM
1266	000024	000200	200 ::FOR APT START UP
1267		000044	. = 44 ::POINT TO APT INDIRECT ADDRESS PNTR.
1268	000044	001000	\$APTHDR ::POINT TO APT HEADER BLOCK
1269		001000	. = .SX ::RESET LOCATION COUNTER
1270			::*****
1271			::SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
1272			::INTERFACE SPEC.
1273			
1274	001000		\$APTHD:
1275	001000	000000	\$HIBTS: .WORD 0 ::TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
1276	001002	001250	\$MBADR: .WORD \$MAIL ::ADDRESS OF APT MAILBOX (BITS 0-15)
1277	001004	000132	\$TSTM: .WORD 90. ::RUN TIM OF LONGEST TEST
1278	001006	001604	\$PASTM: .WORD 900. ::RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
1279	001010	000144	\$UNITH: .WORD 100. ::ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1280	001012	000030	.WORD \$ETEND-\$MAIL/2 ::LENGTH MAILBOX-ETABLE(WORDS)

E03

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 30
 DZR6EA.P11 28-SEP-76 15:31 COMMON TAGS

SEQ 0030

```

1281 .SBTTL COMMON TAGS
1282
1283 ::*****
1284 ::THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
1285 ::USED IN THE PROGRAM.
1286
1287 001100
1288 001100 000000 $CMTAG: .=1100
1289 001100 000000
1290 001102 000 $STSTNM: .WORD 0
1291 001103 000 $SERFLG: .BYTE 0
1292 001104 000000 $SICNT: .WORD 0
1293 001106 000000 $SLPADR: .WORD 0
1294 001110 000000 $SLPERR: .WORD 0
1295 001112 000000 $SERTTL: .WORD 0
1296 001114 000 $SITEMB: .BYTE 0
1297 001115 001 $SERMAX: .BYTE 1
1298 001116 000000 $SERRPC: .WORD 0
1299 001120 000000 $SGADR: .WORD 0
1300 001122 000000 $SBDADR: .WORD 0
1301 001124 000000 $SGDDAT: .WORD 0
1302 001126 000000 $SDDAT: .WORD 0
1303 001130 000000
1304 001132 000000
1305 001134 000 $SAUTOB: .BYTE 0
1306 001135 000 $SINTAG: .BYTE 0
1307 001136 000000
1308 001140 177570 $SWR: .WORD 0SWR
1309 001142 177570 $DISPLAY: .WORD 0DISP
1310 001144 177560 $TKS: 177560
1311 001146 177562 $TKB: 177562
1312 001150 177564 $TPS: 177564
1313 001152 177566 $TPB: 177566
1314 001154 000 $NULL: .BYTE 0
1315 001155 002 $FILLS: .BYTE 2
1316 001156 012 $FILLC: .BYTE 12
1317 001157 000 $TPFLG: .BYTE 0
1318 001160 000000 $REGAD: .WORD 0
1319
1320 001162 000000 $REG0: .WORD 0
1321 001164 000000 $REG1: .WORD 0
1322 001166 000000 $REG2: .WORD 0
1323 001170 000000 $REG3: .WORD 0
1324 001172 000000 $REG4: .WORD 0
1325 001174 000000 $REG5: .WORD 0
1326 001176 000000 $REG6: .WORD 0
1327 001200 000000 $REG7: .WORD 0
1328 001202 000000 $TMP0: .WORD 0
1329 001204 000000 $TMP1: .WORD 0
1330 001206 000000 $TMP2: .WORD 0
1331 001210 000000 $TMP3: .WORD 0
1332 001212 000000 $TMP4: .WORD 0
1333 001214 000000 $TMP5: .WORD 0
1334 001216 000000 $TMP6: .WORD 0
1335 001220 000000 $TMP7: .WORD 0
1336 001222 000000 $TMP10: .WORD 0

```

:: START OF COMMON TAGS

:: CONTAINS THE TEST NUMBER

:: CONTAINS ERROR FLAG

:: CONTAINS SUBTEST ITERATION COUNT

:: CONTAINS SCOPE LOOP ADDRESS

:: CONTAINS SCOPE RETURN FOR ERRORS

:: CONTAINS TOTAL ERRORS DETECTED

:: CONTAINS ITEM CONTROL BYTE

:: CONTAINS MAX. ERRORS PER TEST

:: CONTAINS PC OF LAST ERROR INSTRUCTION

:: CONTAINS ADDRESS OF 'GOOD' DATA

:: CONTAINS ADDRESS OF 'BAD' DATA

:: CONTAINS 'GOOD' DATA

:: CONTAINS 'BAD' DATA

:: RESERVED--NOT TO BE USED

:: AUTOMATIC MODE INDICATOR

:: INTERRUPT MODE INDICATOR

:: ADDRESS OF SWITCH REGISTER

:: ADDRESS OF DISPLAY REGISTER

:: TTY KBD STATUS

:: TTY KBD BUFFER

:: TTY PRINTER STATUS REG. ADDRESS

:: TTY PRINTER BUFFER REG. ADDRESS

:: CONTAINS NULL CHARACTER FOR FILLS

:: CONTAINS # OF FILLER CHARACTERS REQUIRED

:: INSERT FILL CHARS. AFTER A "LINE FEED"

:: "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)

:: CONTAINS THE ADDRESS FROM

:: WHICH (\$REG0) WAS OBTAINED

:: CONTAINS ((\$REGAD)+0)

:: CONTAINS ((\$REGAD)+2)

:: CONTAINS ((\$REGAD)+4)

:: CONTAINS ((\$REGAD)+6)

:: CONTAINS ((\$REGAD)+10)

:: CONTAINS ((\$REGAD)+12)

:: CONTAINS ((\$REGAD)+14)

:: CONTAINS ((\$REGAD)+16)

:: USER DEFINED

:: USER DEFINED

:: USER DEFINED

:: USER DEFINED

:: USER DEFINED

:: USER DEFINED

:: USER DEFINED

:: USER DEFINED

:: USER DEFINED

F03

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 31
 DZR6EA.P11 28-SEP-76 15:31 COMMON TAGS

SEQ 0031

1337	001224	000000	\$TMP11: .WORD	0	::USER DEFINED
1338	001226	000000	\$TMP12: .WORD	0	::USER DEFINED
1339	001230	000000	\$TMP13: .WORD	0	::USER DEFINED
1340	001232	000000	\$TMP14: .WORD	0	::USER DEFINED
1341	001234	000000	\$TIMES: 0		::MAX. NUMBER OF ITERATIONS
1342	001236	000000	\$ESCAPE: 0		::ESCAPE ON ERROR ADDRESS
1343	001240	177607	\$BELL: .ASCIIZ	(207)(377)(377)	::CODE FOR BELL
1344	001244	077	\$QUES: .ASCII	??	::QUESTION MARK
1345	001245	015	\$CRLF: .ASCII	(15)	::CARRIAGE RETURN
1346	001246	000012	\$LF: .ASCIIZ	(12)	::LINE FEED
1347			:*****		
1348			:SBTTL APT MAILBOX-ETABLE		
1349			:*****		
1350			:EVEN		
1351			\$MAIL:		::APT MAILBOX
1352	001250		\$MSGTY: .WORD	AMSGTY	::MESSAGE TYPE CODE
1353	001250	000000	\$FATAL: .WORD	AFATAL	::FATAL ERROR NUMBER
1354	001252	000000	\$TESTN: .WORD	ATESTN	::TEST NUMBER
1355	001254	000000	\$PASS: .WORD	APASS	::PASS COUNT
1356	001256	000000	\$DEVCT: .WORD	ADEVCT	::DEVICE COUNT
1357	001260	000000	\$UNIT: .WORD	AUNIT	::I/O UNIT NUMBER
1358	001262	000000	\$MSGAD: .WORD	AMSGAD	::MESSAGE ADDRESS
1359	001264	000000	\$MSGLG: .WORD	AMSGLG	::MESSAGE LENGTH
1360	001266	000000	\$ETABLE:		::APT ENVIRONMENT TABLE
1361	001270		\$ENV: .BYTE	AENV	::ENVIRONMENT BYTE
1362	001270	000	\$ENVN: .BYTE	AENVN	::ENVIRONMENT MODE BITS
1363	001271	000	\$SWREG: .WORD	ASWREG	::APT SWITCH REGISTER
1364	001272	000000	\$USWR: .WORD	AUSWR	::USER SWITCHES
1365	001274	000000	\$CPUOP: .WORD	ACPUOP	::CPU TYPE, OPTIONS
1366	001276	000000			BITS 15-11=CPU TYPE
1367					11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
1368					11/70=06, PDQ=07, Q=10
1369					BIT 10=REAL TIME CLOCK
1370					BIT 9=FLOATING POINT PROCESSOR
1371					BIT 8=MEMORY MANAGEMENT
1372			\$MAMS1: .BYTE	AMAMS1	::HIGH ADDRESS, M.S. BYTE
1373	001300	000	\$MTYP1: .BYTE	AMTYP1	::MEM. TYPE, BLK#1
1374	001301	000			MEM. TYPE BYTE -- (HIGH BYTE)
1375					900 NSEC CORE=001
1376					300 NSEC BIPOLAR=002
1377					500 NSEC MOS=003
1378			\$MADR1: .WORD	AMADR1	::HIGH ADDRESS, BLK#1
1379	001302	000000			MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
1380			\$MAMS2: .BYTE	AMAMS2	::HIGH ADDRESS, M.S. BYTE
1381	001304	000	\$MTYP2: .BYTE	AMTYP2	::MEM. TYPE, BLK#2
1382	001305	000	\$MADR2: .WORD	AMADR2	::MEM. LAST ADDRESS, BLK#2
1383	001306	000000	\$MAMS3: .BYTE	AMAMS3	::HIGH ADDRESS, M.S. BYTE
1384	001310	000	\$MTYP3: .BYTE	AMTYP3	::MEM. TYPE, BLK#3
1385	001311	000	\$MADR3: .WORD	AMADR3	::MEM. LAST ADDRESS, BLK#3
1386	001312	000000	\$MAMS4: .BYTE	AMAMS4	::HIGH ADDRESS, M.S. BYTE
1387	001314	000	\$MTYP4: .BYTE	AMTYP4	::MEM. TYPE, BLK#4
1388	001315	000	\$MADR4: .WORD	AMADR4	::MEM. LAST ADDRESS, BLK#4
1389	001316	000000	\$VECT1: .WORD	AVECT1	::INTERRUPT VECTOR#1, BUS PRIORITY#1
1390	001320	120210	\$VECT2: .WORD	AVECT2	::INTERRUPT VECTOR#2, BUS PRIORITY#2
1391	001322	000000	\$BASE: .WORD	ABASE	::BASE ADDRESS OF EQUIPMENT UNDER TEST
1392	001324	177440			

G03

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZRG6-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 32
DZRG6A.P11 28-SEP-76 15:31 APT MAILBOX-ETABLE

SEQ 0032

1393 001326 000000
1394 001330
1395

SDEV: .WORD ADEV: ;;DEVICE MAP
SETEND:
.MEXIT

H03

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 33
 DZR6EA.P11 28-SEP-76 15:31 ERROR POINTER TABLE

SEQ 0033

```

1396      .SBTTL  ERROR POINTER TABLE
1397
1398      ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
1399      ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
1400      ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
1401      ;*NOTE1:      IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
1402      ;*NOTE2:      EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
1403
1404      ;*      EM      ::POINTS TO THE ERROR MESSAGE
1405      ;*      DH      ::POINTS TO THE DATA HEADER
1406      ;*      DT      ::POINTS TO THE DATA
1407      ;*      DF      ::POINTS TO THE DATA FORMAT
1408
1409
1410      $ERRTB:
1411      ;      ERROR 1
1412      001330 044512      EM000      ;UNEXPECTED PARITY ERROR
1413      001332 043071      DH000C
1414      001334 041316      DT000
1415      001336 041634      DF000
1416      ;      ERROR 2
1417      001340 000000      0      ;UNUSED
1418      001342 000000      0
1419      001344 000000      0
1420      001346 000000      0
1421      ;      ERROR 3
1422      001350 000000      0      ;UNUSED
1423      001352 000000      0
1424      001354 000000      0
1425      001356 000000      0
1426      ;      ERROR 4
1427      001360 044557      EM5004      ;ERROR ATTEMPTING IMPLIED SEEK
1428      001362 044623      E5004A      ;CSI IN ERROR
1429      001364 041322      DT0004
1430      001366 041640      DF0004
1431
1432      ;      ERROR 5
1433      001370 044557      EM5004      ;MR1 IN ERROR ATTEMPTING IMPLIED SEEK
1434      001372 044642      E5004B
1435      001374 041340      DT0005
1436      001376 041670      DF0005
1437
1438      ;      ERROR 6
1439      001400 044557      EM5004      ;MR2 IN ERROR ATTEMPTING IMPLIED SEEK
1440      001402 044661      E5004C
1441      001404 041350      DT0006
1442      001406 041714      DF0006
1443
1444      ;      ERROR 7
1445      001410 044557      EM5004      ;MR3 IN ERROR ATTEMPTING IMPLIED SEEK
1446      001412 044700      E5004D
1447      001414 041360      DT0007
1448      001416 041740      DF0007
1449
1450      ;      ERROR 10
1451      001420 044771      EM5010      ;MR1 ERROR READING PREAMBLE
  
```

1452	001422	044642	E50048	
1453	001424	041370	DT0010	
1454	001426	041764	DF0010	
1455				
1456			ERROR 11	;CS1 ERROR READING PREAMBLE
1457	001430	044771	EM5010	
1458	001432	044623	E5004A	
1459	001434	041406	DT0011	
1460	001436	042010	DF0011	
1461				
1462			ERROR 12	;MR1 IN ERROR HEADER READ/COMPARE
1463	001440	045051	EM5011	
1464	001442	044642	E50048	
1465	001444	041370	DT0010	
1466	001446	041764	DF0010	
1467				
1468			ERROR 13	;CS1 IN ERROR HEADER READ/COMPARE
1469	001450	045051	EM5011	
1470	001452	044623	E5004A	
1471	001454	041406	DT0011	
1472	001456	042010	DF0011	
1473				
1474			ERROR 14	;PACK ADDRESS ERROR HDR READ/COMPARE
1475	001460	045051	EM5011	
1476	001462	045124	E5011A	
1477	001464	041432	DT0014	
1478	001466	042040	DF0014	
1479				
1480			ERROR 15	;MR1 ERROR READING GAP
1481	001470	045154	EM5015	
1482	001472	044642	E50048	
1483	001474	041370	DT0010	
1484	001476	041764	DF0010	
1485				
1486			ERROR 16	;CS1 ERROR AFTER READING GAP
1487	001500	044771	EM5010	
1488	001502	044623	E5004A	
1489	001504	041406	DT0011	
1490	001506	042010	DF0011	
1491				
1492			ERROR 17	;ERROR IN WRITING
1493	001510	000000	0	;REGISTER IN ERROR AND CLOCK
1494	001512	000000	0	;TRANSITION IS IDENTIFIED
1495	001514	041446	DT0017	
1496	001516	042064	DF0017	
1497				
1498			ERROR 20	;ERROR IN CS1 AFTER WRITING SYNC
1499	001520	045211	EM5020	
1500	001522	044623	E5004A	
1501	001524	041406	DT0011	
1502	001526	042010	DF0011	
1503				
1504			ERROR 21	;ERROR IN ECC WORDS
1505	001530	045270	EM5021	
1506	001532	045336	E5021A	
1507	001534	041470	DT0021	

J03

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 35
 DZR6EA.P11 28-SEP-76 15:31 ERROR POINTER TABLE

SEQ 0035

1508	001536	042110	DF0021	
1509				
1510			;	ERROR 22 ;ERR IN CS1 AFTER WRITE DATA
1511	001540	045270	EM5021	
1512	001542	044623	E5004A	
1513	001544	041406	DT0011	
1514	001546	042010	DF0011	
1515				
1516			;	ERROR 23 ;ERROR IN MR2
1517	001550	045366	EM5023	;
1518	001552	044661	E5004C	DOING MULTI-SECTOR WRITE
1519	001554	041350	DT0006	
1520	001556	041714	DF0006	
1521				
1522			;	ERROR 24 ;MR3 IN ERROR
1523	001560	045366	EM5023	;
1524	001562	044700	E5004D	DOING MULTI-SECTOR WRITE
1525	001564	041360	DT0007	
1526	001566	041740	DF0007	
1527				
1528			;	ERROR 25 ;CS1 IN ERROR
1529	001570	045366	EM5023	;
1530	001572	044623	E5004A	AFTER MULTI-SECTOR WRITE
1531	001574	041322	DT0004	
1532	001576	041640	DF0004	
1533				
1534			;	ERROR 26
1535	001600	046006	EM5026	;
1536	001602	000000	.WORD	0 ;NO COE ERROR
1537	001604	041502	DT0026	
1538	001606	042134	DF0026	
1539				
1540			;	ERROR 27
1541	001610	046146	EM5027	;
1542	001612	043040	DH000A	OTHER ERROR WHILE FORCING COE
1543	001614	041502	DT0026	
1544	001616	042134	DF0026	
1545				
1546			;	ERROR 30
1547	001620	046320	EM5030	;
1548	001622	000000	.WORD	0 ;UNEXPECTED CYL OVERFLOW
1549	001624	041502	DT0026	
1550	001626	042134	DF0026	
1551				
1552			;	ERROR 31
1553	001630	046413	EM5031	;
1554	001632	000000	.WORD	0 ;UNEXPECTED ERROR TESTING COE
1555	001634	041502	DT0026	
1556	001636	042134	DF0026	
1557				
1558			;	ERROR 32
1559	001640	046500	EM5032	;
1560	001642	044642	E50048	MR1 ERROR READING DATA SYNC
1561	001644	041370	DT0010	
1562	001646	041764	DF0010	
1563				

K03

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 36
DZR6EA.P11 28-SEP-76 15:31 ERROR POINTER TABLE

SEQ 0036

1564			;	ERROR 33	;CS1 ERROR AFTER READING SYNC
1565	001650	046500		EMS032	
1566	001652	044623		E5004A	
1567	001654	041406		DT0011	
1568	001656	042010		DF0011	
1569					
1570			;	ERROR 34	;MRI IN ERROR READING DATA OR ECC
1571	001660	046556		EMS034	
1572	001662	044642		E50048	
1573	001664	041370		DT0010	
1574	001666	041764		DF0010	
1575					
1576			;	ERROR 35	;ECC ERROR READING DATA
1577	001670	046623		EMS035	
1578	001672	045336		E5021A	
1579	001674	041470		DT0021	
1580	001676	042110		DF0021	
1581					
1582			;	ERROR 36	;CS1 ERROR AFTER READING DATA OR ECC
1583	001700	046556		EMS034	
1584	001702	044623		E5004A	
1585	001704	041406		DT0011	
1586	001706	042010		DF0011	
1587					
1588			;	ERROR 37	;DATA COMPARE ERROR (1ST MESSAGE)
1589	001710	046664		EMS037	
1590	001712	043040		DH000A	
1591	001714	041524		DT0037	
1592	001716	042170		DF0037	
1593					
1594			;	ERROR 40	;DATA COMPARE ERROR (2ND THRU 10TH ERR)
1595	001720	000000		0	
1596	001722	000000		0	
1597	001724	041530		DT0040	
1598	001726	042210		DF0040	
1599					
1600			;	ERROR 41	;ECC REGISTER INCORRECT AFTER ECC READ
1601	001730	046740		EMS041	
1602	001732	043040		DH000A	
1603	001734	041536		DT0041	
1604	001736	042214		DF0041	
1605					
1606			;	ERROR 42	;ERROR IN ECC PAT CALCULATION AFTER ECC ERROR
1607	001740	047025		EMS042	
1608	001742	043040		DH000A	
1609	001744	041470		DT0021	
1610	001746	042234		DF0042	
1611					
1612			;	ERROR 43	;ERROR IN ECC POSITION COUNTING AFTER ECC ERROR
1613	001750	047106		EMS043	
1614	001752	043040		DH000A	
1615	001754	041554		DT0043	
1616	001756	042254		DF0043	
1617					
1618			;	ERROR 44	;ERROR AFTER PROCESSING DATA CHECK ERR
1619	001760	047165		EMS044	

L03

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 37
 DZR6EA.P11 28-SEP-76 15:31 ERROR POINTER TABLE

SEQ 0037

1620	001762	044623	E5004A	
1621	001764	041322	DT0004	
1622	001766	041640	DF0004	
1623				
1624			;	ERROR 45
1625	001770	047165	EM5044	;ERROR AFTER PROCESSING DATA CHECK ERR
1626	001772	044717	E5004E	
1627	001774	041566	DT0045	
1628	001776	042274	DF0045	
1629				
1630			;	ERROR 46
1631	002000	044557	EM50046	;TRANSFER LENGTH ERROR PARTIAL SEC READ
1632	002002	043040	DH000A	
1633	002004	041576	DT0046	
1634	002006	042320	DF0046	
1635				
1636			;	ERROR 47
1637	002010	047327	EM5047	;BSE DID NOT PREVENT DA OR DC INCREMENT
1638	002012	045121	E5011A	
1639	002014	041432	DT0014	
1640	002016	042040	DF0014	
1641				
1642			;	ERROR 50
1643	002020	047422	EM5050	;BSE DID NOT CAUSE CERR
1644	002022	044623	E5004A	
1645	002024	041322	DT0004	
1646	002026	042340	DF0050	
1647				
1648			;	ERROR 51
1649	002030	047475	EM5051	;BSE FAILED TO SET WHEN EXPECTED
1650	002032	044717	E5004E	
1651	002034	041566	DT0045	
1652	002036	042274	DF0045	
1653				
1654			;	ERROR 52
1655	002040	047552	EM5052	;DATA XFER ABORT TO SOON
1656	002042	043040	DH000A	
1657	002044	041576	DT0046	
1658	002046	042320	DF0046	
1659				
1660			;	ERROR 53
1661	002050	047672	EM5053	;MR1 ERROR IN WRT CHK OR ECC READ AFTER WRT CHK
1662	002052	044642	E50048	
1663	002054	041370	DT0010	
1664	002056	041764	DF0010	
1665				
1666			;	ERROR 54
1667	002060	047761	EM5054	;ECC ERROR IN WRITE CHECK OP
1668	002062	045336	E5021A	
1669	002064	041470	DT0021	
1670	002066	042110	DF0021	
1671				
1672			;	ERROR 55
1673	002070	047672	EM5053	;CS1 ERROR AFTER WRT CHK OR ECC READ IN WRT CHK
1674	002072	044623	E5004A	
1675	002074	041406	DT0011	

M03

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 38
 DZR6EA.P11 28-SEP-76 15:31 ERROR POINTER TABLE

SEQ 0038

1676	002076	042010	DF0011	
1677				
1678			;	ERROR 56
1679	002100	046556	EM5034	;CS1 ERROR AFTER READ DATA OR ECC
1680	002102	044623	E5004A	
1681	002104	041322	DT0004	
1682	002106	041640	DF0004	
1683				
1684			;	ERROR 57
1685	002110	047672	EM5053	;CS1 ERROR AFTER WRITE CHECK OR ECC READ
1686	002112	044623	E5004A	
1687	002114	041322	DT0004	
1688	002116	041640	DF0004	
1689				
1690			;	ERROR 60
1691	002120	050024	EM5060	;WCE FAILED TO SET IN 16 BIT MODE
1692	002122	043040	DH000A	
1693	002124	041524	DT0037	
1694	002126	042364	DF0060	
1695				
1696			;	ERROR 61
1697	002130	050103	EM5061	;TRANSFER LENGTH PROBLEM - RKBA
1698	002132	044735	E5004F	
1699	002134	041576	DT0046	
1700	002136	042404	DF0061	
1701				
1702			;	ERROR 62
1703	002140	050103	EM5061	;TRANSFER LENGTH PROBLEM - RKWC
1704	002142	044753	E5004G	
1705	002144	041606	DT0062	
1706	002146	042430	DF0062	
1707				
1708			;	ERROR 63
1709	002150	050160	EM5063	;WCE FAILED TO SET IN 18 BIT MODE
1710	002152	044753	E5004G	
1711	002154	041616	DT0063	
1712	002156	042454	DF0063	
1713				

1714			.SBTTL	TEMPORARY STORAGE FOR RK611 CONTROLLER REGISTER
1715	002160	000000	T.CS1:	.WORD 0 ;CONTROL AND STATUS REGISTER 1
1716	002162	000000	T.WC:	.WORD 0 ;WORD COUNT REGISTER
1717	002164	000000	T.BA:	.WORD 0 ;BUS ADDRESS REGISTER
1718	002166	000000	T.DA:	.WORD 0 ;DESIRED TRACK SECTOR REGISTER
1719	002170	000000	T.CS2:	.WORD 0 ;CONTROL AND STATUS REGISTER 2
1720	002172	000000	T.DS:	.WORD 0 ;DRIVE STATUS REGISTER
1721	002174	000000	T.ER:	.WORD 0 ;ERROR REGISTER
1722	002176	000000	T.ASOF:	.WORD 0 ;ATTENTION SUMMARY AND OFFSET REGISTER
1723	002200	000000	T.DCYL:	.WORD 0 ;DESIRED CYLINDER REGISTER
1724	002202	000000	T.DB:	.WORD 0 ;DATA BUFFER
1725	002204	000000	T.MR1:	.WORD 0 ;MAINTENANCE REGISTER 1
1726	002206	000000	T.MR2:	.WORD 0 ;MAINTENANCE REGISTER 2
1727	002210	000000	T.MR3:	.WORD 0 ;MAINTENANCE REGISTER 3
1728	002212	000000	T.ECPS:	.WORD 0 ;ECC POSITION INFORMATION
1729	002214	000000	T.ECPT:	.WORD 0 ;ECC PATTERN INFORMATION
1730	002216	000000	T.SPARE:	.WORD 0 ;SPARE REGISTER
1731				

N03

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 39
DZR6EA.P11 28-SEP-76 15:31 EXPECTED RK611 CONTROLLER REGISTERS

SEG 0039

1732			.SBTTL	EXPECTED RK611 CONTROLLER REGISTERS	
1733					
1734	002220	000000	E.CS1:	.WORD 0	; CONTROL AND STATUS REGISTER 1
1735	002222	000000	E.WC:	.WORD 0	; WORD COUNT REGISTER
1736	002224	000000	E.BA:	.WORD 0	; BUS ADDRESS REGISTER
1737	002226	000000	E.DA:	.WORD 0	; DESIRED TRACK SECTOR REGISTER
1738	002230	000000	E.CS2:	.WORD 0	; CONTROL AND STATUS REGISTER 2
1739	002232	000000	E.DS:	.WORD 0	; DRIVE STATUS REGISTER
1740	002234	000000	E.ER:	.WORD 0	; ERROR REGISTER
1741	002236	000000	E.ASOF:	.WORD 0	; ATTENTION SUMMARY AND OFFSET REGISTER
1742	002240	000000	E.DCYL:	.WORD 0	; DESIRED CYLINDER REGISTER
1743	002242	000000	E.DB:	.WORD 0	; DATA BUFFER
1744	002244	000000	E.MR1:	.WORD 0	; MAINTENANCE REGISTER 1
1745	002246	000000	E.MR2:	.WORD 0	; MAINTENANCE REGISTER 2
1746	002250	000000	E.MR3:	.WORD 0	; MAINTENANCE REGISTER 3
1747	002252	000000	E.ECPS:	.WORD 0	; ECC POSITION INFORMATION
1748	002254	000000	E.ECPT:	.WORD 0	; ECC PATTERN INFORMATION
1749	002256	000000	E.SPAR:	.WORD 0	; SPARE REGISTER
1750					
1751			.SBTTL	"L" REGISTERS FOR COMMAND START	
1752	002260	000000	L.CS1:	.WORD 0	; CONTROL AND STATUS 1
1753	002262	000000	L.WC:	.WORD 0	; WORD COUNT
1754	002264	000000	L.BA:	.WORD 0	; BUFR ADDRESS
1755	002266		L.DA:		; DESIRED
1756	002266	000	L.DS:	.BYTE 0	; SECTOR
1757	002267	000	L.DT:	.BYTE 0	; TRACK
1758	002270	000000	L.CS2:	.WORD 0	; CONTROL AND STATUS
1759	002272	000000	L.ASOF:	.WORD 0	; ATTENTION AND OFFSET
1760	002274	000000	L.DCYL:	.WORD 0	; DESIRED CYLINDER
1761	002276	000000	L.MR1:	.WORD 0	; MAINT. REG 1
1762	002300	000000	L.MR2:	.WORD 0	; MAINT. REG 2
1763	002302	000000	L.MR3:	.WORD 0	; MAINT. REG 3
1764					
1765					
1766					
1767			.SBTTL	PROGRAM DEFINED VARIABLES	
1768					
1769	002304	000210	RKVEC:	.WORD 210	; RK611 VECTOR
1770	002306	000240	RKPRI:	.WORD PRS	; RK611 PRIORITY
1771	002310	000000	TRAPPC:	.WORD 0	; PC FOR MEMORY CHECK ENABLE TRAP
1772	002312	000000	SRTFLG:	.WORD 0	; START FLAG
1773					0 = 200
1774					1 = 214
1775					-1 = 204
1776	002314	000000	ERRCNT:	.WORD 0	; ERROR COUNT FOR SWITCH 12 ABORT
1777	002316	000000	P1.BIT:	.WORD 0	; NEXT BIT IN DATA SIMULATION
1778	002320	000000	PR.BIT:	.WORD 0	; PRESENT BIT IN DATA SIMULATION
1779	002322	000000	M1.BIT:	.WORD 0	; PREVIOUS BIT IN DATA SIMULATION
1780	002324	000000	M2.BIT:	.WORD 0	; BIT BEFORE PREVIOUS BIT
1781	002326	000000	BITCNT:	.WORD 0	; BIT POSITION
1782	002330	000000	WRDCNT:	.WORD 0	; WORD COUNT FOR NPR TRANSFER
1783	002332	000000	MEMPAR:	.WORD 0	; MEMORY EMABLE ON FIRST 24K
1784	002334	000000	BADPAR:	.WORD 0	; BAD PARITY FLAG
1785	002336	000000	ECCXOR:	.WORD 0	; ECC XOR SWITCH
1786	002340	000000	ECCHI:	.WORD 0	; ECC HIGH STORAGE
1787	002342	000000	ECCL0:	.WORD 0	; ECC LOW STORAGE

B04

RAB1: DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZ6E-A MAC111 27(1006) 05-OCT-76 09:12 PAGE 40
DZ6E.A.F11 28-SEP-76 15:31 PROGRAM DEFINED VARIABLES

SEQ 0040

1788	002344	000000	ECCSRC: .WORD	0	:ECC SOURCE FLAG
1789	002346	000000	ECPATX: .WORD	0	:EXPECTED PATTERN ON ERROR
1790	002350	000000	ECPOSX: .WORD	0	:EXPECTED POSITION ON ERROR
1791	002352	000000	BSEDES: .WORD	0	:BAD SECTOR DESIRED
1792	002354	000000	HI.BITS: .WORD	0	:HI ORDER BITS FOR 18 BIT WORD XFER
1793	002356	000000	SAVSWR: .WORD	0	:STORAGE FOR SWITCH REGISTER

C04

 RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZ6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 41
 DZ6EA.P11 28-SEP-76 15:31 PROGRAM SETUP

SEG 0041

```

1794      .SBTTL  PROGRAM SETUP
1795
1796 002360 012737 000001 002312 PARM:  MOV  #1,SRTFLG      ;LOAD START FLAG FOR PARMETER START
1797 002366 000406                      BR      START1
1798
1799 002370 012737 177777 002312 RESTR: MOV  #-1,SRTFLG    ;LOAD START FLAG FOR RESTART
1800 002376 000402                      BR      START1
1801
1802 002400 005037 002312          START: CLR  SRTFLG        ;CLEAR START FLAG
1803 002404 000005          START1: RESET                     ;RESET THE WHOLE SYSTEM
1804 002406 105037 001103          CLR  SERFLG             ;CLEAR ERROR FLAG
1805 002412 012706 001100          MOV  #STACK,SP          ;INITIALIZE STACK POINTER
1806 002416 012746 000340          MOV  #PR7,-(SP)         ;LOAD STACK TO LOCK OUT ALL INTERRUPTS
1807 002422 012746 002430          MOV  #15,-(SP)          ;LOAD START OF PROGRAM
1808 002426 000002          RTI                               ;LOAD PSW
1809
1810 002430 004737 037410          IS:   JSR  PC,#STKINT      ;INIT KEYBOARD
1811      .SBTTL  INITIALIZE THE COMMON TAGS
1812      ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1813 002434 012706 001100          MOV  #CMTAG,R6           ;FIRST LOCATION TO BE CLEARED
1814 002440 005026                      CLR  (R6)+          ;CLEAR MEMORY LOCATION
1815 002442 022706 001140          CMP  #SWR,R6 ;;DONE?
1816 002446 001374                      BNE  -6             ;LOOP BACK IF NO
1817 002450 012706 001100          MOV  #STACK,SP          ;SETUP THE STACK POINTER
1818      ;;INITIALIZE A FEW VECTORS
1819 002454 012737 034756 000020      MOV  #SCOPE,#IOTVEC    ;IOT VECTOR FOR SCOPE ROUTINE
1820 002462 012737 000340 000022      MOV  #340,#IOTVEC+2    ;LEVEL 7
1821 002470 012737 035666 000030      MOV  #ERROR,#EMTVEC    ;EMT VECTOR FOR ERROR ROUTINE
1822 002476 012737 000340 000032      MOV  #340,#EMTVEC+2    ;LEVEL 7
1823 002504 012737 041226 000034      MOV  #TRAP,#TRAPVEC   ;TRAP VECTOR FOR TRAP CALLS
1824 002512 012737 000340 000036      MOV  #340,#TRAPVEC+2  ;LEVEL 7
1825 002520 012737 041072 000024      MOV  #SPWRDN,#PWAVEC   ;POWER FAILURE VECTOR
1826 002526 012737 000340 000026      MOV  #340,#PWAVEC+2  ;LEVEL 7
1827 002534 013737 024706 024700      MOV  #ENDCT,#EOPCT    ;SETUP END-OF-PROGRAM COUNTER
1828 002542 005037 001234          CLR  $TIMES             ;INITIALIZE NUMBER OF ITERATIONS
1829 002546 005037 001236          CLR  $ESCAPE            ;CLEAR THE ESCAPE ON ERROR ADDRESS
1830 002552 112737 000001 001115      MOV  #1,$SERMAX      ;ALLOW ONE ERROR PER TEST
1831 002560 012737 002560 001106      MOV  #,$SLPADR      ;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1832 002566 012737 002566 001110      MOV  #,$SLPERR      ;SETUP THE ERROR LOOP ADDRESS
1833      ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1834      ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1835 002574 013746 000004          MOV  #ERRVEC,-(SP)        ;SAVE ERROR VECTOR
1836 002600 012737 002634 000004      MOV  #64,$ERRVEC      ;SET UP ERROR VECTOR
1837 002606 012737 177570 001140      MOV  #DSWR,SWR        ;SETUP FOR A HARDWARE SWICH REGISTER
1838 002614 012737 177570 001142      MOV  #DDISP,DISPLAY   ;AND A HARDWARE DISPLAY REGISTER
1839 002622 022777 177777 176310      CMP  #-1,$SWR        ;TRY TO REFERENCE HARDWARE SWR
1840 002630 001012                      BNE  66$            ;BRANCH IF NO TIMEOUT TRAP OCCURRED
1841
1842 002632 000403                      BR  65$              ;AND THE HARDWARE SWR IS NOT = -1
1843 002634 012716 002642          64$: MOV  #65$,(SP)      ;BRANCH IF NO TIMEOUT
1844 002640 000002                      RTI                  ;SET UP FOR TRAP RETURN
1845 002642 012737 000176 001140      65$: MOV  #SWREG,SWR    ;POINT TO SOFTWARE SWR
1846 002650 012737 000174 001142      MOV  #DISPREG,DISPLAY ;
1847 002656 012637 000004          66$: MOV  (SP)+,#ERRVEC  ;RESTORE ERROR VECTOR
1848
1849 002662 005037 001256          CLR  $PASS                ;CLEAR PASS COUNT

```

D04

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 42
 DZR6EA.P11 28-SEP-76 15:31 INITIALIZE THE COMMON TAGS

SEG 0042

```

1850 002666 132737 000200 001271      BITB      #APTSIZE,$ENVM      ;;TEST USER SIZE UNDER APT
1851 002674 001403                      BEQ        67$           ;;YES,USE NON-APT SWITCH
1852 002676 012737 001272 001140      MOV        #SSWREG,SWR      ;;NO,USE APT SWITCH REGISTER
1853 002704                      67$:
1854 002704 005037 002314      CLR        ERRCNT           ;CLEAR ERROR COUNT FOR SWITCH 12 ABCRT
1855                      .SBTTL      TYPE PROGRAM NAME
1856                      ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
1857 002710 005227 177777      INC        #-1           ;;FIRST TIME?
1858 002714 001063                      BNE        68$           ;;BRANCH IF NO
1859 002716 022737 025042 000042      CMP        #SENDAD,2#42      ;;ACT-11?
1860 002724 001457                      BEQ        68$           ;;BRANCH IF YES
1861 002726 104401 002774      TYPE        69$           ;;TYPE ASCIZ STRING
1862                      .SBTTL      GET VALUE FOR SOFTWARE SWITCH REGISTER
1863 002732 005737 000042      TST        2#42           ;;ARE WE RUNNING UNDER XXDP/ACT?
1864 002736 001012                      BNE        70$           ;;BRANCH IF YES
1865 002740 123727 001270 000001      CMPB       $ENV, #1       ;;ARE WE RUNNING UNDER APT?
1866 002746 001406                      BEQ        70$           ;;BRANCH IF YES
1867 002750 023727 001140 000176      CMP        SWR, #SWREG      ;;SOFTWARE SWITCH REG SELECTED?
1868 002756 001005                      BNE        71$           ;;BRANCH IF NO
1869 002760 104406                      GTSWR           ;;GET SOFT-SWR SETTINGS
1870 002762 000403                      BR        71$
1871 002764 112737 000001 001134      MOVB       #1,$AUTOB      ;;SET AUTO-MODE INDICATOR
1872 002772                      71$:
1873 002772 000434                      BR        68$           ;;GET OVER THE ASCIZ
1874                      ;;69$: .ASCIZ <CRLF>/RK611 DISKLESS DIAGNOSTIC: PART 5 MAINDEC-11-DZR6EA/<CRLF>
1875 003064                      68$:
1876 003064 005227 177777      INC        #-1           ;TEST IF PASS 0
1877 003070 001002                      BNE        4$           ;NO - SKIP
1878 003072 104401 042612      TYPE        ,OPR006      ;TYPE PASS TIME MESSAGE
1879 003076 022737 000001 002312 4$:  CMP        #1,SRTFLG      ;CHECK IF PARAMETER START
1880 003104 001121                      BNE        15$          ;NO, CONTINUE SETUP
1881 003106 104401 042500      TYPE        ,OPR001      ;TYPE "RK611 BUS ADDRESS ( ) ="
1882 003112 013746 001324      MOV        $BASE,-(SP)    ;SAVE $BASE FOR TYPEOUT
1883 003116 104402                      TYPE        ,OPR002      ;GO TYPE--OCTAL ASCII(ALL DIGITS)
1884 003120 104401 042532      TYPE
1885 003124 104412                      RDOCT
1886 003126 012637 001202      MOV        (SP)+,$TMP0    ;GET VALUE
1887 003132 001407                      BEQ        7$           ;CHECK IF <CR>
1888 003134 022737 160000 001202      CMP        #160000,$TMP0  ;CHECK IF IN I/O PAGE
1889 003142 101361                      BHI        5$
1890 003144 013737 001202 001324 7$:  MOV        $TMP0,$BASE    ;LOAD NEW BUS ADDRESS
1891 003152 104401 042540      TYPE        ,OPR003      ;TYPE "RK611 VECTOR ADDRESS ( ) ="
1892 003156 013746 001320      MOV        $VECT1,-(SP)  ;TYPE OUT VECTOR ADDRESS
1893 003162 042716 160000      BIC        #160000,(SP)
1894 003166 104401 042532      TYPE        ,OPR002
1895 003172 104412                      RDOCT
1896 003174 012637 001202      MOV        (SP)+,$TMP0    ;GET VALUE
1897 003200 001412                      BEQ        10$          ;CHECK IF <CR>
1898 003202 022737 001000 001202      CMP        #1000,$TMP0    ;CHECK IF LEGAL
1899 003210 101760                      BLOS       7$
1900 003212 042737 017777 001320      BIC        #17777,$VECT1  ;LOAD NEW VECTOR ADDRESS
1901 003220 053737 001202 001320      BIS        $TMP0,$VECT1
1902 003226 104401 042570 10$:  TYPE        ,OPR004      ;TYPE "RK611 PRIORITY ( ) ="
1903 003232 005046                      CLR        -(SP)
1904 003234 113716 001321      MOVB       $VECT1+1,(SP)
1905 003240 006216                      ASR        (SP)           ;SHIFT 5 BITS RIGHT

```

E04

RK611 DISA-LESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 43
 DZR6EA.P11 28-SEP-76 15:31 GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0042

1906	003242	006216		ASR	(SP)	
1907	003244	006216		ASR	(SP)	
1908	003246	006216		ASR	(SP)	
1909	003250	006216		ASR	(SP)	
1910	003252	104402		TYP0C		
1911	003254	104401	042532	TYPE	,OPR002	
1912	003260	104412		RDOCT		:GET VALUE
1913	003262	012637	001202	MOV	(SP)+,STMP0	
1914	003266	001430		BEQ	15\$	:CHECK FOR DEFAULT
1915	003270	022737	000007 001202	CMP	#7,STMP0	:CHECK IF LEGAL
1916	003276	103753		BLO	10\$	
1917	003300	022737	000004 001202	CMP	#4,STMP0	
1918	003306	101347		BHI	10\$	
1919	003310	006337	001202	ASL	STMP0	:SHIFT 5 BITS LEFT
1920	003314	006337	001202	ASL	STMP0	
1921	003320	006337	001202	ASL	STMP0	
1922	003324	006337	001202	ASL	STMP0	
1923	003330	006337	001202	ASL	STMP0	
1924	003334	042737	160000 001320	BIC	#160000,\$VECT1	:STORE NEW PRIORITY
1925	003342	153737	001202 001321	BISB	STMP0,\$VECT1+1	
1926	003350	013737	001320 002304	MOV	\$VECT1,RKVEC	:STORE RK611 VECTOR
1927	003356	042737	160000 002304	BIC	#160000,RKVEC	
1928	003364	113737	001321 002306	MOVB	\$VECT1+1,RKPRI	:STORE PRIORITY
1929	003372	004737	034450	JSR	PC,\$SIZE	:SIZE MEMORY
1930						
1931	003376	013702	001324	NEWPAS: MOV	\$BASE,R2	:SET RK POINTER
1932	003402	005037	001236	CLR	\$ESCAPE	:CLEAR ESCAPE
1933	003406	004737	034314	JSR	PC,PARCHK	:CHECK OF MEMORY CHECK ENABLE
1934	003412	012746	000000	MOV	#PR0,-(SP)	:LOCK OUT INTERRUPTS
1935	003416	012746	003424	MOV	#TST1,-(SP)	
1936	003422	000002		RTI		

F04

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZRG6-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 44
 DZRG6A.P11 28-SEP-76 15:31 **MULTI-SECTOR WRITE OPERATIONS

SEG CC44

.SBTTL **MULTI-SECTOR WRITE OPERATIONS

 TEST 1 MULTI-SECTOR WRITE (PART 1)

CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.
 PUT THE CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
 DATA OF 401 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
 CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
 AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE
 AND A GOOD HEADER. CLOCK THROUGH 400 WORDS AND THE
 TWO ECC WORDS. MAKE SURE THE NEXT HEADER RECOGNITION
 CYCLE STARTS AND THE PROPER HEADER IS SEARCHED FOR.
 PROVIDE A SECOND GOOD HEADER AND CHECK RECOGNITION
 OF HEADER.

 TEST1: SCOPE
 MOV #10., \$TIMES ;:DO 10. ITERATIONS
 MOV #0CLR, RKCS1(R2) ;CLR CONTROLLER
 CLR BSEDES ;CLEAR BAD SEC DESIRED
 JSR R4, LOADRK ;LOAD "L" REGISTERS
 0 ;CYLINDER 0
 .BYTE 0 ;SECTOR 0
 .BYTE 0 ;TRACK 0
 OBUFF ;BUFFER ADDRESS OBUFF
 -401 ;WORD COUNT -401
 WRDATA ;COMMAND WRDATA
 JSR R4, BLDDAT ;GO BUILD DATA
 1 ;PATTERN 1
 JSR R4, OPSTRT ;START THE OPERATION
 JSR R4, DISEEK ;SIMULATE IMPLIED SEEK
 ERROR 4 ;CS1 MISCOMPARE
 ERROR 5 ;MR1
 ERROR 6 ;MR2
 ERROR 7 ;MR3
 JSR R4, DSECTR ;SIMULATE SECTOR PULSE
 JSR R4, DRSYNC ;SIMULATE HEADER PREAMBLE
 ERROR 10 ;MR1 CONTENTS IN ERROR
 ERROR 11 ;CS1 CONTENTS IN ERROR
 JSR R4, DHDCMP ;SIMULATE HEADER SEARCH
 ERROR 12 ;MR1 CONTENTS IN ERROR
 ERROR 13 ;CS1 IN ERROR AFTER SEARCH
 ERROR 14 ;RKDCYL OR RKDA IN ERROR AFTER SEARCH
 MOV #EM5020, EMW ;SET MESSAGE FOR WRITE FAILURE
 JSR R4, DWGPSN ;SIMULATE GAP AND SYNC FOR WRITE

1937
 1938
 1939
 1940
 1941
 1942
 1943
 1944
 1945
 1946
 1947
 1948
 1949
 1950
 1951
 1952
 1953
 1954 003424 000004
 1955 003426 012737 000012 001234
 1956
 1957 003434 012762 100000 000000
 1958
 1959 003442 005037 002352
 1960
 1961 003446 004437 034026
 1962 003452 000000
 1963 003454 000
 1964 003455 000
 1965 003456 051444
 1966 003460 177377
 1967 003462 000023
 1968
 1969 003464 004437 033612
 1970 003470 000001
 1971
 1972 003472 004437 034072
 1973
 1974 003476 004437 025100
 1975 003502 104004
 1976 003504 104005
 1977 003506 104006
 1978 003510 104007
 1979
 1980 003512 004437 025062
 1981
 1982 003516 004437 025504
 1983 003522 104010
 1984 003524 104011
 1985
 1986 003526 004437 026042
 1987 003532 104012
 1988 003534 104013
 1989 003536 104014
 1990
 1991 003540 012737 045211 001510
 1992 003546 004437 027002

G04

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 45
 DZR6EA.P11 28-SEP-76 15:31 T1 MULTI-SECTOR WRITE (PART 1)

SEG 0045

1993	003552	104015			ERROR	15		;MR1 IN ERROR IN READ GAP
1994	003554	104016			ERROR	16		;CS1 IN ERROR AFTER READ GAP
1995	003556	104017			ERROR	17		;MR1 IN ERROR IN WRITING SYNC
1996	003560	104020			ERROR	20		;CS1 IN ERROR AFTER WRITING SYNC
1997								
1998	003562	012737	045270	001510	MOV	#EM5021,EMW		;SET MESSAGE FOR WRITE FAILURE
1999	003570	004437	027402		JSR	R4,DWRITE		;SIMULATE WRITE DATA AND ECC
2000	003574	104017			ERROR	17		;MR1 IN ERROR IN WRITING DATA OR ECC
2001	003576	104021			ERROR	21		;ECC IN ERROR WHILE WRITING DATA
2002	003600	104022			ERROR	22		;CS1 IN ERROR AFTER WRITING DATA AND ECC
2003								
2004	003602	012700	000034		MOV	#7*4,R0		;ISSUE CLOCKS TO NEXT HDR COMPARE
2005	003606	012762	000440	000026	15:	MOV	#DMD!MCLK,RKMR1(R2)	;CLOCK LOOP
2006	003614	012762	000040	000026	MOV	#DMD,RKMR1(R2)		
2007	003622	005300			DEC	R0		
2008	003624	001370			BNE	15		
2009								
2010	003626	013703	002274		MOV	L.DCYL,R3		;GET DESIRED CYLINDER
2011	003632	004437	034242		JSR	R4,FSBLVV		;INVERT WORD
2012	003636	010337	002246		MOV	R3,E.MR2		;STORE EXPECTED MR2
2013	003642	113703	002267		MOV	L.DT,R3		;GET DESIRED TRACK/SECTOR
2014	003646	042703	177400		BIC	#177400,R3		;CLEAR UNUSED BITS
2015	003652	012700	000005		MOV	#5,R0		;SET COUNT FOR SHIFTING
2016	003656	006303			ASL	R3		;SHIFT TRACK FOR HEADER ALIGNMENT
2017	003660	005300			DEC	R0		;DEC COUNT
2018	003662	001375			BNE	55		;LOOP UNTIL ZERO
2019	003664	153703	002266		BISB	L.DS,R3		;INSERT SECTOR NUMBER
2020								
2021	003670	032737	010000	002260	BIT	#CFMT,L.CS1		;TEST IF 18 BIT MODE
2022	003676	001402			BEQ	65		;NO - SKIP
2023	003700	052703	001000		BIS	#BIT9,R3		;SET FORMAT BIT FOR WORD TWO
2024	003704				65:			
2025	003704	004437	034242		JSR	R4,FSBLVV		;INVERT WORD
2026	003710	010337	002250		MOV	R3,E.MR3		;STORE EXPECTED MR3
2027	003714	016237	000034	002206	MOV	RKMR2(R2),T.MR2		;GET MR2
2028	003722	016237	000036	002210	MOV	RKMR3(R2),T.MR3		;GET MR3
2029	003730	016237	000000	002160	MOV	RKCS1(R2),T.CS1		;GET CS1
2030	003736	013737	002260	002220	MOV	L.CS1,E.CS1		;GET EXPECTED CS1
2031	003744	023737	002160	002220	CMP	T.CS1,E.CS1		;CHECK IF OK
2032	003752	001402			BEQ	35		;YES - SKIP
2033	003754	104025			ERROR	25		;CS1 IN ERROR AT MULTI-SECTOR TIME
2034	003756	000427			BR	TST2		;GO TO NEXT TEST
2035								
2036	003760	023737	002206	002246	35:	CMP	T.MR2,E.MR2	;CHECK IF MR2 OK
2037	003766	001402			BEQ	25		;YES - SKIP
2038	003770	104023			ERROR	23		;MR2 IN ERROR AT MULTI-SECTOR TIME
2039	003772	000421			BR	TST2		;GO TO NEXT TEST
2040								
2041	003774	023737	002210	002250	25:	CMP	T.MR3,E.MR3	;CHECK IF MR3 OK
2042	004002	001402			BEQ	45		;YES - SKIP
2043	004004	104024			ERROR	24		;MR3 IN ERROR AT MULTI-SECTOR TIME
2044	004006	000413			BR	TST2		;GO TO NEXT TEST
2045								
2046	004010	004437	025062		45:	JSR	R4,DSECTR	;SIMULATE 2ND SECTOR PULSE
2047								
2048	004014	004437	025504		JSR	R4,DRSYNC		;SIMULATE PREAMBLE

H04

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 46
 DZR6EA.P11 28-SEP-76 15:31 T1 MULTI-SECTOR WRITE (PART 1)

SEQ 0046

2049	004020	104010		ERROR	10		;MR1 ERROR
2050	004022	104011		ERROR	11		;CS1 ERROR
2051							
2052	004024	004437	026042	JSR	R4,DHDCMP		;SIMULATE HEADER SEARCH
2053	004030	104012		ERROR	12		;MR1 ERROR
2054	004032	104013		ERROR	13		;CS1 ERROR
2055	004034	104014		ERROR	14		;RKDCYL OR RKDA IN ERROR
2056							

 *TEST 2 MULTI-SECTOR WRITE (PART 2)

*
 * CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.
 * PUT THE CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
 * DATA OF 401 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
 * CYLINDER 0, HEAD 0, SECTOR 23. CLOCK THROUGH SEEK
 * AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE
 * AND A GOOD HEADER, CLOCK THROUGH 400 WORDS AND THE
 * TWO ECC WORDS. MAKE SURE THE NEXT HEADER RECOGNITION
 * CYCLE STARTS AND THE PROPER HEADER IS SEARCHED FOR.
 * PROVIDE A SECOND GOOD HEADER AND CHECK RECOGNITION
 * OF HEADER.

2071							
2072	004036	000004		TEST2: SCOPE			
2073	004040	012737	000012 001234	MOV	#10.,\$TIMES		;DO 10. ITERATIONS
2074							
2075	004046	012762	100000 000000	MOV	#CCLR,RKCS1(R2)		;CLEAR CONTROLLER
2076							
2077	004054	005037	002352	CLR	BSEDES		;CLEAR BAD SEC DESIRED
2078							
2079	004060	004437	034026	JSR	R4,LOADRK		;LOAD "L" REGISTERS
2080	004064	000000		0			;CYLINDER 0
2081	004066	023		.BYTE	23		;SECTOR 23
2082	004067	000		.BYTE	0		;TRACK 0
2083	004070	051444		0BUFF			;BUFFER ADDRESS 0BUFF
2084	004072	177377		-401			;WORD COUNT -401
2085	004074	000023		WRDATA			;COMMAND WRDATA
2086							
2087	004076	004437	033612	JSR	R4,BLDDAT		;GO BUILD DATA
2088	004102	000002		2			;PATTERN 2
2089							
2090	004104	004437	034072	JSR	R4,OPSTRT		;START THE OPERATION
2091							
2092	004110	004437	025100	JSR	R4,DISEEK		;SIMULATE IMPLIED SEEK
2093	004114	104004		ERROR	4		;CS1 MISCOMPARE
2094	004116	104005		ERROR	5		;MR1 "
2095	004120	104006		ERROR	6		;MR2 "
2096	004122	104007		ERROR	7		;MR3 "
2097							
2098	004124	004437	025062	JSR	R4,DSECTR		;SIMULATE SECTOR PULSE
2099							
2100	004130	004437	025504	JSR	R4,DRSYNC		;SIMULATE HEADER PREAMBLE
2101	004134	104010		ERROR	10		;MR1 CONTENTS IN ERROR
2102	004136	104011		ERROR	11		;CS1 CONTENTS IN ERROR
2103							
2104	004140	004437	026042	JSR	R4,DHDCMP		;SIMULATE HEADER SEARCH

2105	004144	104012			EPROR	12		;MRI CONTENTS IN ERROR
2106	004146	104013			ERROR	13		;CSI IN ERROR AFTER SEARCH
2107	004150	104014			ERROR	14		;RKDCYL OR RKDA IN ERROR AFTER SEARCH
2108								
2109	004152	012737	045211	001510	MOV	#EM5020,EMW		;SET MESSAGE FOR WRITE FAILURE
2110	004160	004437	027002		JSR	R4,DWGPSN		;SIMULATE GAP AND SYNC FOR WRITE
2111	004164	104015			ERROR	15		;MRI IN ERROR IN READ GAP
2112	004166	104016			ERROR	16		;CSI IN ERROR AFTER READ GAP
2113	004170	104017			ERROR	17		;MRI IN ERROR IN WRITING SYNC
2114	004172	104020			ERROR	20		;CSI IN ERROR AFTER WRITING SYNC
2115								
2116	004174	012737	045270	001510	MOV	#EM5021,EMW		;SET MESSAGE FOR WRITE FAILURE
2117	004202	004437	027402		JSR	R4,DWRITE		;SIMULATE WRITE DATA AND ECC
2118	004206	104017			ERROR	17		;MRI IN ERROR IN WRITING DATA OR ECC
2119	004210	104021			ERROR	21		;ECC IN ERROR WHILE WRITING DATA
2120	004212	104022			ERROR	22		;CSI IN ERROR AFTER WRITING DATA AND ECC
2121								
2122	004214	012700	000034		MOV	#7*4,R0		;ISSUE CLOCKS TO NEXT HDR COMPARE
2123	004220	012762	000440	000026	MOV	#DMD!MCLK,RKMR1(R2)		;CLOCK LOOP
2124	004226	012762	000040	000026	MOV	#DMD,RKMR1(R2)		
2125	004234	005300			DEC	R0		
2126	004236	001370			BNE	1\$		
2127								
2128	004240	013703	002274		MOV	L.DCYL,R3		;GET DESIRED CYLINDER
2129	004244	004437	034242		JSR	R4,FSBLVV		;INVERT WORD
2130	004250	010337	002246		MOV	R3,E.MR2		;STORE EXPECTED MR2
2131	004254	113703	002267		MOVB	L.DT,R3		;GET DESIRED TRACK/SECTOR
2132	004260	042703	177400		BIC	#177400,R3		;CLEAR UNUSED BITS
2133	004264	012700	000005		MOV	#5,R0		;SET COUNT FOR SHIFTING
2134	004270	006303			ASL	R3		;SHIFT TRACK FOR HEADER ALIGNMENT
2135	004272	005300			DEC	R0		;DEC COUNT
2136	004274	001375			BNE	5\$		;LOOP UNTIL ZERO
2137	004276	153703	002266		BISB	L.DS,R3		;INSERT SECTOR NUMBER
2138								
2139	004302	032737	010000	002260	BIT	#CFMT,L.CS1		;TEST IF 18 BIT MODE
2140	004310	001402			BEQ	6\$		;NO - SKIP
2141	004312	052703	001000		BIS	#BIT9,R3		;SET FORMAT BIT FOR WORD TWO
2142	004316							
2143	004316	004437	034242		JSR	R4,FSBLVV		;INVERT WORD
2144	004322	010337	002250		MOV	R3,E.MR3		;STORE EXPECTED MR3
2145	004326	016237	000034	002206	MOV	RKMR2(R2),T.MR2		;GET MR2
2146	004334	016237	000036	002210	MOV	RKMR3(R2),T.MR3		;GET MR3
2147	004342	016237	000000	002160	MOV	RKCS1(R2),T.CS1		;GET CS1
2148	004350	013737	002260	002220	MOV	L.CS1,E.CS1		;GET EXPECTED CS1
2149	004356	023737	002160	002220	CMP	T.CS1,E.CS1		;CHECK IF OK
2150	004364	001402			BEQ	3\$		;YES - SKIP
2151	004366	104025			ERROR	25		;CSI IN ERROR AT MULTI-SECTOR TIME
2152	004370	000427			BR	TST3		;GO TO NEXT TEST
2153								
2154	004372	023737	002206	002246	CMP	T.MR2,E.MR2		;CHECK IF MR2 OK
2155	004400	001402			BEQ	2\$		;YES - SKIP
2156	004402	104023			ERROR	23		;MR2 IN ERROR AT MULTI-SECTOR TIME
2157	004404	000421			BR	TST3		;GO TO NEXT TEST
2158								
2159	004406	023737	002210	002250	CMP	T.MR3,E.MR3		;CHECK IF MR3 OK
2160	004414	001402			BEQ	4\$		;YES - SKIP

J04

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 48
 DZR6E.A.P11 28-SEP-76 15:31 T2 MULTI-SECTOR WRITE (PART 2)

SEG 0048

2161	004416	104024		ERROR	24		;MR3 IN ERROR AT MULTI-SECTOR TIME
2162	004420	000413		BR	TST3		;GO TO NEXT TEST
2163							
2164	004422	004437	025062	4S:	JSR	R4,DSECTR	;SIMULATE 2ND SECTOR PULSE
2165							
2166	004426	004437	025504		JSR	R4,DRSYNC	;SIMULATE PREAMBLE
2167	004432	104010		ERROR	10		;MR1 ERROR
2168	004434	104011		ERROR	11		;CS1 ERROR
2169							
2170	004436	004437	026042		JSR	R4,DHDCMP	;SIMULATE HEADER SEARCH
2171	004442	104012		ERROR	12		;MR1 ERROR
2172	004444	104013		ERROR	13		;CS1 ERROR
2173	004446	104014		ERROR	14		;RKDCYL OR RKDA IN ERROR
2174							
2175				*****			
2176				*TEST 3 MID-TRANSFER SEEK (PART 1)			
2177				*****			
2178				* CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.			
2179				* PUT THE CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE			
2180				* DATA OF 401 WORDS TO AN RK06 IN 26 SECTOR FORMAT.			
2181				* CYLINDER 0, HEAD 0, SECTOR 25. CLOCK THROUGH SEEK			
2182				* AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE			
2183				* AND A GOOD HEADER. CLOCK THROUGH 400 WORDS AND THE			
2184				* TWO WORD ECC. MAKE SURE THE CONTROLLER ISSUES			
2185				* AN IMPLIED SEEK TO THE RIGHT HEAD AND CYLINDER.			
2186				* MAKE SURE THE NEXT HEADER RECOGNITION PROCEEDS PROPERLY.			
2187				*****			
2188				*****			
2189	004450	000004		TST3:	SCOPE		
2190	004452	012737	000012 001234		MOV	#10.,\$TIMES	;DO 10. ITERATIONS
2191							
2192	004460	012762	100000 000000		MOV	#CCLR,RKCS1(R2)	;CLEAR CONTROLLER
2193							
2194	004466	005037	002352		CLR	BSEDES	;CLEAR BAD SEC DESIRED
2195							
2196	004472	004437	034026		JSR	R4,LOADRK	;LOAD "L" REGISTERS
2197	004476	000000			0		;CYLINDER 0
2198	004500	025			.BYTE	25	;SECTOR 25
2199	004501	000			.BYTE	0	;TRACK 0
2200	004502	051444			OBUFF		;BUFFER ADDRESS OBUFF
2201	004504	177377			-401		;WORD COUNT -401
2202	004506	000023			WRDATA		;COMMAND WRDATA
2203							
2204	004510	004437	033612		JSR	R4,BLDDAT	;GO BUILD DATA
2205	004514	000003			3		;PATTERN 3
2206							
2207	004516	004437	034072		JSR	R4,OPSTRT	;START THE OPERATION
2208							
2209	004522	004437	025100		JSR	R4,DISEEK	;SIMULATE IMPLIED SEEK
2210	004526	104004		ERROR	4		;CS1 MISCOMPARE
2211	004530	104005		ERROR	5		;MR1 "
2212	004532	104006		ERROR	6		;MR2 "
2213	004534	104007		ERROR	7		;MR3 "
2214							
2215	004536	004437	025062		JSR	R4,DSECTR	;SIMULATE SECTOR PULSE
2216							

K04

 RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 49
 DZR6EA.P11 28-SEP-76 15:31 T3 MID-TRANSFER SEEK (PART 1)

SEG 0049

2217	004542	004437	025504		JSR	R4,DRSYNC	;SIMULATE HEADER PREAMBLE
2218	004546	104010			ERROR	10	;MR1 CONTENTS IN ERROR
2219	004550	104011			ERROR	11	;CS1 CONTENTS IN ERROR
2220							
2221	004552	004437	026042		JSR	R4,DHDCMP	;SIMULATE HEADER SEARCH
2222	004556	104012			ERROR	12	;MR1 CONTENTS IN ERROR
2223	004560	104013			ERROR	13	;CS1 IN ERROR AFTER SEARCH
2224	004562	104014			ERROR	14	;RKDCYL OR RKDA IN ERROR AFTER SEARCH
2225							
2226	004564	012737	045211	001510	MOV	#EM5020,EMW	;SET MESSAGE FOR WRITE FAILURE
2227	004572	004437	027002		JSR	R4,DWGPSN	;SIMULATE GAP AND SYNC FOR WRITE
2228	004576	104015			ERROR	15	;MR1 IN ERROR IN READ GAP
2229	004600	104016			ERROR	16	;CS1 IN ERROR AFTER READ GAP
2230	004602	104017			ERROR	17	;MR1 IN ERROR IN WRITING SYNC
2231	004604	104020			ERROR	20	;CS1 IN ERROR AFTER WRITING SYNC
2232							
2233	004606	012737	045270	001510	MOV	#EM5021,EMW	;SET MESSAGE FOR WRITE FAILURE
2234	004614	004437	027402		JSR	R4,DWRITE	;SIMULATE WRITE DATA AND ECC
2235	004620	104017			ERROR	17	;MR1 IN ERROR IN WRITING DATA OR ECC
2236	004622	104021			ERROR	21	;ECC IN ERROR WHILE WRITING DATA
2237	004624	104022			ERROR	22	;CS1 IN ERROR AFTER WRITING DATA AND ECC
2238							
2239	004626	012700	000324		MOV	#53.*4,R0	;ISSUE CLOCKS TO NEXT HDR COMPARE
2240	004632	012762	000440	000026	MOV	#DMD!MCLK,RKMR1(R2)	;CLOCK LOOP
2241	004640	012762	000040	000026	MOV	#DMD,RKMR1(R2)	
2242	004646	005300			DEC	R0	
2243	004650	001370			BNE	1\$	
2244							
2245	004652	013703	002274		MOV	L.DCYL,R3	;GET DESIRED CYLINDER
2246	004656	004437	034242		JSR	R4,FSBLVV	;INVERT WORD
2247	004662	010337	002246		MOV	R3,E.MR2	;STORE EXPECTED MR2
2248	004666	113703	002267		MOV	L.DT,R3	;GET DESIRED TRACK/SECTOR
2249	004672	042703	177400		BIC	#177400,R3	;CLEAR UNUSED BITS
2250	004676	012700	000005		MOV	#5,R0	;SET COUNT FOR SHIFTING
2251	004702	006303		5\$:	ASL	R3	;SHIFT TRACK FOR HEADER ALIGNMENT
2252	004704	005300			DEC	R0	;DEC COUNT
2253	004706	001375			BNE	5\$	;LOOP UNTIL ZERO
2254	004710	153703	002266		BISB	L.DS,R3	;INSERT SECTOR NUMBER
2255							
2256	004714	032737	010000	002260	BIT	#CFMT,L.CS1	;TEST IF 18 BIT MODE
2257	004722	001402			BEQ	6\$	;NO - SKIP
2258	004724	052703	001000		BIS	#BIT9,R3	;SET FORMAT BIT FOR WORD TWO
2259	004730			6\$:			
2260	004730	004437	034242		JSR	R4,FSBLVV	;INVERT WORD
2261	004734	010337	002250		MOV	R3,E.MR3	;STORE EXPECTED MR3
2262	004740	016237	000034	002206	MOV	RKMR2(R2),T.MR2	;GET MR2
2263	004746	016237	000036	002210	MOV	RKMR3(R2),T.MR3	;GET MR3
2264	004754	016237	000000	002160	MOV	RKCS1(R2),T.CS1	;GET CS1
2265	004762	013737	002260	002220	MOV	L.CS1,E.CS1	;GET EXPECTED CS1
2266	004770	023737	002160	002220	CMP	T.CS1,E.CS1	;CHECK IF OK
2267	004776	001402			BEQ	3\$	;YES - SKIP
2268	005000	104025			ERROR	25	;CS1 IN ERROR AT MULTI-SECTOR TIME
2269	005002	000427			BR	TST4	;GO TO NEXT TEST
2270							
2271	005004	023737	002206	002246	3\$:	CMP	T.MR2,E.MR2
2272	005012	001402			BEQ	2\$	;YES - SKIP

L04

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 50
DZR6EA.P11 28-SEP-76 15:31 T3 MID-TRANSFER SEEK (PART 1)

SEQ 0050

```

2273 005014 104023          ERROR 23          ;MR2 IN ERROR AT MULTI-SECTOR TIME
2274 005016 000421          BR      TST4          ;;GO TO NEXT TEST
2275
2276 005020 023737 002210 002250 2$: CMP      T.MR3,E.MR3      ;CHECK IF MR3 OK
2277 005026 001402          BEQ      4$          ;YES - SKIP
2278 005030 104024          ERROR 24          ;MR3 IN ERROR AT MULTI-SECTOR TIME
2279 005032 000413          BR      TST4          ;;GO TO NEXT TEST
2280
2281 005034 004437 025062          4$: JSR      R4,DSECTR      ;SIMULATE 2ND SECTOR PULSE
2282
2283 005040 004437 025504          JSR      R4,DRSYNC      ;SIMULATE PREAMBLE
2284 005044 104010          ERROR 10          ;MR1 ERROR
2285 005046 104011          ERROR 11          ;CS1 ERROR
2286
2287 005050 004437 026042          JSR      R4,DHDCMP      ;SIMULATE HEADER SEARCH
2288 005054 104012          ERROR 12          ;MR1 ERROR
2289 005056 104013          ERROR 13          ;CS1 ERROR
2290 005060 104014          ERROR 14          ;RKDCYL OR RKDA IN ERROR
2291
2292          ;*****
2293          ;*TEST 4          MID-TRANSFER SEEK (PART 2)
2294          ;*
2295          ;*
2296          ;* CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.
2297          ;* PUT THE CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A WRITE
2298          ;* DATA OF 401 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
2299          ;* CYLINDER 0, HEAD 2, SECTOR 25.  CLOCK THROUGH SEEK
2300          ;* AND DRIVE CLEAR MESSAGES.  SIMULATE A SECTOR PULSE
2301          ;* AND A GOOD HEADER, CLOCK THROUGH 400 WORDS AND THE
2302          ;* TWO WORD ECC.  MAKE SURE THE CONTROLLER ISSUES
2303          ;* AN IMPLIED SEEK TO THE RIGHT HEAD AND CYLINDER.
2304          ;* MAKE SURE THE NEXT HEADER RECOGNITION PROCEEDS PROPERLY.
2305          ;*****
2306          ;*****
2307 005062 000004          TST4: SCOPE
2308 005064 012737 000012 001234      MC'      #10.,$TIMES      ;,DO 10. ITERATIONS
2309
2310 005072 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
2311
2312 005100 005037 002352          CLR      BSEDES          ;CLEAR BAD SEC DESIRED
2313
2314 005104 004437 034026          JSR      R4,LOADRK      ;LOAD "L" REGISTERS
2315 005110 000000          0          ;CYLINDER 0
2316 005112          025          .BYTE 25          ;SECTOR 25
2317 005113          002          .BYTE 2          ;TRACK 2
2318 005114 051444          OBUFF          ;BUFFER ADDRESS OBUFF
2319 005116 177377          -401          ;WORD COUNT -401
2320 005120 000023          WRDATA          ;COMMAND WRDATA
2321
2322 005122 004437 033612          JSR      R4,BLDDAT      ;GO BUILD DATA
2323 005126 000004          4          ;PATTERN 4
2324
2325 005130 004437 034072          JSR      R4,OPSTRT      ;START THE OPERATION
2326
2327 005134 004437 025100          JSR      R4,DISEEK      ;SIMULATE IMPLIED SEEK
2328 005140 104004          ERROR 4          ;CS1 MISCOMPARE
2329 005142 104005          ERROR 5          ;MR1

```

M04

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 51
 DZR6EA.P11 28-SEP-76 15:31 T4 MID-TRANSFER SEEK (PART 2)

SEQ 0051

2329	005144	104006			ERROR	6		;MR2	"
2330	005146	104007			ERROR	7		;MR3	"
2331									
2332	005150	004437	025062		JSR	R4,DSECTR		;SIMULATE SECTOR PULSE	
2333									
2334	005154	004437	025504		JSR	R4,DRSYNC		;SIMULATE HEADER PREAMBLE	
2335	005160	104010			ERROR	10		;MR1 CONTENTS IN ERROR	
2336	005162	104011			ERROR	11		;CS1 CONTENTS IN ERROR	
2337									
2338	005164	004437	026042		JSR	R4,DHDCMP		;SIMULATE HEADER SEARCH	
2339	005170	104012			ERROR	12		;MR1 CONTENTS IN ERROR	
2340	005172	104013			ERROR	13		;CS1 IN ERROR AFTER SEARCH	
2341	005174	104014			ERROR	14		;RKDCYL OR RKDA IN ERROR AFTER SEARCH	
2342									
2343	005176	012737	045211	001510	MOV	#EM5020,EMW		;SET MESSAGE FOR WRITE FAILURE	
2344	005204	004437	027002		JSR	R4,DWGPSN		;SIMULATE GAP AND SYNC FOR WRITE	
2345	005210	104015			ERROR	15		;MR1 IN ERROR IN READ GAP	
2346	005212	104016			ERROR	16		;CS1 IN ERROR AFTER READ GAP	
2347	005214	104017			ERROR	17		;MR1 IN ERROR IN WRITING SYNC	
2348	005216	104020			ERROR	20		;CS1 IN ERROR AFTER WRITING SYNC	
2349									
2350	005220	012737	045270	001510	MOV	#EM5021,EMW		;SET MESSAGE FOR WRITE FAILURE	
2351	005226	004437	027402		JSR	R4,DWRITE		;SIMULATE WRITE DATA AND ECC	
2352	005232	104017			ERROR	17		;MR1 IN ERROR IN WRITING DATA OR ECC	
2353	005234	104021			ERROR	21		;ECC IN ERROR WHILE WRITING DATA	
2354	005236	104022			ERROR	22		;CS1 IN ERROR AFTER WRITING DATA AND ECC	
2355									
2356	005240	012700	000324		MOV	#53,#4,R0		;ISSUE CLOCKS TO NEXT HDR COMPARE	
2357	005244	012762	000440	000026	MOV	#DMD:MCLK,RKMR1(R2)		;CLOCK LOOP	
2358	005252	012762	000040	000026	MOV	#DMD,RKMR1(R2)			
2359	005260	005300			DEC	R0			
2360	005262	001370			BNE	1\$			
2361									
2362	005264	013703	002274		MOV	L,DCYL,R3		;GET DESIRED CYLINDER	
2363	005270	004437	034242		JSR	R4,FSBLVV		;INVERT WORD	
2364	005274	010337	002246		MOV	R3,E.MR2		;STORE EXPECTED MR2	
2365	005300	113703	002267		MOVB	L,DT,R3		;GET DESIRED TRACK/SECTOR	
2366	005304	042703	177400		BIC	#177400,R3		;CLEAR UNUSED BITS	
2367	005310	012700	000005		MOV	#5,R0		;SET COUNT FOR SHIFTING	
2368	005314	006303			ASL	R3		;SHIFT TRACK FOR HEADER ALIGNMENT	
2369	005316	005300			DEC	R0		;DEC COUNT	
2370	005320	001375			BNE	5\$		;LOOP UNTIL ZERO	
2371	005322	153703	002266		BISB	L,DS,R3		;INSERT SECTOR NUMBER	
2372									
2373	005326	032737	010000	002260	BIT	#CFMT,L.CS1		;TEST IF 18 BIT MODE	
2374	005334	001402			BEQ	6\$		;NO - SKIP	
2375	005336	052703	001000		BIS	#BIT9,R3		;SET FORMAT BIT FOR WORD TWO	
2376	005342								
2377	005342	004437	034242		JSR	R4,FSBLVV		;INVERT WORD	
2378	005346	010337	002250		MOV	R3,E.MR3		;STORE EXPECTED MR3	
2379	005352	016237	000034	002206	MOV	RKMR2(R2),T.MR2		;GET MR2	
2380	005360	016237	000036	002210	MOV	RKMR3(R2),T.MR3		;GET MR3	
2381	005366	016237	000000	002160	MOV	RKCS1(R2),T.CS1		;GET CS1	
2382	005374	013737	002260	002220	MOV	L.CS1,E.CS1		;GET EXPECTED CS1	
2383	005402	023737	002160	002220	CMP	T.CS1,E.CS1		;CHECK IF OK	
2384	005410	001402			BEQ	3\$		;YES - SKIP	

N04

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 52
DZR6EA.P11 28-SEP-76 15:31 T4 MID-TRANSFER SEEK (PART 2)

SEQ 0052

2385	005412	104025			ERROR 25	;CS1 IN ERROR AT MULTI-SECTOR TIME
2386	005414	000427			BR TST5	;GO TO NEXT TEST
2387						
2388	005416	023737	002206	002246	3\$: CMP T.MR2,E.MR2	;CHECK IF MR2 OK
2389	005424	001402			BEG 25	;YES - SKIP
2390	005426	104023			ERROR 23	;MR2 IN ERROR AT MULTI-SECTOR TIME
2391	005430	000421			BR TST5	;GO TO NEXT TEST
2392						
2393	005432	023737	002210	002250	2\$: CMP T.MR3,E.MR3	;CHECK IF MR3 OK
2394	005440	001402			BEG 45	;YES - SKIP
2395	005442	104024			ERROR 24	;MR3 IN ERROR AT MULTI-SECTOR TIME
2396	005444	000413			BR TST5	;GO TO NEXT TEST
2397						
2398	005446	004437	025062		4\$: JSR R4,DSECTR	;SIMULATE 2ND SECTOR PULSE
2399						
2400	005452	004437	025504		JSR R4,DRSYNC	;SIMULATE PREAMBLE
2401	005456	104010			ERROR 10	;MR1 ERROR
2402	005460	104011			ERROR 11	;CS1 ERROR
2403						
2404	005462	004437	026042		JSR R4,DHDCMP	;SIMULATE HEADER SEARCH
2405	005466	104012			ERROR 12	;MR1 ERROR
2406	005470	104013			ERROR 13	;CS1 ERROR
2407	005472	104014			ERROR 14	;RKDCYL OR RKDA IN ERROR
2408						

;TEST 5 MID-TRANSFER SEEK (PART 3)

; CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.
; PUT THE CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
; DATA OF 401 WORDS IN 26 SECTOR FORMAT WITH CDT SET,
; CYLINDER 0, HEAD 2, SECTOR 25. CLOCK THROUGH SEEK
; AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE
; AND A GOOD HEADER. CLOCK THROUGH 400 WORDS AND THE
; TWO WORD ECC. MAKE SURE THE CONTROLLER ISSUES
; AN IMPLIED SEEK TO THE RIGHT HEAD AND CYLINDER.
; MAKE SURE THE NEXT HEADER RECOGNITION PROCEEDS PROPERLY.

2423	005474	000004			TST5: SCOPE	
2424	005476	012737	000012	001234	MOV #10.,\$TIMES	;DO 10. ITERATIONS
2425						
2426	005504	012762	100000	000000	MOV #CCLR,RKCS1(R2)	;CLEAR CONTROLLER
2427						
2428	005512	005037	002352		CLR BSEDES	;CLEAR BAD SEC DESIRED
2429						
2430	005516	004437	034026		JSR R4,LOADRK	;LOAD "L" REGISTERS
2431	005522	000000			0	;CYLINDER 0
2432	005524	025			.BYTE 25	;SECTOR 25
2433	005525	002			.BYTE 2	;TRACK 2
2434	005526	051444			OBUFF	;BUFFER ADDRESS OBUFF
2435	005530	177377			-401	;WORD COUNT -401
2436	005532	002023			WRDATA!CDT	;COMMAND WRDATA!CDT
2437						
2438	005534	004437	033612		JSR R4,BLDDAT	;GO BUILD DATA
2439	005540	000005			5	;PATTERN 5
2440						

B05

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 53
 DZR6EA.P11 28-SEP-76 15:31 TS MID-TRANSFER SEEK (PART 3)

SEQ 0053

2441	005542	004437	034072		JSR	R4,OPSTR	:START THE OPERATION
2442							
2443	005546	004437	025100		JSR	R4,DISEEK	:SIMULATE IMPLIED SEEK
2444	005552	104004			ERROR	4	:CS1 MISCOMPARE
2445	005554	104005			ERROR	5	:MR1 "
2446	005556	104006			ERROR	6	:MR2 "
2447	005560	104007			ERROR	7	:MR3 "
2448							
2449	005562	004437	025062		JSR	R4,DSECTR	:SIMULATE SECTOR PULSE
2450							
2451	005566	004437	025504		JSR	R4,DRSYNC	:SIMULATE HEADER PREAMBLE
2452	005572	104010			ERROR	10	:MR1 CONTENTS IN ERROR
2453	005574	104011			ERROR	11	:CS1 CONTENTS IN ERROR
2454							
2455	005576	004437	026342		JSR	R4,DHDCMP	:SIMULATE HEADER SEARCH
2456	005602	104012			ERROR	12	:MR1 CONTENTS IN ERROR
2457	005604	104013			ERROR	13	:CS1 IN ERROR AFTER SEARCH
2458	005606	104014			ERROR	14	:RKDCYL OR RKDA IN ERROR AFTER SEARCH
2459							
2460	005610	012737	045211	001510	MOV	#EMSO20,EMW	:SET MESSAGE FOR WRITE FAILURE
2461	005616	004437	027002		JSR	R4,DWGPSN	:SIMULATE GAP AND SYNC FOR WRITE
2462	005622	104015			ERROR	15	:MR1 IN ERROR IN READ GAP
2463	005624	104016			ERROR	16	:CS1 IN ERROR AFTER READ GAP
2464	005626	104017			ERROR	17	:MR1 IN ERROR IN WRITING SYNC
2465	005630	104020			ERROR	20	:CS1 IN ERROR AFTER WRITING SYNC
2466							
2467	005632	012737	045270	001510	MOV	#EMSO21,EMW	:SET MESSAGE FOR WRITE FAILURE
2468	005640	004437	027402		JSR	R4,DWRITE	:SIMULATE WRITE DATA AND ECC
2469	005644	104017			ERROR	17	:MR1 IN ERROR IN WRITING DATA OR ECC
2470	005646	104021			ERROR	21	:ECC IN ERROR WHILE WRITING DATA
2471	005650	104022			ERROR	22	:CS1 IN ERROR AFTER WRITING DATA AND ECC
2472							
2473	005652	012700	000324		MOV	#53, #4, R0	:ISSUE CLOCKS TO NEXT HDR COMPARE
2474	005656	012762	000440	000026 15:	MOV	#DMD!MCLK, RKMR1(R2)	:CLOCK LOOP
2475	005664	012762	000040	000026	MOV	#DMD, RKMR1(R2)	
2476	005672	005300			DEC	R0	
2477	005674	001370			BNE	15	
2478							
2479	005676	013703	002274		MOV	L.DCYL, R3	:GET DESIRED CYLINDER
2480	005702	004437	034242		JSR	R4,FSBLVV	:INVERT WORD
2481	005706	010337	002246		MOV	R3, E.MR2	:STORE EXPECTED MR2
2482	005712	113703	002267		MOVB	L.DT, R3	:GET DESIRED TRACK/SECTOR
2483	005716	042703	177400		BIC	#177400, R3	:CLEAR UNUSED BITS
2484	005722	012700	000005		MOV	#5, R0	:SET COUNT FOR SHIFTING
2485	005726	006303		55:	ASL	R3	:SHIFT TRACK FOR HEADER ALIGNMENT
2486	005730	005300			DEC	R0	:DEC COUNT
2487	005732	001375			BNE	55	:LOOP UNTIL ZERO
2488	005734	153703	002266		BISB	L.DS, R3	:INSERT SECTOR NUMBER
2489							
2490	005740	032737	010000	002260	BIT	#CFMT, L.CS1	:TEST IF 18 BIT MODE
2491	005746	001402			BEQ	65	:NO - SKIP
2492	005750	052703	001000		BIS	#BIT9, R3	:SET FORMAT BIT FOR WORD TWO
2493	005754			65:			
2494	005754	004437	034242		JSR	R4,FSBLVV	:INVERT WORD
2495	005760	010337	002250		MOV	R3, E.MR3	:STORE EXPECTED MR3
2496	005764	016237	000034	002206	MOV	RKMR2(R2), T.MR2	:GET MR2

C05

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZ6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 54
 DZ6EA.P11 28-SEP-76 15:31 TS MID-TRANSFER SEEK (PART 3)

SEQ 0054

```

2497 005772 016237 000036 002210      MOV      RKMR3(R2),T.MR3  ;GET MR3
2498 006000 016237 000000 002160      MOV      RKCS1(R2),T.CS1 ;GET CS1
2499 006006 013737 002260 002220      MOV      L.CS1,E.CS1    ;GET EXPECTED CS1
2500 006014 023737 002160 002220      CMP      T.CS1,E.CS1    ;CHECK IF OK
2501 006022 001402                BEQ      3$                ;YES - SKIP
2502 006024 104025                ERROR    25                ;CS1 IN ERROR AT MULTI-SECTOR TIME
2503 006026 000427                BR       TST6                ;GO TO NEXT TEST
2504
2505 006030 023737 002206 002246 3$:    CMP      T.MR2,E.MR2    ;CHECK IF MR2 OK
2506 006036 001402                BEQ      2$                ;YES - SKIP
2507 006040 104023                ERROR    23                ;MR2 IN ERROR AT MULTI-SECTOR TIME
2508 006042 000421                BR       TST6                ;GO TO NEXT TEST
2509
2510 006044 023737 002210 002250 2$:    CMP      T.MR3,E.MR3    ;CHECK IF MR3 OK
2511 006052 001402                BEQ      4$                ;YES - SKIP
2512 006054 104024                ERROR    24                ;MR3 IN ERROR AT MULTI-SECTOR TIME
2513 006056 000413                BR       TST6                ;GO TO NEXT TEST
2514
2515 006060 004437 025062                JSR      R4,DSECTR      ;SIMULATE 2ND SECTOR PULSE
2516
2517 006064 004437 025504                JSR      R4,DRSYNC      ;SIMULATE PREAMBLE
2518 006070 104010                ERROR    10                ;MR1 ERROR
2519 006072 104011                ERROR    11                ;CS1 ERROR
2520
2521 006074 004437 026042                JSR      R4,DHDCMP      ;SIMULATE HEADER SEARCH
2522 006100 104012                ERROR    12                ;MR1 ERROR
2523 006102 104013                ERROR    13                ;CS1 ERROR
2524 006104 104014                ERROR    14                ;RKDCYL OR RKDA IN ERROR
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539

```

 *TEST 6 MID-TRANSFER SEEK (PART 4)
 *

CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.
 PUT THE CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
 DATA OF 401 WORDS IN 26 SECTOR FORMAT WITH CDT SET,
 CYLINDER 0, HEAD 4, SECTOR 25. CLOCK THROUGH SEEK AND
 DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE
 AND A GOOD HEADER, CLOCK THROUGH 400 WORDS AND THE
 TWO WORD ECC. MAKE SURE THE CONTROLLER ISSUES
 IMPLIED SEEK TO THE RIGHT HEAD AND CYLINDER.
 MAKE SURE THE NEXT HEADER RECOGNITION PROCEEDS PROPERLY.

 *TST6: SCOPE
 *

```

2540 006106 000004                MOV      #10.,$TIMES    ;DO 10. ITERATIONS
2541 006110 012737 000012 001234      MOV      #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
2542
2543 006116 012762 100000 000000      CLR      BSEDES        ;CLEAR BAD SEC DESIRED
2544
2545 006124 005037 002352                JSR      R4,LOADRK      ;LOAD "L" REGISTERS
2546
2547 006130 004437 034026                D          0            ;CYLINDER 0
2548 006134 000000                .BYTE    25            ;SECTOR 25
2549 006136 0025                .BYTE    4             ;TRACK 4
2550 006137 004                OBUF      ;BUFFER ADDRESS OBUF
2551 006140 051444                -401          ;WORD COUNT -401
2552 006142 177377

```

2553	006144	002023		WRDATA!CDT		;COMMAND WRDATA!CDT
2554						
2555	006146	004437	033612	JSR	R4,BLDDAT	;GO BUILD DATA
2556	006152	000006		6		;PATTERN 6
2557						
2558	006154	004437	034072	JSR	R4,OPSTRT	;START THE OPERATION
2559						
2560	006160	004437	025100	JSR	R4,DISEEK	;SIMULATE IMPLIED SEEK
2561	006164	104004		ERROR	4	;CSI MISCOMPARE
2562	006166	104005		ERROR	5	;MR1
2563	006170	104006		ERROR	6	;MR2
2564	006172	104007		ERROR	7	;MR3
2565						
2566	006174	004437	025062	JSR	R4,DSECTR	;SIMULATE SECTOR PULSE
2567						
2568	006200	004437	025504	JSR	R4,DRSYNC	;SIMULATE HEADER PREAMBLE
2569	006204	104010		ERROR	10	;MR1 CONTENTS IN ERROR
2570	006206	104011		ERROR	11	;CSI CONTENTS IN ERROR
2571						
2572	006210	004437	026042	JSR	R4,DHDCMP	;SIMULATE HEADER SEARCH
2573	006214	104012		ERROR	12	;MR1 CONTENTS IN ERROR
2574	006216	104013		ERROR	13	;CSI IN ERROR AFTER SEARCH
2575	006220	104014		ERROR	14	;RKDCYL OR RKDA IN ERROR AFTER SEARCH
2576						
2577	006222	012737	045211	MOV	#EM5020,EMW	;SET MESSAGE FOR WRITE FAILURE
2578	006230	004437	027002	JSR	R4,DWGPSN	;SIMULATE GAP AND SYNC FOR WRITE
2579	006234	104015		ERROR	15	;MR1 IN ERROR IN READ GAP
2580	006236	104016		ERROR	16	;CSI IN ERROR AFTER READ GAP
2581	006240	104017		ERROR	17	;MR1 IN ERROR IN WRITING SYNC
2582	006242	104020		ERROR	20	;CSI IN ERROR AFTER WRITING SYNC
2583						
2584	006244	012737	045270	MOV	#EM5021,EMW	;SET MESSAGE FOR WRITE FAILURE
2585	006252	004437	027402	JSR	R4,DWRITE	;SIMULATE WRITE DATA AND ECC
2586	006256	104017		ERROR	17	;MR1 IN ERROR IN WRITING DATA OR ECC
2587	006260	104021		ERROR	21	;ECC IN ERROR WHILE WRITING DATA
2588	006262	104022		ERROR	22	;CSI IN ERROR AFTER WRITING DATA AND ECC
2589						
2590	006264	012700	000324	MOV	#53,*4,R0	;ISSUE CLOCKS TO NEXT HDR COMPARE
2591	006270	012762	000440	MOV	#DMD!MCLK,RKMR1(R2)	;CLOCK LOOP
2592	006276	012762	000040	MOV	#DMD,RKMR1(R2)	
2593	006304	005300		DEC	R0	
2594	006306	001370		BNE	15	
2595						
2596	006310	013703	002274	MOV	L.DCYL,R3	;GET DESIRED CYLINDER
2597	006314	004437	034242	JSR	R4,FSBLVV	;INVERT WORD
2598	006320	010337	002246	MOV	R3,E.MR2	;STORE EXPECTED MR2
2599	006324	113703	002267	MOVB	L.OT,R3	;GET DESIRED TRACK/SECTOR
2600	006330	042703	177400	BIC	#177400,R3	;CLEAR UNUSED BITS
2601	006334	012700	000005	MOV	#5,R0	;SET COUNT FOR SHIFTING
2602	006340	006303		ASL	R3	;SHIFT TRACK FOR HEADER ALIGNMENT
2603	006342	005300		DEC	R0	;DEC COUNT
2604	006344	001375		BNE	55	;LOOP UNTIL ZERO
2605	006346	153703	002266	BISB	L.DS,R3	;INSERT SECTOR NUMBER
2606						
2607	006352	032737	010000	BIT	#CFMT,L.CSI	;TEST IF 18 BIT MODE
2608	006360	001402		BEQ	65	;NO - SKIP

E05

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZ6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 56
DZ6EA.P11 28-SEP-76 15:31 T6 MID-TRANSFER SEEK (PART 4)

SEG 0056

2609	006362	052703	001000		BIS	#BIT9,R3	;SET FORMAT BIT FOR WORD TWO
2610	006366			68:			
2611	006366	004437	034242		JSR	R4,FSBLVV	;INVERT WORD
2612	006372	010337	002250		MOV	R3,E.MR3	;STORE EXPECTED MR3
2613	006376	016237	000034	002206	MOV	RKMR2(R2),T.MR2	;GET MR2
2614	006404	016237	000036	002210	MOV	RKMR3(R2),T.MR3	;GET MR3
2615	006412	016237	000000	002160	MOV	RKCS1(R2),T.CS1	;GET CS1
2616	006420	013737	002260	002220	MOV	L.CS1,E.CS1	;GET EXPECTED CS1
2617	006426	023737	002160	002220	CMP	T.CS1,E.CS1	;CHECK IF OK
2618	006434	001402			BEQ	35	;YES - SKIP
2619	006436	104025			ERROR	25	;CS1 IN ERROR AT MULTI-SECTOR TIME
2620	006440	000427			BR	TST	;GO TO NEXT TEST
2621							
2622	006442	023737	002206	002246	35:	CMP	T.MR2,E.MR2
2623	006450	001402			BEQ	25	;CHECK IF MR2 OK
2624	006452	104023			ERROR	23	;YES - SKIP
2625	006454	000421			BR	TST	;MR2 IN ERROR AT MULTI-SECTOR TIME
2626							;GO TO NEXT TEST
2627	006456	023737	002210	002250	25:	CMP	T.MR3,E.MR3
2628	006464	001402			BEQ	45	;CHECK IF MR3 OK
2629	006466	104024			ERROR	24	;YES - SKIP
2630	006470	000413			BR	TST	;MR3 IN ERROR AT MULTI-SECTOR TIME
2631							;GO TO NEXT TEST
2632	006472	004437	025062		45:	JSR	R4,DSECTR
2633							;SIMULATE 2ND SECTOR PULSE
2634	006476	004437	025504			JSR	R4,DRSYNC
2635	006502	104010			ERROR	10	;SIMULATE PREAMBLE
2636	006504	104011			ERROR	11	;MR1 ERROR
2637							;CS1 ERROR
2638	006506	004437	026042			JSR	R4,DHDCMP
2639	006512	104012			ERROR	12	;SIMULATE HEADER SEARCH
2640	006514	104013			ERROR	13	;MR1 ERROR
2641	006516	104014			ERROR	14	;CS1 ERROR
2642							;RKDCYL OR RKDA IN ERROR
2643							
2644							
2645							
2646							
2647							
2648							
2649							
2650							
2651							
2652							
2653							
2654							
2655							
2656							
2657	006520	000004			TST7:	SCOPE	
2658	006522	012737	000012	001234	MOV	#10,\$TIMES	;DO 10. ITERATIONS
2659							
2660	006530	012762	100000	000000	MOV	#CLR,RKCS1(R2)	;CLEAR CONTROLLER
2661							
2662	006536	005037	002352		CLR	BSEDES	;CLEAR BAD SEC DESIRED
2663	006542	012700	002000		MOV	#2000,R0	;SET A COUNT TO DELAY FOR CLEAR
2664	006546	005300			205:	DEC	R0

*TEST 7 MID-TRANSFER SEEK AND 24 SECTOR FORMAT
*
* CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
* PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE DATA
* DATA OF 401 WORDS TO AN RK06 IN 24 SECTOR FORMAT
* CYLINDER 0, HEAD 0, SECTOR 23. CLOCK THROUGH SEEK
* AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE
* AND A GOOD HEADER. CLOCK THROUGH 400 18 BIT WORDS
* AND THE 32 BIT ECC. CHECK THAT IMPLIED SEEK IS
* ISSUE TO NEXT TRACK. CHECK FOR PROPER HEADER RECOGNITION
* OCCURS.

F05

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZ6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 57
 DZ6EA.P11 28-SEP-76 15:31 T7 MID-TRANSFER SEEK AND 24 SECTOR FORMAT

SEQ 0057

2665	006550	001376		BNE	205		:LOOP UNTIL ZERO
2666							
2667	006552	004437	034026	JSR	R4,LOADRK		:LOAD "L" REGISTERS
2668	006556	000000		0			:CYLINDER 0
2669	006560	023		.BYTE	23		:SECTOR 23
2670	006561	000		.BYTE	0		:TRACK 0
2671	006562	051444		0BUFF			:BUFFER ADDRESS 0BUFF
2672	006564	177377		-401			:WORD COUNT -401
2673	006566	010023		WRDATA!CFMT			:COMMAND WRDATA!CFMT
2674							
2675	006570	004437	033612	JSR	R4,BLDDAT		:GO BUILD DATA
2676	006574	000007		7			:PATTERN 7
2677							
2678	006576	004437	034072	JSR	R4,OPSTR		:START THE OPERATION
2679							
2680	006602	004437	025100	JSR	R4,DISEEK		:SIMULATE IMPLIED SEEK
2681	006606	104004		ERROR	4		:CS1 MISCOMPARE
2682	006610	104005		ERROR	5		:MR1 "
2683	006612	104006		ERROR	6		:MR2 "
2684	006614	104007		ERROR	7		:MR3 "
2685							
2686	006616	004437	025062	JSR	R4,DSECTR		:SIMULATE SECTOR PULSE
2687							
2688	006622	004437	025504	JSR	R4,DRSYNC		:SIMULATE HEADER PREAMBLE
2689	006626	104010		ERROR	10		:MR1 CONTENTS IN ERROR
2690	006630	104011		ERROR	11		:CS1 CONTENTS IN ERROR
2691							
2692	006632	004437	026042	JSR	R4,DHDCMP		:SIMULATE HEADER SEARCH
2693	006636	104012		ERROR	12		:MR1 CONTENTS IN ERROR
2694	006640	104013		ERROR	13		:CS1 IN ERROR AFTER SEARCH
2695	006642	104014		ERROR	14		:RKDCYL OR RKDA IN ERROR AFTER SEARCH
2696							
2697	006644	012737	045211	MOV	#EM5020,EMW		:SET MESSAGE FOR WRITE FAILURE
2698	006652	004437	027002	JSR	R4,DWGPSN		:SIMULATE GAP AND SYNC FOR WRITE
2699	006656	104015		ERROR	15		:MR1 IN ERROR IN READ GAP
2700	006660	104016		ERROR	16		:CS1 IN ERROR AFTER READ GAP
2701	006662	104017		ERROR	17		:MR1 IN ERROR IN WRITING SYNC
2702	006664	104020		ERROR	20		:CS1 IN ERROR AFTER WRITING SYNC
2703							
2704	006666	012737	045270	MOV	#EM5021,EMW		:SET MESSAGE FOR WRITE FAILURE
2705	006674	004437	027402	JSR	R4,DWRITE		:SIMULATE WRITE DATA AND ECC
2706	006700	104017		ERROR	17		:MR1 IN ERROR IN WRITING DATA OR ECC
2707	006702	104021		ERROR	21		:ECC IN ERROR WHILE WRITING DATA
2708	006704	104022		ERROR	22		:CS1 IN ERROR AFTER WRITING DATA AND ECC
2709							
2710	006706	012700	000324	MOV	#53.*4,R0		:ISSUE CLOCKS TO NEXT HDR COMPARE
2711	006712	012762	000440	MOV	#DMD!MCLK,RKMR1(R2)		:CLOCK LOOP
2712	006720	012762	000040	MOV	#DMD,RKMR1(R2)		
2713	006726	005300		DEC	R0		
2714	006730	001370		BNE	15		
2715							
2716	006732	013703	002274	MOV	L.DCYL,R3		:GET DESIRED CYLINDER
2717	006736	004437	034242	JSR	R4,FSBLVV		:INVERT WORD
2718	006742	010337	002246	MOV	R3,E.MR2		:STORE EXPECTED MR2
2719	006746	113703	002267	MOVB	L.DT,R3		:GET DESIRED TRACK/SECTOR
2720	006752	042703	177400	BIC	#177400,R3		:CLEAR UNUSED BITS

G05

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 58
 DZR6EA.P11 28-SEP-76 15:31 T7 MID-TRANSFER SEEK AND 24 SECTOR FORMAT

SEQ 0058

```

2721 006756 012700 000005      MOV    #5,R0      ;SET COUNT FOR SHIFTING
2722 006762 006303      5$:    ASL     R3      ;SHIFT TRACK FOR HEADER ALIGNMENT
2723 006764 005300      DEC     R0      ;DEC COUNT
2724 006766 001375      BNE     5$      ;LOOP UNTIL ZERO
2725 006770 153703 002266      BISB    L.DS,R3    ;INSERT SECTOR NUMBER
2726
2727 006774 032737 010000 002260      BIT     #CFMT,L.CS1 ;TEST IF 18 BIT MODE
2728 007002 001402      BEQ     6$      ;NO - SKIP
2729 007004 052703 001000      BIS     #BIT9,R3   ;SET FORMAT BIT FOR WORD TWO
2730 007010      6$:
2731 007010 004437 034242      JSR     R4,FSBLVV   ;INVERT WORD
2732 007014 010337 002250      MOV     R3,E.MR3   ;STORE EXPECTED MR3
2733 007020 016237 000034 002206      MOV     RKMR2(R2),T.MR2 ;GET MR2
2734 007026 016237 000036 002210      MOV     RKMR3(R2),T.MR3 ;GET MR3
2735 007034 016237 000000 002160      MOV     RKCS1(R2),T.CS1 ;GET CS1
2736 007042 013737 002260 002220      MOV     L.CS1,E.CS1  ;GET EXPECTED CS1
2737 007050 023737 002160 002220      CMP     T.CS1,E.CS1  ;CHECK IF OK
2738 007056 001402      BEQ     3$      ;YES - SKIP
2739 007060 104025      ERROR   25      ;CS1 IN ERROR AT MULTI-SECTOR TIME
2740 007062 000427      BR      TST10   ;GO TO NEXT TEST
2741
2742 007064 023737 002206 002246 3$:    CMP     T.MR2,E.MR2  ;CHECK IF MR2 OK
2743 007072 001402      BEQ     2$      ;YES - SKIP
2744 007074 104023      ERROR   23      ;MR2 IN ERROR AT MULTI-SECTOR TIME
2745 007076 000421      BR      TST10   ;GO TO NEXT TEST
2746
2747 007100 023737 002210 002250 2$:    CMP     T.MR3,E.MR3  ;CHECK IF MR3 OK
2748 007106 001402      BEQ     4$      ;YES - SKIP
2749 007110 104024      ERROR   24      ;MR3 IN ERROR AT MULTI-SECTOR TIME
2750 007112 000413      BR      TST10   ;GO TO NEXT TEST
2751
2752 007114 004437 025062      4$:    JSR     R4,DSECTR  ;SIMULATE 2ND SECTOR PULSE
2753
2754 007120 004437 025504      JSR     R4,DRSYNC   ;SIMULATE PREAMBLE
2755 007124 104010      ERROR   10      ;MR1 ERROR
2756 007126 104011      ERROR   11      ;CS1 ERROR
2757
2758 007130 004437 026042      JSR     R4,DHDCMP   ;SIMULATE HEADER SEARCH
2759 007134 104012      ERROR   12      ;MR1 ERROR
2760 007136 104013      ERROR   13      ;CS1 ERROR
2761 007140 104014      ERROR   14      ;RKDCYL OR RKDA IN ERROR
2762
2763 *****
2764 *TEST 10 CYLINDER OVERFLOW (PART 1)
2765 *
2766 * CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.
2767 * PUT THE CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
2768 * DATA OF 400 WORDS TO AN RK06 IN 26 SECTOR FORMAT.
2769 * CYLINDER 632, HEAD 2, SECTOR 25. CLOCK THROUGH SEEK
2770 * AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE
2771 * AND A GOOD HEADER. CLOCK THROUGH 400 WORDS AND THE
2772 * TWO WORD ECC. MAKE SURE CYLINDER OVERFLOW ERROR DOES
2773 * NOT SET.
2774 *
2775 *****
2776 007142 000004      TST10: SCOPE

```

H05

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZRG6-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 59
 DZRG6A.P11 28-SEP-76 15:31 T10 CYLINDER OVERFLOW (PART 1)

SEG 3059

```

2777 007144 012737 000012 031234      MOV      #10.,$TIMES      ;;DO 10. ITERATIONS
2778
2779 007152 005037 002352                CLR      BSEDES          ;CLEAR BAD SEC DESIRED
2780 007156 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
2781
2782 007164 004437 034026                JSR      R4,LOADRK        ;LOAD "L" REGISTERS
2783 007170 000632                        632                ;CYLINDER 632
2784 007172 025                          .BYTE      25                ;SECTOR 25
2785 007173 002                          .BYTE      2                ;TRACK 2
2786 007174 051444                      OBUFF          ;BUFFER ADDRESS OBUFF
2787 007176 177400                      -400          ;WORD COUNT -400
2788 007200 000023                      WRDATA         ;COMMAND WRDATA
2789
2790 007202 004437 033612                JSR      R4,BLDDAT        ;GO BUILD DATA
2791 007206 000001                      1                ;PATTERN 1
2792
2793 007210 004437 034072                JSR      R4,OPSTAT        ;START THE OPERATION
2794
2795 007214 004437 025100                JSR      R4,DISEEK        ;SIMULATE IMPLIED SEEK
2796 007220 104004                      ERROR      4                ;CSI MISCOMPARE
2797 007222 104005                      ERROR      5                ;MR1
2798 007224 104006                      ERROR      6                ;MR2
2799 007226 104007                      ERROR      7                ;MR3
2800
2801 007230 004437 025062                JSR      R4,DSECTR        ;SIMULATE SECTOR PULSE
2802
2803 007234 004437 025504                JSR      R4,DRSYNC        ;SIMULATE HEADER PREAMBLE
2804 007240 104010                      ERROR      10               ;MR1 CONTENTS IN ERROR
2805 007242 104011                      ERROR      11               ;CSI CONTENTS IN ERROR
2806
2807 007244 004437 026042                JSR      R4,DHDCMP        ;SIMULATE HEADER SEARCH
2808 007250 104012                      ERROR      12               ;MR1 CONTENTS IN ERROR
2809 007252 104013                      ERROR      13               ;CSI IN ERROR AFTER SEARCH
2810 007254 104014                      ERROR      14               ;RKDCYL OR RKDA IN ERROR AFTER SEARCH
2811
2812 007256 012737 045211 001510      MOV      #EM5020,EMW      ;SET MESSAGE FOR WRITE FAILURE
2813 007264 004437 027002                JSR      R4,DWGPSN        ;SIMULATE GAP AND SYNC FOR WRITE
2814 007270 104015                      ERROR      15               ;MR1 IN ERROR IN READ GAP
2815 007272 104016                      ERROR      16               ;CSI IN ERROR AFTER READ GAP
2816 007274 104017                      ERROR      17               ;MR1 IN ERROR IN WRITING SYNC
2817 007276 104020                      ERROR      20               ;CSI IN ERROR AFTER WRITING SYNC
2818
2819 007300 012737 045270 001510      MOV      #EM5021,EMW      ;SET MESSAGE FOR WRITE FAILURE
2820 007306 004437 027402                JSR      R4,DWRITE        ;SIMULATE WRITE DATA AND ECC
2821 007312 104017                      ERROR      17               ;MR1 IN ERROR IN WRITING DATA OR ECC
2822 007314 104021                      ERROR      21               ;ECC IN ERROR WHILE WRITING DATA
2823 007316 104022                      ERROR      22               ;CSI IN ERROR AFTER WRITING DATA AND ECC
2824
2825
2826 007320 012700 000016                MOV      #3*4+2,R0        ;SET COUNT TO END OPERATION
2827 007324 012762 000440 000026 1$:  MOV      #DMD!MCLK,RKMR1(R2) ;SET CLOCK
2828 007332 012762 000040 000026      MOV      #DMD,RKMR1(R2) ;CLEAR CLOCK
2829 007340 005300                      RO                ;DEC COUNT
2830 007342 001370                      BNE      1$              ;LOOP UNTIL 0
2831
2832 007344 004437 034154                JSR      R4,GETREG        ;GET RK611 REGISTERS

```

105

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 60
DZR6EA.P11 28-SEP-76 15:31 T10 CYLINDER OVERFLOW (PART 1)

SEG 0060

```

2833 007350 032737 100000 002160 BIT #CERR,T.CS1 ;TEST IF ANY ERROR SET
2834 007356 001415 BEQ 3$ ;NO - SKIP TO EXIT
2835
2836 007360 032737 001000 002174 BIT #COE,T.ER ;TEST IF IT IS COE
2837 007366 001405 BEQ 2$ ;NO - SKIP
2838 007370 012737 046061 001622 MOV #E5026A,EH030 ;SET MESSAGE HEADER
2839 007376 104030 ERROR 30 ;"UNEXPECTED COE ERROR"
2840 007400 000404 BR 3$ ;EXIT
2841
2842 007402 012737 046061 001632 2$: MOV #E5026A,EH031 ;SET HEADER
2843 007410 104031 ERROR 31 ;"UNEXPECTED ERROR TESTING COE"
2844
2845 007412 3$:
2846
2847 ;*****
2848 ;TEST 11 CYLINDER OVERFLOW (PART 2)
2849 ;*
2850 ;* CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.
2851 ;* PUT THE CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
2852 ;* DATA OF 401 WORDS TO AN RK06 IN 26 SECTOR FORMAT.
2853 ;* CYLINDER 632, HEAD 2, SECTOR 25. CLOCK THROUGH SEEK
2854 ;* AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE
2855 ;* AND A GOOD HEADER. CLOCK THROUGH 400 WORDS AND
2856 ;* TWO ECC WORDS. MAKE SURE CYLINDER OVERFLOW ERROR SETS.
2857 ;*
2858 ;*****
2859 007412 000004 ST11: SCOPE
2860 007414 012737 000012 001234 MOV #10,$TIMES ;;DO 10. ITERATIONS
2861
2862 007422 005037 002352 CLR BSEDES ;CLEAR BAD SEC DESIRED
2863 007426 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
2864
2865 007434 004437 034026 JSR R4,LOADRK ;LOAD "L" REGISTERS
2866 007440 000632 632 ;CYLINDER 632
2867 007442 025 .BYTE 25 ;SECTOR 25
2868 007443 002 .BYTE 2 ;TRACK 2
2869 007444 051444 OBUFF ;BUFFER ADDRESS OBUFF
2870 007446 177377 -401 ;WORD COUNT -401
2871 007450 000023 WRDATA ;COMMAND WRDATA
2872
2873 007452 004437 033612 JSR R4,BLDDAT ;GO BUILD DATA
2874 007456 000002 2 ;PATTERN 2
2875
2876 007460 004437 034072 JSR R4,OPSTRT ;START THE OPERATION
2877
2878 007464 004437 025100 JSR R4,DISEEK ;SIMULATE IMPLIED SEEK
2879 007470 104004 ERROR 4 ;CS1 MISCOMPARE
2880 007472 104005 ERROR 5 ;MR1
2881 007474 104006 ERROR 6 ;MR2
2882 007476 104007 ERROR 7 ;MR3
2883
2884 007500 004437 025062 JSR R4,DSECTR ;SIMULATE SECTOR PULSE
2885
2886 007504 004437 025504 JSR R4,DRSYNC ;SIMULATE HEADER PREAMBLE
2887 007510 104010 ERROR 10 ;MR1 CONTENTS IN ERROR
2888 007512 104011 ERROR 11 ;CS1 CONTENTS IN ERROR

```

J05

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 61
 DZR6EA.P11 28-SEP-76 15:31 T11 CYLINDER OVERFLOW (PART 2)

SEG 0061

```

2889
2890 007514 004437 026042      JSR      R4,DHDCMP      ;SIMULATE HEADER SEARCH
2891 007520 104012      ERROR    12      ;MRI CONTENTS IN ERROR
2892 007522 104013      ERROR    13      ;CSI IN ERROR AFTER SEARCH
2893 007524 104014      ERROR    14      ;RKDCYL OR RKDA IN ERROR AFTER SEARCH
2894
2895 007526 012737 045211 001510  MOV      #EM5020,EMW      ;SET MESSAGE FOR WRITE FAILURE
2896 007534 004437 027002      JSR      R4,DWGPN      ;SIMULATE GAP AND SYNC FOR WRITE
2897 007540 104015      ERROR    15      ;MRI IN ERROR IN READ GAP
2898 007542 104016      ERROR    16      ;CSI IN ERROR AFTER READ GAP
2899 007544 104017      ERROR    17      ;MRI IN ERROR IN WRITING SYNC
2900 007546 104020      ERROR    20      ;CSI IN ERROR AFTER WRITING SYNC
2901
2902 007550 012737 045270 001510  MOV      #EM5021,EMW      ;SET MESSAGE FOR WRITE FAILURE
2903 007556 004437 027402      JSR      R4,DWRITE      ;SIMULATE WRITE DATA AND ECC
2904 007562 104017      ERROR    17      ;MRI IN ERROR IN WRITING DATA OR ECC
2905 007564 104021      ERROR    21      ;ECC IN ERROR WHILE WRITING DATA
2906 007566 104022      ERROR    22      ;CSI IN ERROR AFTER WRITING DATA AND ECC
2907
2908
2909 007570 012700 000016      MOV      #3*4+2,R0      ;SET COUNT TO COMPLETE OPERATION
2910 007574 012762 000440 000026 15:  MOV      #DMD!MCLK,RKMR1(R2) ;RUN THE CLOCK
2911 007602 012762 000040 000026      MOV      #DMD,RKMR1(R2)
2912 007610 005300      DEC      R0      ;DEC COUNT
2913 007612 001370      BNE      15      ;LOOP UNTIL 0
2914
2915 007614 004437 034154      JSR      R4,GETREG      ;GET RK611 REGS
2916 007620 032737 001000 002174  BIT      #COE,T.ER      ;TEST IF COE SET
2917 007626 001012      BNE      35      ;YES - EXIT TEST
2918
2919 007630 032737 100000 002160  BIT      #CERR,T.CSI      ;TEST IF ANY OTHER ERROR SET
2920 007636 001402      BEQ      25      ;NO - SKIP
2921 007640 104027      ERROR    27      ;"NO COE WHEN EXPECTED"
2922 007642 000404      BR       35      ;GO TO EXIT
2923
2924 007644 012737 046061 001602 25:  MOV      #E5026A,EH026      ;SET HEADER
2925 007652 104026      ERROR    26      ;"UNEXPECTED ERROR WHEN FORCING COE"
2926
2927 007654      35:
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942 007654 000004      ;*****
2943 007656 012737 000012 001234  ;TEST 12  CYLINDER OVERFLOW (PART 3)
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4590
4591
4592
4593
4594
4595
4596
4597
4598
4599
4600
4601
4602
4603
4604
4605
4606
4607
4608
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666
4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871
4872
4873
4874
4875
4876
4877
4878
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899
4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000
5001
5002
5003
5004
5005
5006
5007
5008
5009
5010
5011
5012
5013
5014
5015
5016
5017
5018
5019
5020
5021
5022
5023
5024
5025
5026
5027
5028
5029
5030
5031
5032
5033
5034
5035
5036
5037
5038
5039
5040
5041
5042
5043
5044
5045
5046
5047
5048
5049
5050
5051
5052
5053
5054
5055
5056
5057
5058
5059
5060
5061
5062
5063
5064
5065
5066
5067
5068
5069
5070
5071
5072
5073
5074
5075
5076
5077
5078
5079
5080
5081
5082
5083
5084
5085
5086
5087
5088
5089
5090
5091
5092
5093
5094
5095
5096
5097
5098
5099
5100
5101
5102
5103
5104

```

K05

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 62
 DZR6EA.P11 28-SEP-76 15:31 T12 CYLINDER OVERFLOW (PART 3)

SEQ 0062

2945	007664	012762	100000	000000	MOV	#CCLR,RKCS1(R2)	;CLEAR CONTROLLER
2946							
2947	007672	005037	002352		CLR	BSEDES	;CLEAR BAD SEC DESIRED
2948							
2949	007676	004437	034026		JSR	R4,LOADRK	;LOAD "L" REGISTERS
2950	007702	000632			632		;CYLINDER 632
2951	007704	025			.BYTE	25	;SECTOR 25
2952	007705	004			.BYTE	4	;TRACK 4
2953	007706	051444			OBUFF		;BUFFER ADDRESS OBUFF
2954	007710	177377			-401		;WORD COUNT -401
2955	007712	002023			WRDATA!CDT		;COMMAND WRDATA!CDT
2956							
2957	007714	004437	033612		JSR	R4,BLDDAT	;GO BUILD DATA
2958	007720	000003			3		;PATTERN 3
2959							
2960	007722	004437	034072		JSR	R4,OPSTRT	;START THE OPERATION
2961							
2962	007726	004437	025100		JSR	R4,DISEEK	;SIMULATE IMPLIED SEEK
2963	007732	104004			ERROR	4	;CS1 MISCOMPARE
2964	007734	104005			ERROR	5	;MR1 "
2965	007736	104006			ERROR	6	;MR2 "
2966	007740	104007			ERROR	7	;MR3 "
2967							
2968	007742	004437	025062		JSR	R4,DSECTR	;SIMULATE SECTOR PULSE
2969							
2970	007746	004437	025504		JSR	R4,DRSYNC	;SIMULATE HEADER PREAMBLE
2971	007752	104010			ERROR	10	;MR1 CONTENTS IN ERROR
2972	007754	104011			ERROR	11	;CS1 CONTENTS IN ERROR
2973							
2974	007756	004437	026042		JSR	R4,DHDCMP	;SIMULATE HEADER SEARCH
2975	007762	104012			ERROR	12	;MR1 CONTENTS IN ERROR
2976	007764	104013			ERROR	13	;CS1 IN ERROR AFTER SEARCH
2977	007766	104014			ERROR	14	;RKDCYL OR RKDA IN ERROR AFTER SEARCH
2978							
2979	007770	012737	045211	001510	MOV	#EM5020,EMW	;SET MESSAGE FOR WRITE FAILURE
2980	007776	004437	027002		JSR	R4,DWGPSN	;SIMULATE GAP AND SYNC FOR WRITE
2981	010002	104015			ERROR	15	;MR1 IN ERROR IN READ GAP
2982	010004	104016			ERROR	16	;CS1 IN ERROR AFTER READ GAP
2983	010006	104017			ERROR	17	;MR1 IN ERROR IN WRITING SYNC
2984	010010	104020			ERROR	20	;CS1 IN ERROR AFTER WRITING SYNC
2985							
2986	010012	012737	045270	001510	MOV	#EM5021,EMW	;SET MESSAGE FOR WRITE FAILURE
2987	010020	004437	027402		JSR	R4,DWRITE	;SIMULATE WRITE DATA AND ECC
2988	010024	104017			ERROR	17	;MR1 IN ERROR IN WRITING DATA OR ECC
2989	010026	104021			ERROR	21	;ECC IN ERROR WHILE WRITING DATA
2990	010030	104022			ERROR	22	;CS1 IN ERROR AFTER WRITING DATA AND ECC
2991							
2992							
2993	010032	012700	000016		MOV	#3*4+2,R0	;SET COUNT TO END OPERATION
2994	010036	012762	000440	000026	MOV	#DMD!MCLK,RKMR1(R2)	;SET CLOCK
2995	010044	012762	000040	000026	MOV	#DMD,RKMR1(R2)	;CLEAR CLOCK
2996	010052	005300			DEC	R0	;DEC COUNT
2997	010054	001370			BNE	15	;LOOP UNTIL 0
2998							
2999	010056	004437	034154		JSR	R4,GETREG	;GET RK611 REGISTERS
3000	010062	032737	100000	002160	BIT	#CERR,T.CS1	;TEST IF ANY ERROR SET

L05

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-CZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 63
 DZR6EA.P11 28-SEP-76 15:31 T12 CYLINDER OVERFLOW (PART 3)

SEQ 0053

```

3001 010070 001415      BEQ      3$      ;NO - SKIP TO EXIT
3002 010072 032737 001000 002174  BIT      #COE,T.ER  ;CHECK IF COE
3003 010100 001405      BEQ      2$      ;NO - SKIP
3004
3005 010102 012737 046232 001622  MOV      #E5026B,EH030 ;SET HEADER
3006 010110 104030      ERROR     30      ;"UNEXPECTED COE ERROR"
3007 010112 000404      BR        3$      ;EXIT
3008
3009 010114 012737 046232 001632 2$: MOV      #E5026B,EH031 ;SET HEADER
3010 010122 104031      ERROR     31      ;"UNEXPECTED ERROR WHILE TESTING COE"
3011
3012 010124      3$:
3013
3014 *****
3015 ;TEST 13      CYLINDER OVERFLOW (PART 4)
3016 ;
3017 ;      CLEAR THE RK611 CONTROLLER WITH A CONTROLLER CLEAR.
3018 ;      PUT THE CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A WRITE
3019 ;      DATA OF 401 WORDS IN 26 SECTOR FORMAT WITH CDT SET,
3020 ;      CYLINDER 1456, HEAD 4, SECTOR 25.  CLOCK THROUGH SEEK
3021 ;      AND DRIVE CLEAR MESSAGES.  SIMULATE A SECTOR PULSE
3022 ;      AND A GOOD HEADER.  CLOCK THROUGH 400 WORDS AND
3023 ;      TWO ECC WORDS.  MAKE SURE CYLINDER OVERFLOW ERROR SETS.
3024 ;
3025 *****
3026 010124 000004      TST13: SCOPE
3027 010126 012737 000012 001234  MOV      #10.,$TIMES  ;;DO 10. ITERATIONS
3028
3029 010134 005037 002352      CLR      BSEDES  ;CLEAR BAD SEC DESIRED
3030 010140 012762 100000 000000  MOV      #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
3031
3032 010146 004437 034026      JSR      R4,LOADRK  ;LOAD "L" REGISTERS
3033 010152 001456      1456      ;CYLINDER 1456
3034 010154      025      .BYTE     25      ;SECTOR 25
3035 010155      004      .BYTE     4       ;TRACK 4
3036 010156 051444      OBUFF     ;BUFFER ADDRESS OBUFF
3037 010160 177377      -401      ;WORD COUNT -401
3038 010162 002023      WRDATA!CDT ;COMMAND WRDATA!CDT
3039
3040 010164 004437 033612      JSR      R4,BLDDAT ;GO BUILD DATA
3041 010170 000004      4       ;PATTERN 4
3042
3043 010172 004437 034072      JSR      R4,OPSTRT ;START THE OPERATION
3044
3045 010176 004437 025100      JSR      R4,DISEEK ;SIMULATE IMPLIED SEEK
3046 010202 104004      ERROR     4       ;CS1 MISCOMPARE
3047 010204 104005      ERROR     5       ;MR1
3048 010206 104006      ERROR     6       ;MR2
3049 010210 104007      ERROR     7       ;MR3
3050
3051 010212 004437 025062      JSR      R4,DSECTR ;SIMULATE SECTOR PULSE
3052
3053 010216 004437 025504      JSR      R4,DRSYNC ;SIMULATE HEADER PREAMBLE
3054 010222 104010      ERROR     10      ;MR1 CONTENTS IN ERROR
3055 010224 104011      ERROR     11      ;CS1 CONTENTS IN ERROR
3056

```

M05

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 64
 DZR6EA.P11 28-SEP-76 15:31 T13 CYLINDER OVERFLOW (PART 4)

SEQ 0064

```

3057 010226 004437 026042      JSR      R4,DHDCMP      ;SIMULATE HEADER SEARCH
3058 010232 104012      ERROR    12      ;MR1 CONTENTS IN ERROR
3059 010234 104013      ERROR    13      ;CS1 IN ERROR AFTER SEARCH
3060 010236 104014      ERROR    14      ;RKDCYL OR RKDA IN ERROR AFTER SEARCH
3061
3062 010240 012737 045211 001510  MOV      #EM5020,EMW      ;SET MESSAGE FOR WRITE FAILURE
3063 010246 004437 027002      JSR      R4,DWGP5N      ;SIMULATE GAP AND SYNC FOR WRITE
3064 010252 104015      ERROR    15      ;MR1 IN ERROR IN READ GAP
3065 010254 104016      ERROR    16      ;CS1 IN ERROR AFTER READ GAP
3066 010256 104017      ERROR    17      ;MR1 IN ERROR IN WRITING SYNC
3067 010260 104020      ERROR    20      ;CS1 IN ERROR AFTER WRITING SYNC
3068
3069 010262 012737 045270 001510  MOV      #EM5021,EMW      ;SET MESSAGE FOR WRITE FAILURE
3070 010270 004437 027402      JSR      R4,DWRITE      ;SIMULATE WRITE DATA AND ECC
3071 010274 104017      ERROR    17      ;MR1 IN ERROR IN WRITING DATA OR ECC
3072 010276 104021      ERROR    21      ;ECC IN ERROR WHILE WRITING DATA
3073 010300 104022      ERROR    22      ;CS1 IN ERROR AFTER WRITING DATA AND ECC
3074
3075
3076 010302 012700 000016      MOV      #3*4+2,R0      ;SET CCUNT TO END OPERATION
3077 010306 012762 000440 000026 1$: MOV      #DMD!MCLK,RKMR1(R2) ;SET CLOCK
3078 010314 012762 000040 000026  MOV      #DMD,RKMR1(R2) ;CLEAR CLOCK
3079 010322 005300      DEC      R0      ;DEC COUNT
3080 010324 001370      BNE      1$      ;LOOP UNTIL 0
3081
3082 010326 004437 034154      JSR      R4,GETREG      ;GET RK611 REGISTERS
3083 010332 032737 001000 002174  BIT      #COE,T.ER      ;TEST IF COE SET
3084 010340 001012      BNE      3$      ;YES - EXIT
3085
3086 010342 032737 100000 002160  BIT      #CERR,T.CS1      ;TEST IF ANY OTHER ERROR SET
3087 010350 001402      BEQ      2$      ;NO SKIP
3088 010352 104027      ERROR    27      ;"UNEXPECTED ERROR WHILE TESTING COE"
3089 010354 000404      BR       3$      ;EXIT
3090
3091 010356 012737 046232 001602 2$: MOV      #E5026B,EH026      ;SET HEADER
3092 010364 104026      ERROR    26      ;"COE DID NOT SET WHEN EXPECTED"
3093 010366      3$:
3094
3095      .SBTTL  **NPR WRITING OF MEMORY
3096
3097      ;*****
3098      ;*TEST 14      NPR WRITING OF MEMORY
3099      ;*
3100      ;*      CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
3101      ;*      PUT CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A READ DATA
3102      ;*      OF 400 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
3103      ;*      CYLINDER 0, HEAD 0, SECTOR 0.  CLOCK THROUGH SEEK
3104      ;*      AND DRIVE CLEAR MESSAGES.  SIMULATE A SECTOR PULSE,
3105      ;*      A GOOD HEADER, 40 DATA WORDS.  VERIFY THAT THE WORDS
3106      ;*      ARE WRITTEN PROPERLY IN MEMORY.
3107      ;*
3108      ;*****
3109 010366 000004      1$T14: SCOPE
3110 010370 012737 000012 001234  MOV      #10.,$TIMES      ;;DO 10. ITERATIONS
3111
3112 010376 005037 002352      CLR      BSEDES      ;CLEAR BAD SEC DESIRED

```

N05

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-0CT-76 09:12 PAGE 65
 DZR6EA.P11 28-SEP-76 15:31 T14 NPR WRITING OF MEMORY

SEQ 0065

3113	010402	012762	100000	000000	MOV	#CCLR,RKCS1(R2)	;CLEAR CONTROLLER
3114							
3115	010410	005037	002344		CLR	ECCSRC	;CLEAR ECC SOURCE
3116	010414	005037	002354		CLR	HIBITS	;CLEAR HI ORDER BITS
3117	010420	004437	034026		JSR	R4,LOADRK	;LOAD "L" REGISTERS
3118	010424	000000			0		;CYLINDER 0
3119	010426	000			.BYTE	0	;SECTOR 0
3120	010427	000			.BYTE	0	;TRACK 0
3121	010430	050440			IBUFF		;BUFFER ADDRESS IBUFF
3122	010432	177400			-400		;WORD COUNT -400
3123	010434	000021			RDDATA		;COMMAND RDDATA
3124							
3125	010436	004437	033566		JSR	R4,CLRIBF	;GO CLEAR INPUT BUFFER
3126							
3127	010442	004437	033612		JSR	R4,BLDDAT	;GO BUILD DATA
3128	010446	000003			3		;PATTERN 3
3129							
3130							
3131	010450	004437	034072		JSR	R4,OPSTRT	;START THE OPERATION
3132	010454	004437	025100		JSR	R4,DISEEK	;SIMULATE IMPLIED SEEK
3133	010460	104004			ERROR	4	;CS1 MISCOMPARE
3134	010462	104005			ERROR	5	;MR1 "
3135	010464	104006			ERROR	6	;MR2 "
3136	010466	104007			ERROR	7	;MR3 "
3137							
3138	010470	004437	025062		JSR	R4,DSECTR	;SIMULATE SECTOR PULSE
3139							
3140	010474	004437	025504		JSR	R4,DRSYNC	;SIMULATE HEADER PREAMBLE
3141	010500	104010			ERROR	10	;MR1 CONTENTS IN ERROR
3142	010502	104011			ERROR	11	;CS1 CONTENTS IN ERROR
3143							
3144	010504	004437	026042		JSR	R4,DHDCMP	;SIMULATE HEADER SEARCH
3145	010510	104012			ERROR	12	;MR1 CONTENTS IN ERROR
3146	010512	104013			ERROR	13	;CS1 IN ERROR AFTER SEARCH
3147	010514	104014			ERROR	14	;RKDCYL OR RKDA IN ERROR AFTER SEARCH
3148							
3149	010516	004437	030276		JSR	R4,DRGPSN	;GO READ GAP AND SYNC
3150	010522	104015			ERROR	15	;MR1 IN ERROR READING GAP
3151	010524	104016			ERROR	16	;CS1 IN ERROR AFTER READING GAP
3152	010526	104032			ERROR	32	;MR1 IN ERROR READING SYNC
3153	010530	104033			ERROR	33	;CS1 IN ERROR AFTER READING SYNC
3154							
3155	010532	012737	000000	002346	MOV	#0,ECPATX	;LOAD EXPECTED PATTERN
3156	010540	012737	004066	002350	MOV	#4066,ECPOSX	;LOAD EXPECTED POSITION
3157	010546	004437	030774		JSR	R4,DREAD	;GO SIMULATE DATA READ
3158	010552	000060			60		;NUMBER OF WORDS TO SIMULATE
3159	010554	104034			ERROR	34	;MR1 IN ERROR READING DATA OR ECC
3160	010556	104035			ERROR	35	;ECC ERROR READING DATA
3161	010560	104036			ERROR	36	;CS1 ERROR AFTER READING DATA OR ECC
3162	010562	104041			ERROR	41	;ECC REG INCORRECT AFTER ECC READ
3163	010564	104042			ERROR	42	;ERR IN ECC PAT CALC.
3164	010566	104043			ERROR	43	;ERR IN ECC POS COUNT
3165							
3166							
3167	010570	012700	051444		MOV	#OBUFF,R0	;SET POINTER TO GOOD DATA
3168	010574	012701	050440		MOV	#IBUFF,R1	;SET POINTER TO INPUT DATA

B06

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZ6E-A MACY11 27.1006) 05-OCT-76 09:12 PAGE 66
 DZ6EA.P11 28-SEP-76 15:31 T14 NPR WRITING OF MEMORY

SEG 0066

```

3169
3170 010600 012704 000012      MOV      #10,R4      ;SET ERROR LIMIT COUNT
3171 010604 005037 001236      CLR      $ESCAPE    ;CLEAR ESCAPE
3172 010610 005037 001220      CLR      $TMP7      ;CLEAR ERROR REPORT SWITCH
3173 010614 012703 000040      MOV      #40,R3      ;SET COMPARE LIMIT
3174 010620 005005              CLR      R5          ;CLEAR WORD COUNTER
3175
3176 010622 021011      15:    CMP      (R0),(R1)    ;COMPARE DATA
3177 010624 001005              BNE      25          ;SKIP IF NOT EQUAL
3178 010626 005303      45:    DEC      R3          ;ELSE DEC WORD COUNT LIMIT
3179 010630 001424              BEQ      55          ;IF 0 - SKIP TO EXIT
3180 010632 005205              INC      R5          ;BUMP WORD COUNT
3181 010634 022021      CMP      (R0)+,(R1)+    ;BUMP DATA POINTERS
3182 010636 000771      BR       15          ;LOOP
3183
3184 010640 005304      25:    DEC      R4          ;DEC ERROR LIMIT
3185 010642 001417              BEQ      55          ;IF 0 - SKIP
3186
3187 010644 011037 001162      MOV      (R0),$REG0    ;GET GOOD WORD FOR REPORT
3188 010650 011137 001164      MOV      (R1),$REG1    ;GET BAD WORD
3189 010654 010537 001166      MOV      R5,$REG2    ;GET WORD NUMBER
3190
3191 010660 005737 001220      TST      $TMP7      ;DECIDE WHICH ERROR REPORT TO USE
3192 010664 001004              BNE      35          ;SKIP IF NOT 0
3193 010666 005237 001220      INC      $TMP7      ;ELSE BUMP $TMP7
3194 010672 104037      ERROR  37          ;REPORT FIRST ERROR LINE
3195 010674 000754      BR       45          ;SKIP
3196
3197 010676 104040      35:    ERROR  40          ;REPORT ALL OTHER LINES
3198 010700 000752      BR       45          ;SKIP
3199
3200 010702      55:
3201
3202      .SBTTL  **ECC ERROR DETECTION/CORRECTION TESTS
3203
3204      ;*****
3205      ;TEST 15      READ DATA WITH NO ERROR
3206      ;
3207      ;      CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
3208      ;      PUT CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A READ DATA
3209      ;      OF 400 WORDS TO AN RLL IN 26 SECTOR FORMAT.
3210      ;      CYLINDER 0, HEAD 0, SECTOR 0.  CLOCK THROUGH SEEK
3211      ;      AND DRIVE CLEAR MESSAGES.  SIMULATE A SECTOR PULSE.
3212      ;      A GOOD HEADER, 400 DATA WORDS, AND A GOOD 32 BIT ECC.
3213      ;      VERIFY THAT COMMAND COMPLETES WITHOUT ERROR.
3214      ;*****
3215
3216 010702 000004      15:    SCOPE
3217 010704 012737 000001 001234      MOV      #1,$TIMES    ;DO 1 ITERATION
3218
3219 010712 005037 002352      CLR      BSEDES      ;CLEAR BAD SEC DESIRED
3220 010716 012762 100000 000000      MOV      #CLR,RKCS1(R2) ;CLEAR CONTROLLER
3221 010724 005037 002344      CLR      ECCSRC      ;CLEAR ECC SOURCE FLAG
3222 010730 005037 002344      CLR      ECCSRC      ;CLEAR ECC SOURCE
3223 010734 005037 002354      CLR      HIBITS      ;CLEAR HI ORDER BITS
3224 010740 004437 034026      JSR      R4,LOADRK    ;LOAD "L" REGISTERS

```

C06

 RA611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZ6E-A MACY11 27(1006)
 DZ6EA.P11 28-SEP-76 15:31 T15 READ DATA WITH NO ERROR

05-OCT-76 09:12 PAGE 67

SEQ 0067

3225	010744	000000		0		:CYLINDER 0
3226	010746	000		.BYTE 0		:SECTOR 0
3227	010747	000		.BYTE 0		:TRACK 0
3228	010750	050440		IBUFF		:BUFFER ADDRESS IBUFF
3229	010752	177400		-400		:WORD COUNT -400
3230	010754	000021		RDDATA		:COMMAND RDDATA
3231						
3232	010756	004437	033566	JSR	R4,CLRIBF	:GO CLEAR INPUT BUFFER
3233						
3234	010762	004437	033612	JSR	R4,BLDDAT	:GO BUILD DATA
3235	010766	000004		4		:PATTERN 4
3236						
3237						
3238	010770	004437	034072	JSR	R4,OPSTRT	:START THE OPERATION
3239	010774	004437	025100	JSR	R4,DISEEK	:SIMULATE IMPLIED SEEK
3240	011000	104004		ERROR	4	:CSI MISCOMPARE
3241	011002	104005		ERROR	5	:MR1
3242	011004	104006		ERROR	6	:MR2
3243	011006	104007		ERROR	7	:MR3
3244						
3245	011010	004437	025062	JSR	R4,DSECTR	:SIMULATE SECTOR PULSE
3246						
3247	011014	004437	025504	JSR	R4,DRSYNC	:SIMULATE HEADER PREAMBLE
3248	011020	104010		ERROR	10	:MR1 CONTENTS IN ERROR
3249	011022	104011		ERROR	11	:CSI CONTENTS IN ERROR
3250						
3251	011024	004437	026042	JSR	R4,DHDCMP	:SIMULATE HEADER SEARCH
3252	011030	104012		ERROR	12	:MR1 CONTENTS IN ERROR
3253	011032	104013		ERROR	13	:CSI IN ERROR AFTER SEARCH
3254	011034	104014		ERROR	14	:RKDCYL OR RKDA IN ERROR AFTER SEARCH
3255						
3256	011036	004437	030276	JSR	R4,DRGPSN	:GO READ GAP AND SYNC
3257	011042	104015		ERROR	15	:MR1 IN ERROR READING GAP
3258	011044	104016		ERROR	16	:CSI IN ERROR AFTER READING GAP
3259	011046	104032		ERROR	32	:MR1 IN ERROR READING SYNC
3260	011050	104033		ERROR	33	:CSI IN ERROR AFTER READING SYNC
3261						
3262	011052	012737	000000 002346	MOV	#0,ECPATX	:LOAD EXPECTED PATTERN
3263	011060	012737	004066 002350	MOV	#4066,ECPOSX	:LOAD EXPECTED POSITION
3264	011066	004437	030774	JSR	R4,DREAD	:GO SIMULATE DATA READ
3265	011072	000400		400		:NUMBER OF WORDS TO SIMULATE
3266	011074	104034		ERROR	34	:MR1 IN ERROR READING DATA OR ECC
3267	011076	104035		ERROR	35	:ECC ERROR READING DATA
3268	011100	104036		ERROR	36	:CSI ERROR AFTER READING DATA OR ECC
3269	011102	104041		ERROR	41	:ECC REG INCORRECT AFTER ECC READ
3270	011104	104042		ERROR	42	:ERR IN ECC PAT CALC.
3271	011106	104043		ERROR	43	:ERR IN ECC POS COUNT
3272						
3273						
3274	011110	012700	002710	MOV	#40,*37.,R0	:SET COUNT TO EMPTY SILO
3275						
3276	011114	012762	000440 000026 15:	MOV	#DMD!MCLK,RKMR1,R2	:CLOCK CONTROLLER
3277	011122	012762	000040 000026	MOV	#DMD,RKMR1(R2)	
3278	011130	005300		DEC	R0	:DEC COUNT
3279	011132	001370		BNE	15	:LOOP UNTIL 0
3280						

006

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MC-11-CZ66E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 68
CZ66E.A.P11 28-SEP-76 15:31 T15 READ DATA WITH NO ERROR

SEG 0062

3281 011134 016237 000000 002160
3282 011142 004437 034154
3283 011146 042737 000001 002220
3284 011154 052737 000200 002220
3285 011162 023737 002160 002220
3286 011170 001401
3287 011172 104056
3288 011174

MOV R4,CS1,R2),T.CS1 :GET CS1
JSR R4,GETREG :GET 611 REGS
BIC #00,E.CS1 :CLEAR GO BIT
BIS #00,E.CS1 :SET READY BIT
CMP #0,CS1.E.CS1 :CHECK IF CS1 CORRECT
BEG 25 :YES - SKIP
ERROR 55 :CS1 IN ERROR AFTER READ DATA"

25:

TEST 16 READ DATA WITH ONE BIT BURST ERROR (PART 1)

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ DATA
OF 400 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE,
A GOOD HEADER, 400 DATA WORDS, AND AN ECC INDICATING
A ONE BIT BURST ERROR OF A 0 THAT SHOULD BE A 1.
VERIFY THE COUNTING OF THE POSITION
REGISTER AND THE FINAL PATTERN AND POSITION INFORMATION.
MAKE SURE DATA CHECK AND CONTROLLER ERROR SETS.

TEST 16: SCOPE

3304 011174 000004
3305 011176 012737 000001 001234
3306
3307 011204 005037 002352
3308 011210 012762 100000 000000
3309
3310 011216 005037 002354
3311 011222 004437 034026
3312 011226 000000
3313 011230 000
3314 011231 000
3315 011232 050440
3316 011234 177400
3317 011236 000021
3318
3319 011240 004437 033566
3320
3321 011244 004437 033612
3322 011250 000007
3323
3324 011252 012700 000530
3325
3326 011256 012737 177446 052444
3327 011264 012737 015457 052446
3328 011272 012737 000001 002344
3329 011300 032760 000100 051444
3330 011306 001004
3331 011310 052760 000100 051444
3332 011316 000403
3333 011320 042760 000100 051444
3334 011326
3335
3336 011326 004437 034072

MOV #1,STIMES :DO 1 ITERATION
CLR BSEDES :CLEAR BAD SEC DESIRED
MOV #0CLR,R4,CS1,R2) :CLEAR CONTROLLER
CLR HIBITS :CLEAR HI ORDER BITS
JSR R4,LOADRK :LOAD "L" REGISTERS
D :CYLINDER 0
.BYTE C :SECTOR 0
.BYTE C :TRACK 0
IBUFF :BUFFER ADDRESS IBUFF
-400 :WORD COUNT -400
RDATA :COMMAND RDATA
JSR R4,CLRIBF :GO CLEAR INPUT BUFFER
JSR R4,BLDDAT :GO BUILD DATA
? :PATTERN ?
MOV #530,R0 :SET INDEX INTO BUFFER
MOV #177446,0BUFF+1000 :STORE ECC WORDS
MOV #15457,0BUFF+1002
MOV #1,ECCSRC :SET SOURCE TO INDICATE ECC IN BUFFER
BIT #BIT6,0BUFF(R0) :TEST IF BIT SET
100\$:YES - SKIP
BNE #BIT6,0BUFF(R0) :ELSE SET BIT
BIS #BIT6,0BUFF(R0)
101\$
BIC #BIT6,0BUFF(R0) :CLEAR BIT
JSR R4,OPSTAT :START THE OPERATION

100\$:
101\$:

E06

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MC-11-D2R6E-A MACY11 27 1006 05-OCT-76 09:12 PAGE 69
 D2R6E.A.P11 28-SEP-76 15:31 T16 REAC DATA WITH ONE BIT BURST ERROR (PART 1)

SEG 0069

3337	011332	004437	025100		JSR	R4,DISEEK	:SIMULATE IMPLIED SEEK
3338	011336	104004			ERROR	4	:CSI MISCOMPARE
3339	011340	104005			ERROR	5	:MR1
3340	011342	104006			ERROR	6	:MR2
3341	011344	104007			ERROR	7	:MR3
3342							
3343	011346	004437	025062		JSR	R4,DSECTR	:SIMULATE SECTOR PULSE
3344							
3345	011352	004437	025504		JSR	R4,DRSYNC	:SIMULATE HEADER PREAMBLE
3346	011356	104010			ERROR	10	:MR1 CONTENTS IN ERROR
3347	011360	104011			ERROR	11	:CSI CONTENTS IN ERROR
3348							
3349	011362	004437	026042		JSR	R4,DHDCMP	:SIMULATE HEADER SEARCH
3350	011366	104012			ERROR	12	:MR1 CONTENTS IN ERROR
3351	011370	104013			ERROR	13	:CSI IN ERROR AFTER SEARCH
3352	011372	104014			ERROR	14	:RKDCYL OR RKDA IN ERROR AFTER SEARCH
3353							
3354	011374	004437	030276		JSR	R4,DRGPSN	:GO READ GAP AND SYNC
3355	011400	104015			ERROR	15	:MR1 IN ERROR READING GAP
3356	011402	104016			ERROR	16	:CSI IN ERROR AFTER READING GAP
3357	011404	104032			ERROR	32	:MR1 IN ERROR READING SYNC
3358	011406	104033			ERROR	33	:CSI IN ERROR AFTER READING SYNC
3359							
3360	011410	012737	000001	002346	MOV	#1,ECPATX	:LOAD EXPECTED PATTERN
3361	011416	012737	004066	002350	MOV	#4066,ECP0SX	:LOAD EXPECTED POSITION
3362	011424	004437	030774		JSR	R4,DREAD	:GO SIMULATE DATA READ
3363	011430	000400			400		:NUMBER OF WORDS TO SIMULATE
3364	011432	104034			ERROR	34	:MR1 IN ERROR READING DATA OF ECC
3365	011434	104035			ERROR	35	:ECC ERROR READING DATA
3366	011436	104036			ERROR	36	:CSI ERROR AFTER READING DATA OR ECC
3367	011440	104041			ERROR	41	:ECC REG INCORRECT AFTER ECC READ
3368	011442	104042			ERROR	42	:ERR IN ECC PAT CALC.
3369	011444	104043			ERROR	43	:ERR IN ECC POS COUNT
3370							
3371							
3372	011446	012737	000020	002326	MOV	#20,BITCNT	:SET BIT COUNT FOR CORRECTION COUNT
3373	011454	012700	000005		MOV	#5,R0	:SET COUNT FOR ECCPOS PASS THRU 0
3374	011460	012701	013672		MOV	#13672,R1	:SET COUNT FOR ECCPOS TO 0 FIRST PASS
3375	011464	005037	001214		CLR	\$TMP5	:CLEAR SWITCH
3376							
3377	011470	004737	033414		JSR	PC,RDBIT	:GO READ BIT
3378	011474	004737	032260		JSR	PC,ECCGEN	:GENERATE ECC
3379	011500	016237	000032	002214	MOV	RKECPT(R2),T.ECPT	:GET PATTERN
3380	011506	016237	000030	002212	MOV	RKECPS(R2),T.ECPS	:GET POSITION
3381	011514	005237	002350		INC	ECP0SX	:BUMP POSITION EXPECTED
3382	011520	013737	002350	002252	MOV	ECP0SX,E.ECPS	:SET FOR COMPARE AND REPORT
3383	011526	023737	002254	002214	CMP	E.ECPT,T.ECPT	:CHECK IF PATTERN CORRECT
3384	011534	001401			BEQ	1\$	:YES - SKIP
3385	011536	104042			ERROR	42	
3386	011540	023737	002252	002212	1\$: CMP	E.ECPS,T.ECPS	:CHECK IF POSITION CORRECT
3387	011546	001401			BEQ	2\$	:YES - SKIP
3388	011550	104043			ERROR	43	
3389	011552	005237	002326		2\$: INC	BITCNT	:BUMP BIT COUNT
3390	011556	005301			DEC	R1	:DEC COUNT TO ZERO IN ECCPOS
3391	011560	001343			BNE	3\$	:NOT YET ZERO - LOOP
3392	011562	005300			DEC	R0	:DEC PASS THRU 0 COUNT

F06

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZ6E-A MACY11 27,1006) 05-OCT-76 09:12 PAGE 70
 DZ6EA.P11 28-SEP-76 15:31 T16 READ DATA WITH ONE BIT BURST ERROR (PART 1)

SEQ 0070

```

3393 011564 001406      BEQ      4$      ;IF 0 - SKIP
3394 011566 012737 177777 002350  MOV      #-1,ECPOSX ;PRESET EXPECTED POS TO GO TO ZERO
3395 011574 012701 020000      MOV      #20000,R1 ;SET COUNT TO NEXT 0 IN ECC POS
3396 011600 000733      BR       3$      ;GO DO LOOP AGAIN
3397
3398 011602 005737 001214      4$:   TST      $TMP5      ;TEST SWITCH
3399 011606 001012      BNE      5$      ;ALREADY SET - SKIP
3400 011610 012701 005276      MOV      #5276,R1 ;SET COUNT TO FINAL POSITION
3401 011614 012700 000001      MOV      #1,R0 ;SET R0 TO RETURN AFTER ONE PASS
3402 011620 012737 177777 002350  MOV      #-1,ECPOSX ;SET EXP POS TO GO TO 0 ON NEXT BIT
3403 011626 005237 001214      INC      $TMP5      ;BUMP SWITCH
3404 011632 000716      BR       3$      ;GO DO LOOP AGAIN
3405
3406 011634 004737 033414      5$:   JSR      PC,RDBIT ;ONE MORE BIT TO SET READY
3407 011640 004437 034154      JSR      R4,GETREG ;GET '611 REGISTERS
3408 011644 013737 002260 002220  MOV      L.CS1,E.CS1 ;GET EXPECTED CS1
3409 011652 052737 100200 002220  BIS      #RDY!CERR,E.CS1 ;SET READY AND ERROR
3410 011660 042737 000001 002220  BIC      #GO,E.CS1 ;CLEAR GO
3411 011666 023737 002160 002220  CMP      T.CS1,E.CS1 ;CHECK IF CS1 CORRECT
3412 011674 001402      BEQ      6$      ;YES - SKIP
3413 011676 104044      ERROR     44
3414 011700 003410      BR       7$
3415 011702 032737 000100 002174  6$:   BIT      #ECH,T.ER ;TEST IF ECC HARD ERROR SET
3416 011710 001404      BEQ      7$      ;NO - SKIP
3417 011712 012737 100000 002234  MOV      #DCK,E.ER ;SET EXPECTED ERROR REG
3418 011720 104045      ERROR     45
3419 011722
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435 011722 000004      TEST17: SCOPE
3436 011724 012737 000001 001234  MOV      #1,$TIMES ;DO 1 ITERATION
3437 011732 005037 002352      CLR      BSEDES ;CLEAR BAD SEC DESIRED
3438 011736 012762 100000 000000  MOV      #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
3439
3440 011744 005037 002354      CLR      HIBITS ;CLEAR HI ORDER BITS
3441 011750 004437 034026      JSR      R4,LOADRK ;LOAD "L" REGISTERS
3442 011754 000000      0 ;CYLINDER 0
3443 011756 000 ;SECTOR 0
3444 011757 000 ;TRACK 0
3445 011760 050440      Ibuff ;BUFFER ADDRESS Ibuff
3446 011762 177400      -400 ;WORD COUNT -400
3447 011764 000021      Rddata ;COMMAND Rddata
3448

```

 :TEST 17 READ DATA WITH 11 BIT BURST ERROR

 : CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
 : PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ DATA
 : OF 400 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
 : CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
 : AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE
 : A GOOD HEADER, 400 DATA WORDS, AND AN ECC INDICATING
 : AN 11 BIT BURST ERROR. VERIFY THE COUNTING OF THE POSITION
 : REGISTER AND THE FINAL PATTERN AND POSITION INFORMATION.
 : MAKE SURE DATA CHECK AND CONTROLLER ERROR SETS.

 :TEST17: SCOPE

```

MOV      #1,$TIMES ;DO 1 ITERATION
CLR      BSEDES ;CLEAR BAD SEC DESIRED
MOV      #CCLR,RKCS1(R2) ;CLEAR CONTROLLER

CLR      HIBITS ;CLEAR HI ORDER BITS
JSR      R4,LOADRK ;LOAD "L" REGISTERS
0 ;CYLINDER 0
.BYTE 0 ;SECTOR 0
.BYTE 0 ;TRACK 0
Ibuff ;BUFFER ADDRESS Ibuff
-400 ;WORD COUNT -400
Rddata ;COMMAND Rddata

```

G06

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 71
 DZR6EA.P11 28-SEP-76 15:31 T17 READ DATA WITH 11 BIT BURST ERROR

SEQ 0071

3449	011766	004437	033566		JSR	R4,CLRIBF	;GO CLEAR INPUT BUFFER
3450							
3451	011772	004437	033612		JSR	R4,BLDDAT	;GO BUILD DATA
3452	011776	000007			7		;PATTERN 7
3453							
3454	012000	012700	000530		MOV	#530,R0	;SET INDEX INTO BUFFER
3455							
3456	012004	012737	177446	052444	MOV	#177446,0BUFF+1000	;STORE ECC WORDS
3457	012012	012737	015457	052446	MOV	#15457,0BUFF+1002	
3458	012020	012737	000001	002344	MOV	#1,ECCSRC	;SET SOURCE TO INDICATE ECC IN BUFFER
3459	012026	032760	000100	051444	BIT	#BIT6,0BUFF(R0)	;TEST IF BIT SET
3460	012034	001004			BNE	100\$	;YES - SKIP
3461	012036	052760	000100	051444	BIS	#BIT6,0BUFF(R0)	;ELSE SET BIT
3462	012044	000403			BR	101\$	
3463	012046	042760	000100	051444	BIC	#BIT6,0BUFF(R0)	;CLEAR BIT
3464	012054						
3465							
3466	012054	004437	034072		JSR	R4,OPSTRT	;START THE OPERATION
3467	012060	004437	025100		JSR	R4,DISEEK	;SIMULATE IMPLIED SEEK
3468	012064	104004			ERROR	4	;CS1 MISCOMPARE
3469	012066	104005			ERROR	5	;MR1 "
3470	012070	104006			ERROR	6	;MR2 "
3471	012072	104007			ERROR	7	;MR3 "
3472							
3473	012074	004437	025062		JSR	R4,DSECTR	;SIMULATE SECTOR PULSE
3474							
3475	012100	004437	025504		JSR	R4,DRSYNC	;SIMULATE HEADER PREAMBLE
3476	012104	104010			ERROR	10	;MR1 CONTENTS IN ERROR
3477	012106	104011			ERROR	11	;CS1 CONTENTS IN ERROR
3478							
3479	012110	004437	026042		JSR	R4,DHDCMP	;SIMULATE HEADER SEARCH
3480	012114	104012			ERROR	12	;MR1 CONTENTS IN ERROR
3481	012116	104013			ERROR	13	;CS1 IN ERROR AFTER SEARCH
3482	012120	104014			ERROR	14	;RKDCYL OR RKDA IN ERROR AFTER SEARCH
3483							
3484	012122	004437	030276		JSR	R4,DRGPSN	;GO READ GAP AND SYNC
3485	012126	104015			ERROR	15	;MR1 IN ERROR READING GAP
3486	012130	104016			ERROR	16	;CS1 IN ERROR AFTER READING GAP
3487	012132	104032			ERROR	32	;MR1 IN ERROR READING SYNC
3488	012134	104033			ERROR	33	;CS1 IN ERROR AFTER READING SYNC
3489							
3490	012136	012737	000001	002346	MOV	#1,ECPATX	;LOAD EXPECTED PATTERN
3491	012144	012737	004066	002350	MOV	#4066,ECPOSX	;LOAD EXPECTED POSITION
3492	012152	004437	030774		JSR	R4,DREAD	;GO SIMULATE DATA READ
3493	012156	000400			400		;NUMBER OF WORDS TO SIMULATE
3494	012160	104034			ERROR	34	;MR1 IN ERROR READING DATA OR ECC
3495	012162	104035			ERROR	35	;ECC ERROR READING DATA
3496	012164	104036			ERROR	36	;CS1 ERROR AFTER READING DATA OR ECC
3497	012166	104041			ERROR	41	;ECC REG INCORRECT AFTER ECC READ
3498	012170	104042			ERROR	42	;ERR IN ECC PAT CALC.
3499	012172	104043			ERROR	43	;ERR IN ECC POS COUNT
3500							
3501							
3502	012174	012737	000020	002326	MOV	#20,BITCNT	;SET BIT COUNT FOR CORRECTION COUNT
3503	012202	012700	000005		MOV	#5,R0	;SET COUNT FOR ECCPOS PASS THRU 0
3504	012206	012701	013672		MOV	#13672,R1	;SET COUNT FOR ECCPOS TO 0 FIRST PASS

H06

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 72
 DZR6EA.P11 28-SEP-76 15:31 T17 READ DATA WITH 11 BIT BURST ERROR

SEG 0072

```

3505 012212 005037 001214 CLR STMP5 ;CLEAR SWITCH
3506
3507 012216 004737 033414 3$: JSR PC,RDBIT ;GO READ BIT
3508 012222 004737 032260 JSR PC,ECCGEN ;GENERATE ECC
3509 012226 016237 000032 002214 MOV RKECPT(R2),T.ECPT ;GET PATTERN
3510 012234 016237 000030 002212 MOV RKECPS(R2),T.ECPS ;GET POSITION
3511 012242 005237 002350 INC ECPOSX ;BUMP POSITION EXPECTED
3512 012246 013737 002350 002252 MOV ECPOSX,E.ECPS ;SET FOR COMPARE AND REPORT
3513 012254 023737 002254 002214 CMP E.ECPT,T.ECPT ;CHECK IF PATTERN CORRECT
3514 012262 001401 BEQ 1$ ;YES - SKIP
3515 012264 104042 ERROR 42
3516 012266 023737 002252 002212 1$: CMP E.ECPS,T.ECPS ;CHECK IF POSITION CORRECT
3517 012274 001401 BEQ 2$ ;YES - SKIP
3518 012276 104043 ERROR 43
3519 012300 005237 002326 2$: INC BITCNT ;BUMP BIT COUNT
3520 012304 005301 DEC R1 ;DEC COUNT TO ZERO IN ECCPOS
3521 012306 001343 BNE 3$ ;NOT YET ZERO - LOOP
3522 012310 005300 DEC R0 ;DEC PASS THRU 0 COUNT
3523 012312 001406 BEQ 4$ ;IF 0 - SKIP
3524 012314 012737 177777 002350 MOV #-1,ECPOSX ;PRESET EXPECTED POS TO GO TO ZERO
3525 012322 012701 020000 MOV #20000,R1 ;SET COUNT TO NEXT 0 IN ECC POS
3526 012326 000733 BR 3$ ;GO DO LOOP AGAIN
3527
3528 012330 005737 001214 4$: TST STMP5 ;TEST SWITCH
3529 012334 001012 BNE 5$ ;ALREADY SET - SKIP
3530 012336 012701 005276 MOV #5276,R1 ;SET COUNT TO FINAL POSITION
3531 012342 012700 000001 MOV #1,R0 ;SET R0 TO RETURN AFTER ONE PASS
3532 012346 012737 177777 002350 MOV #-1,ECPOSX ;SET EXP POS TO GO TO 0 ON NEXT BIT
3533 012354 005237 001214 INC STMP5 ;BUMP SWITCH
3534 012360 000716 BR 3$ ;GO DO LOOP AGAIN
3535
3536 012362 004737 033414 5$: JSR PC,RDBIT ;ONE MORE BIT TO SET READY
3537 012366 004437 034154 JSR R4,GETREG ;GET '611 REGISTERS
3538 012372 013737 002260 002220 MOV L.CS1,E.CS1 ;GET EXPECTED CS1
3539 012400 052737 100200 002220 BIS #RDY!CERR,E.CS1 ;SET READY AND ERROR
3540 012406 042737 000001 002220 BIC #GO,E.CS1 ;CLEAR GO
3541 012414 023737 002160 002220 CMP T.CS1,E.CS1 ;CHECK IF CS1 CORRECT
3542 012422 001402 BEQ 6$ ;YES - SKIP
3543 012424 104044 ERROR 44
3544 012426 000410 BR 7$
3545 012430 032737 000100 002174 6$: BIT #ECH,T.ER ;TEST IF ECC HARD ERROR SET
3546 012436 001404 BEQ 7$ ;NO SKIP
3547 012440 012737 100000 002234 MOV #DCK,E.ER ;SET EXPECTED ERROR REG
3548 012446 104045 ERROR 45
3549
3550 012450 7$:
3551
3552 ;*****
3553 ;*TEST 20 READ DATA WITH ONE BIT BURST ERROR (PART 2)
3554 ;*
3555 ;*
3556 ;* CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
3557 ;* PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ DATA
3558 ;* OF 400 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
3559 ;* CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
3560 ;* AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE,
;* A GOOD HEADER, 400 DATA WORDS, AND AN ECC INDICATING

```

```

3561      ;*      A ONE BIT BURST ERROR OF A 1 THAT SHOULD BE A 0.
3562      ;*      VERIFY THE COUNTING OF THE POSITION
3563      ;*      REGISTER AND THE FINAL PATTERN AND POSITION INFORMATION.
3564      ;*      MAKE SURE DATA CHECK AND CONTROLLER ERROR SETS.
3565      ;*
3566      ;*****
3567 012450 000004      TST20: SCOPE
3568 012452 012737 000001 001234      MOV      #1,$TIMES      ;;DO 1 ITERATION
3569
3570 012460 005037 002352      CLR      BSEDES      ;CLEAR BAD SEC DESIRED
3571 012464 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
3572
3573 012472 005037 002354      CLR      HIBITS      ;CLEAR HI ORDER BITS
3574 012476 004437 034026      JSR      R4,LOADRK      ;LOAD "L" REGISTERS
3575 012502 000000      0      ;CYLINDER 0
3576 012504      000      .BYTE 0      ;SECTOR 0
3577 012505      000      .BYTE 0      ;TRACK 0
3578 012506 050440      Ibuff      ;BUFFER ADDRESS Ibuff
3579 012510 177400      -400      ;WORD COUNT -400
3580 012512 000021      RDDATA      ;COMMAND RDDATA
3581
3582 012514 004437 033566      JSR      R4,CLRIBF      ;GO CLEAR INPUT BUFFER
3583
3584 012520 004437 033612      JSR      R4,BLDDAT      ;GO BUILD DATA
3585 012524 000005      5      ;PATTERN 5
3586
3587 012526 012700 000530      MOV      #530,R0      ;SET INDEX INTO BUFFER
3588
3589 012532 012737 126073 052444      MOV      #126073,Obuff+1000      ;STORE ECC WORDS
3590 012540 012737 151052 052446      MOV      #151052,Obuff+1002
3591 012546 012737 000001 002344      MOV      #1,ECCSRC      ;SET SOURCE TO INDICATE ECC IN BUFFER
3592 012554 032760 000100 051444      BIT      #BIT6,Obuff(R0) ;TEST IF BIT SET
3593 012562 001004      BNE      100$      ;YES - SKIP
3594 012564 052760 000100 051444      BIS      #BIT6,Obuff(R0) ;ELSE SET BIT
3595 012572 000403      BR      101$
3596 012574 042760 000100 051444 100$: BIC      #BIT6,Obuff(R0) ;CLEAR BIT
3597 012602      101$:
3598
3599 012602 004437 034072      JSR      R4,OPSTRAT      ;START THE OPERATION
3600 012606 004437 025100      JSR      R4,DISEEK      ;SIMULATE IMPLIED SEEK
3601 012612 104004      ERROR 4      ;CS1 MISCOMPARE
3602 012614 104005      ERROR 5      ;MR1
3603 012616 104006      ERROR 6      ;MR2
3604 012620 104007      ERROR 7      ;MR3
3605
3606 012622 004437 025062      JSR      R4,DSECTR      ;SIMULATE SECTOR PULSE
3607
3608 012626 004437 025504      JSR      R4,DRSYNC      ;SIMULATE HEADER PREAMBLE
3609 012632 104010      ERROR 10      ;MR1 CONTENTS IN ERROR
3610 012634 104011      ERROR 11      ;CS1 CONTENTS IN ERROR
3611
3612 012636 004437 026042      JSR      R4,DHDCMP      ;SIMULATE HEADER SEARCH
3613 012642 104012      ERROR 12      ;MR1 CONTENTS IN ERROR
3614 012644 104013      ERROR 13      ;CS1 IN ERROR AFTER SEARCH
3615 012646 104014      ERROR 14      ;RKDCYL OR RKDA IN ERROR AFTER SEARCH
3616

```

3617	012650	004437	030276		JSR	R4,DRGPSN	;GO READ GAP AND SYNC
3618	012654	104015			ERROR	15	;MR1 IN ERROR READING GAP
3619	012656	104016			ERROR	16	;CS1 IN ERROR AFTER READING GAP
3620	012660	104032			ERROR	32	;MR1 IN ERROR READING SYNC
3621	012662	104033			ERROR	33	;CS1 IN ERROR AFTER READING SYNC
3622							
3623	012664	012737	000001	002346	MOV	#1,ECPATX	;LOAD EXPECTED PATTERN
3624	012672	012737	004066	002350	MOV	#4066,ECP0SX	;LOAD EXPECTED POSITION
3625	012700	004437	030774		JSR	R4,DREAD	;GO SIMULATE DATA READ
3626	012704	000400			400		;NUMBER OF WORDS TO SIMULATE
3627	012706	104034			ERROR	34	;MR1 IN ERROR READING DATA OR ECC
3628	012710	104035			ERROR	35	;ECC ERROR READING DATA
3629	012712	104036			ERROR	36	;CS1 ERROR AFTER READING DATA OR ECC
3630	012714	104041			ERROR	41	;ECC REG INCORRECT AFTER ECC READ
3631	012716	104042			ERROR	42	;ERR IN ECC PAT CALC.
3632	012720	104043			ERROR	43	;ERR IN ECC POS COUNT
3633							
3634							
3635	012722	012737	000020	002326	MOV	#20,BITCNT	;SET BIT COUNT FOR CORRECTION COUNT
3636	012730	012700	000005		MOV	#5,R0	;SET COUNT FOR ECCPOS PASS THRU 0
3637	012734	012701	013672		MOV	#13672,R1	;SET COUNT FOR ECCPOS TO 0 FIRST PASS
3638	012740	005037	001214		CLR	\$TMP5	;CLEAR SWITCH
3639							
3640	012744	004737	033414		JSR	PC,RDBIT	;GO READ BIT
3641	012750	004737	032260	3\$:	JSR	PC,ECCGEN	;GENERATE ECC
3642	012754	016237	000032	002214	MOV	RKECPT(R2),T.ECPT	;GET PATTERN
3643	012762	016237	000030	002212	MOV	RKECPS(R2),T.ECPS	;GET POSITION
3644	012770	005237	002350		INC	ECP0SX	;BUMP POSITION EXPECTED
3645	012774	013737	002350	002252	MOV	ECP0SX,E.ECPS	;SET FOR COMPARE AND REPORT
3646	013002	023737	002254	002214	CMP	E.ECPT,T.ECPT	;CHECK IF PATTERN CORRECT
3647	013010	001401			BEQ	1\$	;YES - SKIP
3648	013012	104042			ERROR	42	
3649	013014	023737	002252	002212	1\$:	CMP	E.ECPS,T.ECPS
3650	013022	001401			BEQ	2\$	;CHECK IF POSITION CORRECT
3651	013024	104043			ERROR	43	;YES - SKIP
3652	013026	005237	002326	2\$:	INC	BITCNT	;BUMP BIT COUNT
3653	013032	005301			DEC	R1	;DEC COUNT TO ZERO IN ECCPOS
3654	013034	001343			BNE	3\$	;NOT YET ZERO - LOOP
3655	013036	005300			DEC	R0	;DEC PASS THRU 0 COUNT
3656	013040	001406			BEQ	4\$	;IF 0 - SKIP
3657	013042	012737	177777	002350	MOV	#-1,ECP0SX	;PRESET EXPECTED POS TO GO TO ZERO
3658	013050	012701	020000		MOV	#20000,R1	;SET COUNT TO NEXT 0 IN ECC POS
3659	013054	000733			BR	3\$	;GO DO LOOP AGAIN
3660							
3661	013056	005737	001214	4\$:	TST	\$TMP5	;TEST SWITCH
3662	013062	001012			BNE	5\$	;ALREADY SET - SKIP
3663	013064	012701	005276		MOV	#5276,R1	;SET COUNT TO FINAL POSITION
3664	013070	012700	000001		MOV	#1,R0	;SET R0 TO RETURN AFTER ONE PASS
3665	013074	012737	177777	002350	MOV	#-1,ECP0SX	;SET EXP POS TO GO TO 0 ON NEXT BIT
3666	013102	005237	001214		INC	\$TMP5	;BUMP SWITCH
3667	013106	000716			BR	3\$	;GO DO LOOP AGAIN
3668							
3669	013110	004737	033414	5\$:	JSR	PC,RDBIT	;ONE MORE BIT TO SET READY
3670	013114	004437	034154		JSR	R4,GETREG	;GET '611 REGISTERS
3671	013120	013737	002260	002220	MOV	L.CS1,E.CS1	;GET EXPECTED CS1
3672	013126	052737	100200	002220	BIS	#RDY!CERR,E.CS1	;SET READY AND ERROR

K06

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 75
 DZR6EA.P11 28-SEP-76 15:31 T20 READ DATA WITH ONE BIT BURST ERROR (PART 2)

SEG 0075

3673	013134	042737	000001	002220	BIC	#G0,E.CS1	:CLEAR G0
3674	013142	023737	002160	002220	CMP	T.CS1,E.CS1	:CHECK IF CS1 CORRECT
3675	013150	001402			BEQ	6\$	:YES - SKIP
3676	013152	104044			ERROR	4	
3677	013154	000410			BR	7\$	
3678	013156	032737	000100	002174	BIT	#ECH,T.ER	:TEST IF ECC HARD ERROR SET
3679	013164	001404			BEQ	7\$	:NO SKIP
3680	013166	012737	100000	002234	MOV	#DCK,E.ER	:SET EXPECTED ERROR REG
3681	013174	104045			ERROR	45	
3682							
3683	013176				7\$:		
3684							
3685							
3686							
3687							
3688							
3689							
3690							
3691							
3692							
3693							
3694							
3695							
3696							
3697							
3698							
3699	013176	000004					
3700	013200	012737	000001	001234	TEST21:	SCOPE	
3701					MOV	#1,\$TIMES	::DO 1 ITERATION
3702	013206	005037	002352		CLR	BSEDES	:CLEAR BAD SEC DESIRED
3703							
3704	013212	012762	100000	000000	MOV	#CCLR,RKCS1(R2)	:CLEAR CONTROLLER
3705							
3706	013220	005037	002354		CLR	HIBITS	:CLEAR HI ORDER BITS
3707	013224	004437	034026		JSR	R4,LOADRK	:LOAD "L" REGISTERS
3708	013230	000000			0		:CYLINDER 0
3709	013232	000			.BYTE	0	:SECTOR 0
3710	013233	000			.BYTE	0	:TRACK 0
3711	013234	050440			IBUFF		:BUFFER ADDRESS IBUFF
3712	013236	177400			-400		:WORD COUNT -400
3713	013240	000021			RDDATA		:COMMAND RDDATA
3714							
3715	013242	004437	033566		JSR	R4,CLRIBF	:GO CLEAR INPUT BUFFER
3716							
3717	013246	004437	033612		JSR	R4,BLDDAT	:GO BUILD DATA
3718	013252	000005			5		:PATTERN 5
3719							
3720	013254	012700	000530		MOV	#530,R0	:SET INDEX INTO BUFFER
3721							
3722	013260	012737	126073	052444	MOV	#126073,0BUFF+1000	:STORE ECC WORDS
3723	013266	012737	151052	052446	MOV	#151052,0BUFF+1002	
3724	013274	012737	000001	002344	MOV	#1,ECCSRC	:SET SOURCE TO INDICATE ECC IN BUFFER
3725	013302	042760	005000	051444	BIC	#BIT9!BIT11,0BUFF(R0)	:CREATE 11 BIT BURST ERROR
3726	013310	052760	050100	051444	BIS	#BIT6!BIT12!BIT14,0BUFF(R0)	
3727	013316	062700	000002		ADD	#2,R0	:BUMP TO NEXT WORD
3728	013322	052760	000001	051444	BIS	#BIT0,0BUFF(R0)	

L06

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 76
DZR6EA.P11 28-SEP-76 15:31 T21 READ DATA WITH 11 BIT BURST ERROR

SEQ 0076

3729									
3730	013330	004437	034072		JSR	R4,OPSTRT		;START THE OPERATION	
3731	013334	004437	025100		JSR	R4,DISEEK		;SIMULATE IMPLIED SEEK	
3732	013340	104004			ERROR	4		;CS1 MISCOMPARE	
3733	013342	104005			ERROR	5		;MR1 "	
3734	013344	104006			ERROR	6		;MR2 "	
3735	013346	104007			ERROR	7		;MR3 "	
3736									
3737	013350	004437	025062		JSR	R4,DSECTR		;SIMULATE SECTOR PULSE	
3738									
3739	013354	004437	025504		JSR	R4,DRSYNC		;SIMULATE HEADER PREAMBLE	
3740	013360	104010			ERROR	10		;MR1 CONTENTS IN ERROR	
3741	013362	104011			ERROR	11		;CS1 CONTENTS IN ERROR	
3742									
3743	013364	004437	026042		JSR	R4,DHDCMP		;SIMULATE HEADER SEARCH	
3744	013370	104012			ERROR	12		;MR1 CONTENTS IN ERROR	
3745	013372	104013			ERROR	13		;CS1 IN ERROR AFTER SEARCH	
3746	013374	104014			ERROR	14		;RKDCYL OR RKDA IN ERROR AFTER SEARCH	
3747									
3748	013376	004437	030276		JSR	R4,DRGPSN		;GO READ GAP AND SYNC	
3749	013402	104015			ERROR	15		;MR1 IN ERROR READING GAP	
3750	013404	104016			ERROR	16		;CS1 IN ERROR AFTER READING GAP	
3751	013406	104032			ERROR	32		;MR1 IN ERROR READING SYNC	
3752	013410	104033			ERROR	33		;CS1 IN ERROR AFTER READING SYNC	
3753									
3754	013412	012737	000001	002346	MOV	#1,ECPATX		;LOAD EXPECTED PATTERN	
3755	013420	012737	004066	002350	MOV	#4066,ECP0SX		;LOAD EXPECTED POSITION	
3756	013426	004437	030774		JSR	R4,DREAD		;GO SIMULATE DATA READ	
3757	013432	000400			400			;NUMBER OF WORDS TO SIMULATE	
3758	013434	104034			ERROR	34		;MR1 IN ERROR READING DATA OR ECC	
3759	013436	104035			ERROR	35		;ECC ERROR READING DATA	
3760	013440	104036			ERROR	36		;CS1 ERROR AFTER READING DATA OR ECC	
3761	013442	104041			ERROR	41		;ECC REG INCORRECT AFTER ECC READ	
3762	013444	104042			ERROR	42		;ERR IN ECC PAT CALC.	
3763	013446	104043			ERROR	43		;ERR IN ECC POS COUNT	
3764									
3765									
3766	013450	012737	000020	002326	MOV	#20,BITCNT		;SET BIT COUNT FOR CORRECTION COUNT	
3767	013456	012700	000005		MOV	#5,R0		;SET COUNT FOR ECCPOS PASS THRU 0	
3768	013462	012701	013672		MOV	#13672,R1		;SET COUNT FOR ECCPOS TO 0 FIRST PASS	
3769	013466	005037	001214		CLR	STMP5		;CLEAR SWITCH	
3770									
3771	013472	004737	033414		3\$: JSR	PC,RDBIT		;GO READ BIT	
3772	013476	004737	032260		JSR	PC,ECCGEN		;GENERATE ECC	
3773	013502	016237	000032	002214	MOV	RKECPT(R2),T.ECPT		;GET PATTERN	
3774	013510	016237	000030	002212	MOV	RKECPS(R2),T.ECPS		;GET POSITION	
3775	013516	005237	002350		INC	ECPOSX		;BUMP POSITION EXPECTED	
3776	013522	013737	002350	002252	MOV	ECPOSX,E.ECPS		;SET FOR COMPARE AND REPORT	
3777	013530	023737	002254	002214	CMP	E.ECPT,T.ECPT		;CHECK IF PATTERN CORRECT	
3778	013536	001401			BEG	1\$		;YES - SKIP	
3779	013540	104042			ERROR	42			
3780	013542	023737	002252	002212	1\$: CMP	E.ECPS,T.ECPS		;CHECK IF POSITION CORRECT	
3781	013550	001401			BEG	2\$		;YES - SKIP	
3782	013552	104043			ERROR	43			
3783	013554	005237	002326		2\$: INC	BITCNT		;BUMP BIT COUNT	
3784	013560	005301			DEC	R1		;DEC COUNT TO ZERO IN ECCPOS	

M06

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 77
 DZR6EA.P11 28-SEP-76 15:31 T21 READ DATA WITH 11 BIT BURST ERROR

SEQ 0077

```

3785 013562 001343      BNE      3$      ;NOT YET ZERO - LOOP
3786 013564 005300      DEC      R0      ;DEC PASS THRU 0 COUNT
3787 013566 001406      BEQ      4$      ;IF 0 - SKIP
3788 013570 012737 177777 002350      MOV      #-1,ECPOSX ;PRESET EXPECTED POS TO GO TO ZERO
3789 013576 012701 020000      MOV      #20000,R1 ;SET COUNT TO NEXT 0 IN ECC POS
3790 013602 000733      BR       3$      ;GO DO LOOP AGAIN
3791
3792 013604 005737 001214      4$:    TST      $TMP5      ;TEST SWITCH
3793 013610 001012      BNE      5$      ;ALREADY SET - SKIP
3794 013612 012701 005310      MOV      #5310,R1      ;SET COUNT TO FINAL POSITION
3795 013616 012700 000001      MOV      #1,R0      ;SET R0 TO RETURN AFTER ONE PASS
3796 013622 012737 177777 002350      MOV      #-1,ECPOSX ;SET EXP POS TO GO TO 0 ON NEXT BIT
3797 013630 005237 001214      INC      $TMP5      ;BUMP SWITCH
3798 013634 000716      BR       3$      ;GO DO LOOP AGAIN
3799
3800 013636 004737 033414      5$:    JSR      PC,RDBIT      ;ONE MORE BIT TO SET READY
3801 013642 004437 034154      JSR      R4,GETREG      ;GET '611 REGISTERS
3802 013646 013737 002260 002220      MOV      L.CS1,E.CS1 ;GET EXPECTED CS1
3803 013654 052737 100200 002220      BIS      #RDY!CERR,E.CS1 ;SET READY AND ERROR
3804 013662 042737 000001 002220      BIC      #GO,E.CS1      ;CLEAR GO
3805 013670 023737 002160 002220      CMP      T.CS1,E.CS1 ;CHECK IF CS1 CORRECT
3806 013676 001402      BEQ      6$      ;YES - SKIP
3807 013700 104044      ERROR      44
3808 013702 000410      BR       7$
3809 013704 032737 000100 002174      6$:    BIT      #ECH,T.ER      ;TEST IF ECC HARD ERROR SET
3810 013712 001404      BEQ      7$      ;NO SKIP
3811 013714 012737 100000 002234      MOV      #DCK,E.ER      ;SET EXPECTED ERROR REG
3812 013722 104045      ERROR      45
3813
3814 013724      7$:
3815
3816 *****
3817 *TEST 22      READ DATA WITH 12 BIT BURST ERROR
3818 *
3819 *      CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
3820 *      PUT CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A READ DATA
3821 *      OF 400 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
3822 *      CYLINDER 0, HEAD 0, SECTOR 0.  CLOCK THROUGH SEEK
3823 *      AND DRIVE CLEAR MESSAGES.  SIMULATE A SECTOR PULSE, A
3824 *      GOOD HEADER, 400 DATA WORDS AND AN ECC INDICATING
3825 *      A 12 BIT BURST ERROR.  VERIFY THE COUNTING OF THE
3826 *      POSITION REGISTER.  MAKE SURE DATA CHECK, ECC HARD,
3827 *      AND CONTROLLER ERROR SETS.
3828 *
3829 *****
3830 013724 000004      †ST22:  SCOPE
3831 013726 012737 000001 001234      MOV      #1,$TIMES      ;;DO 1 ITERATION
3832
3833 013734 005037 002352      CLR      BSEDES      ;CLEAR BAD SEC DESIRED
3834 013740 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
3835
3836 013746 005037 002354      CLR      HIBITS      ;CLEAR HI ORDER BITS
3837 013752 004437 034026      JSR      R4,LOADRK      ;LOAD "L" REGISTERS
3838 013756 000000      0      ;CYLINDER 0
3839 013760      000      .BYTE      0      ;SECTOR 0
3840 013761      000      .BYTE      0      ;TRACK 0

```

N06

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 78
DZR6EA.P11 28-SEP-76 15:31 T22 READ DATA WITH 12 BIT BURST ERROR

SEQ 0078

3841	013762	050440			IBUFF		; BUFFER ADDRESS IBUFF
3842	013764	177400			-400		; WORD COUNT -400
3843	013766	000021			RDDATA		; COMMAND RDDATA
3844							
3845	013770	004437	033566		JSR	R4,CLRIBF	; GO CLEAR INPUT BUFFER
3846							
3847	013774	004437	033612		JSR	R4,BLDDAT	; GO BUILD DATA
3848	014000	000005			5		; PATTERN 5
3849							
3850	014002	012700	000530		MOV	#530,R0	; SET INDEX INTO BUFFER
3851							
3852	014006	012737	126073	052444	MOV	#126073,0BUFF+1000	; STORE ECC WORDS
3853	014014	012737	151052	052446	MOV	#151052,0BUFF+1002	
3854	014022	012737	000001	002344	MOV	#1,ECCSRC	; SET SOURCE TO INDICATE ECC IN BUFFER
3855	014030	042760	005000	051444	BIC	#BIT9:BIT11,0BUFF(R0)	; CREATE 11 BIT BURST ERROR
3856	014036	052760	050100	051444	BIS	#BIT6:BIT12:BIT14,0BUFF(R0)	
3857	014044	062700	000002		ADD	#2,R0	; BUMP TO NEXT WORD
3858	014050	052760	000001	051444	BIS	#BIT0,0BUFF(R0)	
3859	014056	042760	000002	051444	BIC	#BIT1,0BUFF(R0)	; MAKE IT A 12 BIT BURST ERROR
3860							
3861	014064	004437	034072		JSR	R4,OPSTRT	; START THE OPERATION
3862	014070	004437	025100		JSR	R4,DISEEK	; SIMULATE IMPLIED SEEK
3863	014074	104004			ERROR	4	; CS1 MISCOMPARE
3864	014076	104005			ERROR	5	; MR1
3865	014100	104006			ERROR	6	; MR2
3866	014102	104007			ERROR	7	; MR3
3867							
3868	014104	004437	025062		JSR	R4,DSECTR	; SIMULATE SECTOR PULSE
3869							
3870	014110	004437	025504		JSR	R4,DRSYNC	; SIMULATE HEADER PREAMBLE
3871	014114	104010			ERROR	10	; MR1 CONTENTS IN ERROR
3872	014116	104011			ERROR	11	; CS1 CONTENTS IN ERROR
3873							
3874	014120	004437	026042		JSR	R4,DHDCMP	; SIMULATE HEADER SEARCH
3875	014124	104012			ERROR	12	; MR1 CONTENTS IN ERROR
3876	014126	104013			ERROR	13	; CS1 IN ERROR AFTER SEARCH
3877	014130	104014			ERROR	14	; RKDCYL OR RKDA IN ERROR AFTER SEARCH
3878							
3879	014132	004437	030276		JSR	R4,DRGPSN	; GO READ GAP AND SYNC
3880	014136	104015			ERROR	15	; MR1 IN ERROR READING GAP
3881	014140	104016			ERROR	16	; CS1 IN ERROR AFTER READING GAP
3882	014142	104032			ERROR	32	; MR1 IN ERROR READING SYNC
3883	014144	104033			ERROR	33	; CS1 IN ERROR AFTER READING SYNC
3884							
3885	014146	012737	000001	002346	MOV	#1,ECPATX	; LOAD EXPECTED PATTERN
3886	014154	012737	004066	002350	MOV	#4066,ECPOSX	; LOAD EXPECTED POSITION
3887	014162	004437	030774		JSR	R4,DREAD	; GO SIMULATE DATA READ
3888	014166	000400			400		; NUMBER OF WORDS TO SIMULATE
3889	014170	104034			ERROR	34	; MR1 IN ERROR READING DATA OR ECC
3890	014172	104035			ERROR	35	; ECC ERROR READING DATA
3891	014174	104036			ERROR	36	; CS1 ERROR AFTER READING DATA OR ECC
3892	014176	104041			ERROR	41	; ECC REG INCORRECT AFTER ECC READ
3893	014200	104042			ERROR	42	; ERR IN ECC PAT CALC.
3894	014202	104043			ERROR	43	; ERR IN ECC POS COUNT
3895							
3896							

B07

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZ6E-A MACY11 27.1006) 05-007-76 09:12 PAGE 79
 DZ6EA.P11 28-SEP-76 15:31 T22 READ DATA WITH 12 BIT BURST ERROR

SEQ 0079

```

3897 014204 012737 000020 002326 MOV #20,BITCNT ;SET BIT COUNT FOR CORRECTION COUNT
3898 014212 012700 000005 MOV #5,R0 ;SET COUNT FOR ECCPOS PASS THRU 0
3899 014216 012701 013672 MOV #13&72,R1 ;SET COUNT FOR ECCPOS TO 0 FIRST PASS
3900 014222 005037 001214 CLR $TMP5 ;CLEAR SWITCH
3901
3902 014226 004737 033414 35: JSR PC,ROBIT ;GO READ BIT
3903 014232 004737 032260 JSR PC,ECCGEN ;GENERATE ECC
3904 014235 016237 000032 002214 MOV RKECPT(R2),T.ECPT ;GET PATTERN
3905 014244 016237 000030 002212 MOV RKECPS(R2),T.ECPS ;GET POSITION
3906 014252 005237 002350 INC ECPOSX ;BUMP POSITION EXPECTED
3907 014256 013737 002350 002252 MOV ECPOSX,E.ECPS ;SET FOR COMPARE AND REPORT
3908 014264 023737 002254 002214 CMP E.ECPT,T.ECPT ;CHECK IF PATTERN CORRECT
3909 014272 001401 BEQ 15 ;YES - SKIP
3910 014274 104042 ERROR 42
3911 014276 023737 002252 002212 15: CMP E.ECPS,T.ECPS ;CHECK IF POSITION CORRECT
3912 014304 001401 BEQ 25 ;YES - SKIP
3913 014306 104043 ERROR 43
3914 014310 005237 002326 35: INC BITCNT ;BUMP BIT COUNT
3915 014314 005301 DEC R1 ;DEC COUNT TO ZERO IN ECCPOS
3916 014316 001343 BNE 35 ;NOT YET ZERO - LOOP
3917 014320 005300 DEC R0 ;DEC PASS THRU 0 COUNT
3918 014322 001406 BEQ 45 ;IF 0 - SKIP
3919 014324 012737 177777 002350 MOV #-1,ECPOSX ;PRESET EXPECTED POS TO GO TO ZERO
3920 014332 012701 020000 MOV #20000,R1 ;SET COUNT TO NEXT 0 IN ECC POS
3921 014336 000733 BR 35 ;GO DO LOOP AGAIN
3922
3923 014340 005737 001214 45: TST $TMP5 ;TEST SWITCH
3924 014344 001012 BNE 55 ;ALREADY SET - SKIP
3925 014346 012701 010041 MOV #10041,R1 ;SET COUNT TO FINAL POSITION
3926 014352 012700 000001 MOV #1,R0 ;SET R0 TO RETURN AFTER ONE PASS
3927 014356 012737 177777 002350 MOV #-1,ECPOSX ;SET EXP POS TO GO TO 0 ON NEXT BIT
3928 014364 005237 001214 INC $TMP5 ;BUMP SWITCH
3929 014370 000716 BR 35 ;GO DO LOOP AGAIN
3930
3931 014372 004737 033414 55: JSR PC,ROBIT ;ONE MORE BIT TO SET READY
3932 014376 004437 034154 JSR R4,GETREG ;GET '611 REGISTERS
3933 014402 013737 002260 002220 MOV L.CS1,E.CS1 ;GET EXPECTED CS1
3934 014410 052737 100200 002220 BIS #RDY!ERR,E.CS1 ;SET READY AND ERROR
3935 014416 042737 000001 002220 BIC #GO,E.CS1 ;CLEAR GO
3936 014424 023737 002160 002220 CMP T.CS1,E.CS1 ;CHECK IF CS1 CORRECT
3937 014432 001402 BEQ 65 ;YES - SKIP
3938 014434 104044 ERROR 44
3939 014436 000410 BR 75
3940 014440 032737 000100 002174 65: BIT #ECH,T.ER ;TEST IF ECC HARD ERROR SET
3941 014446 001004 BNE 75 ;YES - SKIP
3942 014450 012737 100100 002234 MOV #DCK!ECH,E.ER ;SET EXPECTED ERROR REG
3943 014456 104045 ERROR 45
3944
3945 014460 75:
3946
3947
3948 *****
3949 *TEST 23 READ DATA WITH 23 BIT BURST ERROR
3950 *
3951 * CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
3952 * PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ DATA
  OF 400 WORDS TO AN RK06 IN 26 SECTOR FORMAT,

```

C07

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 90
 DZR6EA.P11 28-SEP-76 15:31 T23 READ DATA WITH 23 BIT BURST ERROR

SEG 0090

*** CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
 AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE. A
 GOOD HEADER, 400 DATA WORDS AND AN ECC INDICATING
 A 23 BIT BURST ERROR. VERIFY THE COUNTING OF THE
 POSITION REGISTER. MAKE SURE DATA CHECK, ECC HARD,
 AND CONTROLLER ERROR SETS.

ST23: SCOPE
 MOV #1, \$TIMES ;DO 1 ITERATION
 CLR BSEDES ;CLEAR BAD SEC DESIRED
 MOV #CCLR, RKCS1(R2) ;CLEAR CONTROLLER
 CLR HIBITS ;CLEAR HI ORDER BITS
 JSR R4, LOADRK ;LOAD "L" REGISTERS
 0 ;CYLINDER 0
 .BYTE 0 ;SECTOR 0
 .BYTE 0 ;TRACK 0
 Ibuff 050440 ;BUFFER ADDRESS Ibuff
 -400 ;WORD COUNT -400
 RDATA ;COMMAND RDATA
 JSR R4, CLPIBF ;GO CLEAR INPUT BUFFER
 JSR R4, BLDDAT ;GO BUILD DATA
 5 ;PATTERN 5
 MOV #530, R0 ;SET INDEX INTO BUFFER
 MOV #126073, Obuff+1000 ;STORE ECC WORDS
 MOV #151052, Obuff+1002
 MOV #1, ECCSRC ;SET SOURCE TO INDICATE ECC IN BUFFER
 BIC #BIT9!BIT11, Obuff(R0) ;CREATE 11 BIT BURST ERROR
 BIS #BIT6!BIT12!BIT14, Obuff(R0)
 ADD #2, R0 ;BUMP TO NEXT WORD
 BIS #BIT0, Obuff(R0)
 BIS #BIT12, Obuff(R0) ;MAKE IT A 23 BIT BURST ERROR
 JSR R4, OPSTRT ;START THE OPERATION
 JSR R4, DISEEK ;SIMULATE IMPLIED SEEK
 ERROR 4 ;CS1 MISCOMPARE
 ERROR 5 ;MR1
 ERROR 6 ;MR2
 ERROR 7 ;MR3
 JSR R4, DSECTR ;SIMULATE SECTOR PULSE
 JSR R4, DRSYNC ;SIMULATE HEADER PREAMBLE
 ERROR 10 ;MR1 CONTENTS IN ERROR
 ERROR 11 ;CS1 CONTENTS IN ERROR
 JSR R4, DHDCMP ;SIMULATE HEADER SEARCH
 ERROR 12 ;MR1 CONTENTS IN ERROR
 ERROR 13 ;CS1 IN ERROR AFTER SEARCH
 ERROR 14 ;RKDCYL OR R4DA IN ERROR AFTER SEARCH

3953
 3954
 3955
 3956
 3957
 3958
 3959
 3960
 3961 014460 000004
 3962 014462 012737 000001 001234
 3963
 3964 014470 005037 002352
 3965 014474 012762 100000 000000
 3966
 3967 014502 005037 002354
 3968 014506 004437 034026
 3969 014512 000000
 3970 014514 000
 3971 014515 000
 3972 014516 050440
 3973 014520 177400
 3974 014522 000021
 3975
 3976 014524 004437 033566
 3977
 3978 014530 004437 033612
 3979 014534 000005
 3980
 3981 014536 012700 000530
 3982
 3983 014542 012737 126073 052444
 3984 014550 012737 151052 052446
 3985 014556 012737 000001 002344
 3986 014564 042760 005000 051444
 3987 014572 052760 050100 051444
 3988 014600 062700 000002
 3989 014604 052760 000001 051444
 3990 014612 052760 010000 051444
 3991
 3992 014620 004437 034072
 3993 014624 004437 025100
 3994 014630 104004
 3995 014632 104005
 3996 014634 104006
 3997 014636 104007
 3998
 3999 014640 004437 025062
 4000
 4001 014644 004437 025504
 4002 014650 104010
 4003 014652 104011
 4004
 4005 014654 004437 026042
 4006 014660 104012
 4007 014662 104013
 4008 014664 104014

E07

RA611: 075A LESS CONTROLLER DIAGNOSTIC: PS MC-11-D2R6E-A MAC111 27(1006) 05-OCT-76 09:12 PAGE 82
 D2R6E-A: 28-SEP-76 15:31 T23 READ DATA WITH 23 BIT BURST ERROR

SEQ 0092

015205	001514	052737	100200	002220	BIS	#RDY!CERR,E.CS1	:SET READY AND ERROR
015206	001515	042737	002001	002220	BIC	#GO,E.CS1	:CLEAR GO
015207	001516	023737	002160	002220	CMP	T.CS1,E.CS1	:CHECK IF CS1 CORRECT
015208	001516	001402			BEO	68	:YES - SKIP
015209	001517	104044			ERROR	44	
015210	001517	000410			BR	78	
015211	001517	032737	000100	002174	BIT	#ECH,T.ER	:TEST IF ECC HARD ERROR SET
015212	001518	001004			BNE	78	:YES - SKIP
015213	001518	002737	100100	002234	MOV	#DCK!ECH,E.ER	:SET EXPECTED ERROR REG
015214	001519	104045			ERROR	45	

78:

 *EST 24 READ DATA WITH 1 BIT ECC WORD ERROR

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
 PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ DATA
 OF 400 WORDS TO AN RK06 IN 26 SECTOR FORMAT.
 CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
 AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE, A
 GOOD HEADER, 400 DATA WORDS AND AN ECC INDICATING
 A 1 BIT ERROR IN ECC WORD 2, BIT 6. VERIFY THE
 COUNTING OF THE POSITION REGISTER. MAKE SURE DATA
 CHECK AND CONTROLLER ERROR SETS.

 *EST 24: SCOPE

015214	000004				MOV	#1,STIMES	:DO 1 ITERATION
015216	012737	000001	001234		CLR	BSEDES	:CLEAR BAD SEC DESIRED
015224	005037	002352			MOV	#CLR,RKCS1(R2)	:CLEAR CONTROLLER
015230	012762	100000	000000		CLR	HIBITS	:CLEAR HI ORDER BITS
015236	005037	002354			JSR	R4,LOADRK	:LOAD "L" REGISTERS
015242	004437	034026			0		:CYLINDER 0
015246	000000				.BYTE	0	:SECTOR 0
015250	000				.BYTE	0	:TRACK 0
015251	000				IBUFF		:BUFFER ADDRESS IBUFF
015252	050440				-400		:WORD COUNT -400
015254	177400				RDDATA		:COMMAND RDDATA
015256	000021						
015260	004437	033566			JSR	R4,CLRIBF	:GO CLEAR INPUT BUFFER
015264	004437	033612			JSR	R4,BLDDAT	:GO BUILD DATA
015270	000005				5		:PATTERN 5
015272	012700	001002			MOV	#1002,R0	:SET INDEX INTO BUFFER
015276	012737	126073	052444		MOV	#126073,0BUFF+1000	:STORE ECC WORDS
015304	012737	151052	052446		MOV	#151052,0BUFF+1002	
015312	012737	000001	002344		MOV	#1,ECCSRC	:SET SOURCE TO INDICATE ECC IN BUFFER
015320	032760	000100	051444		BIT	#BIT6,0BUFF(R0)	:TEST IF BIT SET
015326	001004				BNE	1005	:YES - SKIP
015330	052760	000100	051444		BIS	#BIT6,0BUFF(R0)	:ELSE SET BIT
015336	000403				BR	1015	

F07

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZRB6-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 83
 DZRB6A.P11 28-SEP-76 15:31 T24 READ DATA WITH 1 BIT ECC WORD ERROR

SEG 0083

4121	015340	042760	000100	051444	1005:	BIC	#BIT6,0BUFF(R0)	:CLEAR BIT
4122	015346				1015:			
4123								
4124	015346	004437	034072			JSR	R4,OPSTRT	:START THE OPERATION
4125	015352	004437	025100			JSR	R4,DISEEK	:SIMULATE IMPLIED SEEK
4126	015356	104004				ERROR	4	:CS1 MISCOMPARE
4127	015360	104005				ERROR	5	:MR1
4128	015362	104006				ERROR	6	:MR2
4129	015364	104007				ERROR	7	:MR3
4130								
4131	015366	004437	025062			JSR	R4,DSECTR	:SIMULATE SECTOR PULSE
4132								
4133	015372	004437	025504			JSR	R4,DRSYNC	:SIMULATE HEADER PREAMBLE
4134	015376	104010				ERROR	10	:MR1 CONTENTS IN ERROR
4135	015400	104011				ERROR	11	:CS1 CONTENTS IN ERROR
4136								
4137	015402	004437	026042			JSR	R4,DHDCMP	:SIMULATE HEADER SEARCH
4138	015406	104012				ERROR	12	:MR1 CONTENTS IN ERROR
4139	015410	104013				ERROR	13	:CS1 IN ERROR AFTER SEARCH
4140	015412	104014				ERROR	14	:RKDCYL OR RKDA IN ERROR AFTER SEARCH
4141								
4142	015414	004437	030276			JSR	R4,DRGPSN	:GO READ GAP AND SYNC
4143	015420	104015				ERROR	15	:MR1 IN ERROR READING GAP
4144	015422	104016				ERROR	16	:CS1 IN ERROR AFTER READING GAP
4145	015424	104032				ERROR	32	:MR1 IN ERROR READING SYNC
4146	015426	104033				ERROR	33	:CS1 IN ERROR AFTER READING SYNC
4147								
4148	015430	012737	000001	002346		MOV	#1,ECPATX	:LOAD EXPECTED PATTERN
4149	015436	012737	004066	002350		MOV	#4066,ECP0SX	:LOAD EXPECTED POSITION
4150	015444	004437	030774			JSR	R4,DREAD	:GO SIMULATE DATA READ
4151	015450	000400				400		:NUMBER OF WORDS TO SIMULATE
4152	015452	104034				ERROR	34	:MR1 IN ERROR READING DATA OR ECC
4153	015454	104035				ERROR	35	:ECC ERROR READING DATA
4154	015456	104036				ERROR	36	:CS1 ERROR AFTER READING DATA OR ECC
4155	015460	104041				ERROR	41	:ECC REG INCORRECT AFTER ECC READ
4156	015462	104042				ERROR	42	:ERR IN ECC PAT CALC.
4157	015464	104043				ERROR	43	:ERR IN ECC POS COUNT
4158								
4159								
4160	015466	012737	000020	002326		MOV	#20,BITCNT	:SET BIT COUNT FOR CORRECTION COUNT
4161	015474	012700	000005			MOV	#5,R0	:SET COUNT FOR ECCPOS PASS THRU 0
4162	015500	012701	013672			MOV	#13672,R1	:SET COUNT FOR ECCPOS TO 0 FIRST PASS
4163	015504	005037	001214			CLR	\$TMP5	:CLEAR SWITCH
4164								
4165	015510	004737	033414		35:	JSR	PC,RDBIT	:GO READ BIT
4166	015514	004737	032260			JSR	PC,ECCGEN	:GENERATE ECC
4167	015520	016237	000032	002214		MOV	RKECPT(R2),T.ECPT	:GET PATTERN
4168	015526	016237	000030	002212		MOV	RKECPS(R2),T.ECPS	:GET POSITION
4169	015534	005237	002350			INC	ECPOSX	:BUMP POSITION EXPECTED
4170	015540	013737	002350	002252		MOV	ECPOSX,E.ECPS	:SET FOR COMPARE AND REPORT
4171	015546	023737	002254	002214		CMP	E.ECPT,T.ECPT	:CHECK IF PATTERN CORRECT
4172	015554	001401				BEQ	15	:YES - SKIP
4173	015556	104042				ERROR	42	
4174	015560	023737	002252	002212	15:	CMP	E.ECPS,T.ECPS	:CHECK IF POSITION CORRECT
4175	015566	001401				BEQ	25	:YES - SKIP
4176	015570	104043				ERROR	43	

G07

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 84
 DZR6EA.P11 28-SEP-76 15:31 T24 READ DATA WITH 1 BIT ECC WORD ERROR

SEQ 0094

```

4177 015572 005237 002326      25:  INC  BITCNT      ;BUMP BIT COUNT
4178 015576 005301              DEC  R1          ;DEC COUNT TO ZERO IN ECCPOS
4179 015600 001343              BNE  . 3$        ;NOT YET ZERO - LOOP
4180 015602 005300              DEC  R0          ;DEC PASS THRU 0 COUNT
4181 015604 001406              BEQ  4$          ;IF 0 - SKIP
4182 015606 012737 177777 002350  MOV  #-1,ECP0SX  ;PRESET EXPECTED POS TO GO TO ZERO
4183 015614 012701 020000      MOV  #20000,R1      ;SET COUNT TO NEXT 0 IN ECC POS
4184 015620 000733              BR   3$          ;GO DO LOOP AGAIN
4185
4186 015622 005737 001214      45:  TST  $TMP5      ;TEST SWITCH
4187 015626 001012              BNE  5$          ;ALREADY SET - SKIP
4188 015630 012701 010016      MOV  #10016,R1      ;SET COUNT TO FINAL POSITION
4189 015634 012700 000001      MOV  #1,R0        ;SET R0 TO RETURN AFTER ONE PASS
4190 015640 012737 177777 002350  MOV  #-1,ECP0SX  ;SET EXP POS TO GO TO 0 ON NEXT BIT
4191 015646 005237 001214      INC  $TMP5      ;BUMP SWITCH
4192 015652 000716              BR   3$          ;GO DO LOOP AGAIN
4193
4194 015654 004737 033414      55:  JSR  PC,R0BIT    ;ONE MORE BIT TO SET READY
4195 015660 004437 034154      JSR  R4,GETREG   ;GET '611 REGISTERS
4196 015664 013737 002260 002220  MOV  L.CS1,E.CS1 ;GET EXPECTED CS1
4197 015672 052737 100200 002220  BIS  #RDY!CERR,E.CS1 ;SET READY AND ERROR
4198 015700 042737 000001 002220  BIC  #GO,E.CS1   ;CLEAR GO
4199 015706 023737 002160 002220  CMP  T.CS1,E.CS1 ;CHECK IF CS1 CORRECT
4200 015714 001402              BEQ  6$          ;YES - SKIP
4201 015716 104044              ERROR 44
4202 015720 000410              BR   7$
4203 015722 032737 000100 002174 65:  BIT  #ECH,T.ER    ;TEST IF ECC HARD ERROR SET
4204 015730 001404              BEQ  7$          ;YES - SKIP
4205 015732 012737 100000 002234  MOV  #DCK,E.ER    ;SET EXPECTED ERROR REG
4206 015740 104045              ERROR 45
4207
4208 015742      75:
4209
4210      ;*****
4211      ;*TEST 25      18 BIT NPR WRITING OF MEMORY
4212      ;*
4213      ;*  CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
4214      ;*  PUT CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A READ DATA
4215      ;*  OF 400 WORDS TO AN RK06 IN 24 SECTOR FORMAT,
4216      ;*  CYLINDER 0, HEAD 0, SECTOR 0.  CLOCK THROUGH SEEK
4217      ;*  AND DRIVE CLEAR MESSAGES.  SIMULATE A SECTOR PULSE,
4218      ;*  A GOOD HEADER, 40 DATA WORDS.  VERIFY THAT THE WORDS
4219      ;*  ARE WRITTEN PROPERLY IN MEMORY.
4220      ;*
4221      ;*****
4222 015742 000004      1ST25: SCOPE
4223 015744 012737 000012 001234  MOV  #10.,$TIMES  ;;DO 10. ITERATIONS
4224
4225 015752 005037 002352      CLR  BSEDES      ;CLEAR BAD SEC DESIRED
4226 015756 012762 100000 000000  MOV  #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
4227
4228 015764 005037 002344      CLR  ECCSRC      ;CLEAR ECC SOURCE
4229 015770 005037 002354      CLR  HIBITS      ;CLEAR HI ORDER BITS
4230 015774 004437 034026      JSR  R4,LOADRK   ;LOAD "L" REGISTERS
4231 016000 000000      0          ;CYLINDER 0
4232 016002      000          .BYTE 0          ;SECTOR 0

```

H07

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 85
 DZR6EA.P11 28-SEP-76 15:31 T25 18 BIT NPR WRITING OF MEMORY

SEQ 0095

4233	016003	000		.BYTE 0	;TRACK 0
4234	016004	050440		IBUFF	;BUFFER ADDRESS IBUFF
4235	016006	177400		-400	;WORD COUNT -400
4236	016010	010021		RDDATA!CFMT	;COMMAND RDDATA!CFMT
4237					
4238	016012	004437	033566	JSR R4,CLRIBF	;GO CLEAR INPUT BUFFER
4239					
4240	016016	004437	033612	JSR R4,BLDDAT	;GO BUILD DATA
4241	016022	000003		3	;PATTERN 3
4242					
4243					
4244	016024	004437	034072	JSR R4,OPSTRT	;START THE OPERATION
4245	016030	004437	025100	JSR R4,DISEEK	;SIMULATE IMPLIED SEEK
4246	016034	104004		ERROR 4	;CSI MISCOMPARE
4247	016036	104005		ERROR 5	;MR1 "
4248	016040	104006		ERROR 6	;MR2 "
4249	016042	104007		ERROR 7	;MR3 "
4250					
4251	016044	004437	025062	JSR R4,DSECTR	;SIMULATE SECTOR PULSE
4252					
4253	016050	004437	025504	JSR R4,DRSYNC	;SIMULATE HEADER PREAMBLE
4254	016054	104010		ERROR 10	;MR1 CONTENTS IN ERROR
4255	016056	104011		ERROR 11	;CSI CONTENTS IN ERROR
4256					
4257	016060	004437	026042	JSR R4,DHDCMP	;SIMULATE HEADER SEARCH
4258	016064	104012		ERROR 12	;MR1 CONTENTS IN ERROR
4259	016066	104013		ERROR 13	;CSI IN ERROR AFTER SEARCH
4260	016070	104014		ERROR 14	;RKDCYL OR RKDA IN ERROR AFTER SEARCH
4261					
4262	016072	004437	030276	JSR R4,DRGPSN	;GO READ GAP AND SYNC
4263	016076	104015		ERROR 15	;MR1 IN ERROR READING GAP
4264	016100	104016		ERROR 16	;CSI IN ERROR AFTER READING GAP
4265	016102	104032		ERROR 32	;MR1 IN ERROR READING SYNC
4266	016104	104033		ERROR 33	;CSI IN ERROR AFTER READING SYNC
4267					
4268	016106	012737	000000 002346	MOV #0,ECPATX	;LOAD EXPECTED PATTERN
4269	016114	012737	005066 002350	MOV #5066,ECPOSX	;LOAD EXPECTED POSITION
4270	016122	004437	030774	JSR R4,DREAD	;GO SIMULATE DATA READ
4271	016126	000060		60	;NUMBER OF WORDS TO SIMULATE
4272	016130	104034		ERROR 34	;MR1 IN ERROR READING DATA OR ECC
4273	016132	104035		ERROR 35	;ECC ERROR READING DATA
4274	016134	104036		ERROR 36	;CSI ERROR AFTER READING DATA OR ECC
4275	016136	104041		ERROR 41	;ECC REG INCORRECT AFTER ECC READ
4276	016140	104042		ERROR 42	;ERR IN ECC PAT CALC.
4277	016142	104043		ERROR 43	;ERR IN ECC POS COUNT
4278					
4279					
4280	016144	012700	051444	MOV #0,BUFF,RO	;SET POINTER TO GOOD DATA
4281	016150	012701	050440	MOV #IBUFF,R1	;SET POINTER TO INPUT DATA
4282					
4283	016154	012704	000012	MOV #10,R4	;SET ERROR LIMIT COUNT
4284	016160	005037	001236	CLR \$ESCAPE	;CLEAR ESCAPE
4285	016164	005037	001220	CLR \$TMP7	;CLEAR ERROR REPORT SWITCH
4286	016170	012703	000040	MOV #40,R3	;SET COMPARE LIMIT
4287	016174	005005		CLR R5	;CLEAR WORD COUNTER
4288					

```

4289 016176 021011      1$:  CMP      (R0),(R1)      ;COMPARE DATA
4290 016200 001005      2$:  BNE      2$          ;SKIP IF NOT EQUAL
4291 016202 005303      4$:  DEC      R3          ;ELSE DEC WORD COUNT LIMIT
4292 016204 001424      BEQ      5$          ;IF 0 - SKIP TO EXIT
4293 016206 005205      INC      R5          ;BUMP WORD COUNT
4294 016210 022021      CMP      (R0)+,(R1)+    ;BUMP DATA POINTERS
4295 016212 000771      BR       1$          ;LOOP
4296
4297 016214 005304      2$:  DEC      R4          ;DEC ERROR LIMIT
4298 016216 001417      BEQ      5$          ;IF 0 - SKIP
4299
4300 016220 011037 001162  MOV      (R0),SREG0    ;GET GOOD WORD FOR REPORT
4301 016224 011137 001164  MOV      (R1),SREG1    ;GET BAD WORD
4302 016230 010537 001166  MOV      R5,SREG2    ;GET WORD NUMBER
4303
4304 016234 005737 001220  TST      $TMP7      ;DECIDE WHICH ERROR REPORT TO USE
4305 016240 001004      BNE      3$          ;SKIP IF NOT 0
4306 016242 005237 001220  INC      $TMP7      ;ELSE BUMP $TMP7
4307 016246 104037      ERROR    37          ;REPORT FIRST ERROR LINE
4308 016250 000754      BR       4$          ;SKIP
4309
4310 016252 104040      3$:  ERROR    40          ;REPORT ALL OTHER LINES
4311 016254 000752      BR       4$          ;SKIP
4312
4313 016256      5$:
4314
4315 ;*****
4316 ;*TEST 26      18 BIT READ DATA WITH NO ERROR
4317 ;*
4318 ;*      CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
4319 ;*      PUT CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A READ DATA
4320 ;*      OF 400 WORDS TO AN RK06 IN 24 SECTOR FORMAT,
4321 ;*      CYLINDER 0, HEAD 0, SECTOR 0.  CLOCK THROUGH SEEK
4322 ;*      AND DRIVE CLEAR MESSAGES.  SIMULATE A SECTOR PULSE,
4323 ;*      A GOOD HEADER, 400 DATA WORDS, AND A GOOD 32 BIT ECC.
4324 ;*      VERIFY THAT COMMAND COMPLETES WITHOUT ERROR.
4325 ;*
4326 ;*****
4327 016256 000004      TST26: SCOPE
4328 016260 012737 000001 001234  MOV      #1,$TIMES      ;;DO 1 ITERATION
4329
4330 016266 005037 002352  CLR      BSEDES      ;CLEAR BAD SEC DESIRED
4331 016272 012762 100000 000000  MOV      #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
4332 016300 005037 002344      CLR      ECCSRC      ;CLEAR ECC SOURCE FLAG
4333 016304 005037 002344      CLR      ECCSRC      ;CLEAR ECC SOURCE
4334 016310 005037 002354      CLR      HIBITS      ;CLEAR HI ORDER BITS
4335 016314 004437 034026  JSR      R4,LOADRK    ;LOAD "L" REGISTERS
4336 016320 000000      0          ;CYLINDER 0
4337 016322      000          .BYTE    0          ;SECTOR 0
4338 016323      000          .BYTE    0          ;TRACK 0
4339 016324 050440      Ibuff      ;BUFFER ADDRESS Ibuff
4340 016326 177400      -400        ;WORD COUNT -400
4341 016330 010021      Rddata!CFMT ;COMMAND Rddata!CFMT
4342
4343 016332 004437 033566  JSR      R4,CLRIBF      ;GO CLEAR INPUT BUFFER
4344

```

J07

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 87
 DZR6EA.P11 28-SEP-76 15:31 T26 18 BIT READ DATA WITH NO ERROR

SEQ 0097

```

4345 016336 004437 033612      JSR      R4,BLDDAT      ;GO BUILD DATA
4346 016342 000004              4      ;PATTERN 4
4347
4348
4349 016344 004437 034072      JSR      R4,OPSTRT      ;START THE OPERATION
4350 016350 004437 025100      JSR      R4,DISEEK      ;SIMULATE IMPLIED SEEK
4351 016354 104004              ERROR 4      ;CS1 MISCOMPARE
4352 016356 104005              ERROR 5      ;MR1
4353 016360 104006              ERROR 6      ;MR2
4354 016362 104007              ERROR 7      ;MR3
4355
4356 016364 004437 025762      JSR      R4,DSECTR      ;SIMULATE SECTOR PULSE
4357
4358 016370 004437 025504      JSR      R4,DRSYNC      ;SIMULATE HEADER PREAMBLE
4359 016374 104010              ERROR 10     ;MR1 CONTENTS IN ERROR
4360 016376 104011              ERROR 11     ;CS1 CONTENTS IN ERROR
4361
4362 016400 004437 026042      JSR      R4,DHDCMP      ;SIMULATE HEADER SEARCH
4363 016404 104012              ERROR 12     ;MR1 CONTENTS IN ERROR
4364 016406 104013              ERROR 13     ;CS1 IN ERROR AFTER SEARCH
4365 016410 104014              ERROR 14     ;RKDCYL OR RKDA IN ERROR AFTER SEARCH
4366
4367 016412 004437 030276      JSR      R4,DRGPSN      ;GO READ GAP AND SYNC
4368 016416 104015              ERROR 15     ;MR1 IN ERROR READING GAP
4369 016420 104016              ERROR 16     ;CS1 IN ERROR AFTER READING GAP
4370 016422 104032              ERROR 32     ;MR1 IN ERROR READING SYNC
4371 016424 104033              ERROR 33     ;CS1 IN ERROR AFTER READING SYNC
4372
4373 016426 012737 000000 002346  MOV      #0,ECPATX      ;LOAD EXPECTED PATTERN
4374 016434 012737 005066 002350  MOV      #5066,ECPOSX     ;LOAD EXPECTED POSITION
4375 016442 004437 030774              JSR      R4,DREAD      ;GO SIMULATE DATA READ
4376 016446 000400              400      ;NUMBER OF WORDS TO SIMULATE
4377 016450 104034              ERROR 34     ;MR1 IN ERROR READING DATA OR ECC
4378 016452 104035              ERROR 35     ;ECC ERROR READING DATA
4379 016454 104036              ERROR 36     ;CS1 ERROR AFTER READING DATA OR ECC
4380 016456 104041              ERROR 41     ;ECC REG INCORRECT AFTER ECC READ
4381 016460 104042              ERROR 42     ;ERR IN ECC PAT CALC.
4382 016462 104043              ERROR 43     ;ERR IN ECC POS COUNT
4383
4384
4385 016464 012700 003100      MOV      #40.*40.,R0      ;SET COUNT TO EMPTY SILO
4386
4387 016470 012762 000440 000026 1S:  MOV      #DMD!MCLK,RKMR1(R2) ;CLOCK CONTROLLER
4388 016476 012762 000040 000026      MOV      #DMD,RKMR1(R2)
4389 016504 005300              DEC      R0      ;DEC COUNT
4390 016506 001370              BNE      1S      ;LOOP UNTIL 0
4391
4392 016510 004437 034154      JSR      R4,GETREG      ;GET 611 REGS
4393 016514 013737 002260 002220      MOV      L.CS1,E.CS1      ;GET EXPECTED CS1
4394 016522 042737 000001 002220      BIC      #GO,E.CS1      ;CLEAR GO BIT
4395 016530 052737 000200 002220      BIS      #RDY,E.CS1      ;SET READY BIT
4396 016536 023737 002160 002220      CMP      T.CS1,E.CS1      ;CHECK IF CS1 CORRECT
4397 016544 001401              BEQ      2S      ;YES - SKIP
4398 016546 104056              ERROR 56     ;"CS1 IN ERROR AFTER READ DATA"
4399 016550
4400
2S:
;*****

```

K07

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 88
 DZR6EA.P11 28-SEP-76 15:31 T27 18 BIT READ DATA WITH DCK

SEG 0088

```

4401 ;*TEST 27      18 BIT READ DATA WITH DCK
4402 ;
4403 ;
4404 ;
4405 ;
4406 ;
4407 ;
4408 ;
4409 ;
4410 ;
4411 ;
4412 ;
4413 ;
4414 ;
4415 ;
4416 ;
4417 ;
4418 ;
4419 ;
4420 ;
4421 ;
4422 ;
4423 ;
4424 ;
4425 ;
4426 ;
4427 ;
4428 ;
4429 ;
4430 ;
4431 ;
4432 ;
4433 ;
4434 ;
4435 ;
4436 ;
4437 ;
4438 ;
4439 ;
4440 ;
4441 ;
4442 ;
4443 ;
4444 ;
4445 ;
4446 ;
4447 ;
4448 ;
4449 ;
4450 ;
4451 ;
4452 ;
4453 ;
4454 ;
4455 ;
4456 ;

```

016550	000004				
016552	012737	000001	001234		
016560	005037	002352			
016564	012762	100000	000000		
016572	005037	002354			
016576	004437	034026			
016602	000000				
016604	000				
016605	000				
016606	050440				
016610	177400				
016612	010021				
016614	004437	033566			
016620	004437	033612			
016624	000007				
016626	012700	000734			
016632	012737	006030	052444		
016640	012737	045070	052446		
016646	012737	000001	002344		
016654	032760	000100	051444		
016662	001004				
016664	052760	000100	051444		
016672	000403				
016674	042760	000100	051444	100\$:	
016702				101\$:	
016702	004437	034072			
016706	004437	025100			
016712	104004				
016714	104005				
016716	104006				
016720	104007				
016722	004437	025062			
016726	004437	025504			

```

*****
TST27: SCOPE
MOV      #1,$TIMES      ;;DO 1 ITERATION
CLR      BSEDES          ;CLEAR BAD SEC DESIRED
MOV      #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
CLR      HIBITS          ;CLEAR HI ORDER BITS
JSR      R4,LOADRK       ;LOAD "L" REGISTERS
0        ;CYLINDER 0
.BYTE    0               ;SECTOR 0
.BYTE    0               ;TRACK 0
IBUFF    ;BUFFER ADDRESS IBUFF
-400     ;WORD COUNT -400
RDDATA!CFMT              ;COMMAND RDDATA!CFMT
JSR      R4,CLRIBF       ;GO CLEAR INPUT BUFFER
JSR      R4,BLDDAT       ;GO BUILD DATA
7        ;PATTERN 7
MOV      #734,R0         ;SET INDEX INTO BUFFER
MOV      #6030,0BUFF+1000 ;STORE ECC WORDS
MOV      #45070,0BUFF+1002
MOV      #1,ECCSRC       ;SET SOURCE TO INDICATE ECC IN BUFFER
BIT      #BIT6,0BUFF(R0) ;TEST IF BIT SET
BNE      100$            ;YES - SKIP
BIS      #BIT6,0BUFF(R0) ;ELSE SET BIT
BR       101$
BIC      #BIT6,0BUFF(R0) ;CLEAR BIT
100$:
101$:
JSR      R4,OPSTRT       ;START THE OPERATION
JSR      R4,DISEEK       ;SIMULATE IMPLIED SEEK
ERROR    4              ;CS1 MISCOMPARE
ERROR    5              ;MR1
ERROR    6              ;MR2
ERROR    7              ;MR3
JSR      R4,DSECTR       ;SIMULATE SECTOR PULSE
JSR      R4,DRSYNC       ;SIMULATE HEADER PREAMBLE

```

L07

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 89
 DZR6EA.P11 28-SEP-76 15:31 T27 18 BIT READ DATA WITH DCK

SEQ 0089

4457	016732	104010			ERROR	10		;MR1 CONTENTS IN ERROR
4458	016734	104011			ERROR	11		;CS1 CONTENTS IN ERROR
4459								
4460	016736	004437	026042		JSR	R4,DHDCMP		;SIMULATE HEADER SEARCH
4461	016742	104012			ERROR	12		;MR1 CONTENTS IN ERROR
4462	016744	104013			ERROR	13		;CS1 IN ERROR AFTER SEARCH
4463	016746	104014			ERROR	14		;RKDCYL OR RKDA IN ERROR AFTER SEARCH
4464								
4465	016750	004437	030276		JSR	R4,DRGPSN		;GO READ GAP AND SYNC
4466	016754	104015			ERROR	15		;MR1 IN ERROR READING GAP
4467	016756	104016			ERROR	16		;CS1 IN ERROR AFTER READING GAP
4468	016760	104032			ERROR	32		;MR1 IN ERROR READING SYNC
4469	016762	104033			ERROR	33		;CS1 IN ERROR AFTER READING SYNC
4470								
4471	016764	012737	000001	002346	MOV	#1,ECPATX		;LOAD EXPECTED PATTERN
4472	016772	012737	005066	002350	MOV	#5066,ECPOSX		;LOAD EXPECTED POSITION
4473	017000	004437	030774		JSR	R4,DRDREAD		;GO SIMULATE DATA READ
4474	017004	000400			400			;NUMBER OF WORDS TO SIMULATE
4475	017006	104034			ERROR	34		;MR1 IN ERROR READING DATA OR ECC
4476	017010	104035			ERROR	35		;ECC ERROR READING DATA
4477	017012	104036			ERROR	36		;CS1 ERROR AFTER READING DATA OR ECC
4478	017014	104041			ERROR	41		;ECC REG INCORRECT AFTER ECC READ
4479	017016	104042			ERROR	42		;ERR IN ECC PAT CALC.
4480	017020	104043			ERROR	43		;ERR IN ECC POS COUNT
4481								
4482								
4483	017022	012737	000022	002326	MOV	#22,BITCNT		;SET BIT COUNT FOR CORRECTION COUNT
4484	017030	012700	000005		MOV	#5,R0		;SET COUNT FOR ECCPOS PASS THRU 0
4485	017034	012701	012672		MOV	#12672,R1		;SET COUNT FOR ECCPOS TO 0 FIRST PASS
4486	017040	005037	001214		CLR	\$TMP5		;CLEAR SWITCH
4487								
4488	017044	004737	033414		JSR	PC,RDBIT		;GO READ BIT
4489	017050	004737	032260		JSR	PC,ECCGEN		;GENERATE ECC
4490	017054	016237	000032	002214	MOV	RKECPT(R2),T.ECPT		;GET PATTERN
4491	017062	016237	000030	002212	MOV	RKECPS(R2),T.ECPS		;GET POSITION
4492	017070	005237	002350		INC	ECPOSX		;BUMP POSITION EXPECTED
4493	017074	013737	002350	002252	MOV	ECPOSX,E.ECFS		;SET FOR COMPARE AND REPORT
4494	017102	023737	002254	002214	CMP	E.ECPT,T.ECPT		;CHECK IF PATTERN CORRECT
4495	017110	001401			BEQ	1\$		;YES - SKIP
4496	017112	104042			ERROR	42		
4497	017114	023737	002252	002212	1\$: CMP	E.ECPS,T.ECPS		;CHECK IF POSITION CORRECT
4498	017122	001401			BEQ	2\$		;YES - SKIP
4499	017124	104043			ERROR	43		
4500	017126	005237	002326		2\$: INC	BITCNT		;BUMP BIT COUNT
4501	017132	005301			DEC	R1		;DEC COUNT TO ZERO IN ECCPOS
4502	017134	001343			BNE	3\$		;NOT YET ZERO - LOOP
4503	017136	005300			DEC	R0		;DEC PASS THRU 0 COUNT
4504	017140	001406			BEQ	4\$		;IF 0 - SKIP
4505	017142	012737	177777	002350	MOV	#-1,ECPOSX		;PRESET EXPECTED POS TO GO TO ZERO
4506	017150	012701	020000		MOV	#20000,R1		;SET COUNT TO NEXT 0 IN ECC POS
4507	017154	000733			BR	3\$		;GO DO LOOP AGAIN
4508								
4509	017156	005737	001214		4\$: TST	\$TMP5		;TEST SWITCH
4510	017162	001012			BNE	5\$		;ALREADY SET - SKIP
4511	017164	012701	010272		MOV	#10272,R1		;SET COUNT TO FINAL POSITION
4512	017170	012700	000001		MOV	#1,R0		;SET R0 TO RETURN AFTER ONE PASS

M07

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 90
 DZR6EA.P11 28-SEP-76 15:31 T27 18 BIT READ DATA WITH DCK

SEQ 0090

```

4513 017174 012737 177777 002350      MOV    #1,ECPOSX      ;SET EXP POS TO GO TO 0 ON NEXT BIT
4514 017202 005237 001214              INC    $TMP5        ;BUMP SWITCH
4515 017206 000716                  BR      3$           ;GO DO LOOP AGAIN
4516
4517 017210 004737 033414          5$:   JSR    PC,RDBIT      ;ONE MORE BIT TO SET READY
4518 017214 004437 034154          JSR    R4,GETREG      ;GET '611 REGISTERS
4519 017220 013737 002260 002220      MOV    L.CS1,E.CS1    ;GET EXPECTED CS1
4520 017226 052737 100200 002220      BIS    #RDY!CERR,E.CS1 ;SET READY AND ERROR
4521 017234 042737 000001 002220      BIC    #GO,E.CS1      ;CLEAR GO
4522 017242 023737 002160 002220      CMP    T.CS1,E.CS1    ;CHECK IF CS1 CORRECT
4523 017250 001402                  BEQ     6$           ;YES - SKIP
4524 017252 104044                  ERROR  44
4525 017254 000410                  BR      7$
4526 017256 032737 000100 002174 6$:   BIT     #ECH,T.ER      ;TEST IF ECC HARD ERROR SET
4527 017264 001404                  BEQ     7$           ;NO SKIP
4528 017266 012737 100000 002234      MOV    #DCK,E.ER      ;SET EXPECTED ERROR REG
4529 017274 104045                  ERROR  45
4530
4531 017276          7$:
4532
4533      ;*****
4534      ;TEST 30      18 BIT READ DATA WITH ECH
4535      ;
4536      ;      CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
4537      ;      PUT CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A READ DATA
4538      ;      OF 400 WORDS TO AN RK06 IN 24 SECTOR FORMAT,
4539      ;      CYLINDER 0, HEAD 0, SECTOR 0.  CLOCK THROUGH SEEK
4540      ;      AND DRIVE CLEAR MESSAGES.  SIMULATE A SECTOR PULSE,
4541      ;      A GOOD HEADER, 400 DATA WORDS AND AND ECC INDICATING
4542      ;      A 12 BIT BURST ERROR.  VERIFY THE COUNTING OF THE
4543      ;      POSITION REGISTER.  MAKE SURE DATE CHECK, ECC HARD,
4544      ;      AND CONTROLLER ERROR SETS.
4545      ;*****
4546
4547 017276 000004      ST30:  SCOPE
4548 017300 012737 000001 001234      MOV    #1,$TIMES      ;;DO 1 ITERATION
4549
4550 017306 005037 002352          CLR    BSEDES      ;CLEAR BAD SEC DESIRED
4551 017312 012762 100000 000000      MOV    #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
4552
4553 017320 005037 002354          CLR    HIBITS      ;CLEAR HI ORDER BITS
4554 017324 004437 034026      JSR    R4,LOADRK      ;LOAD "L" REGISTERS
4555 017330 000000          0              ;CYLINDER 0
4556 017332 000          .BYTE  0              ;SECTOR 0
4557 017333 000          .BYTE  0              ;TRACK 0
4558 017334 050440          Ibuff      ;BUFFER ADDRESS Ibuff
4559 017336 177400          -400          ;WORD COUNT -400
4560 017340 010021      Rddata!CFMT      ;COMMAND Rddata!CFMT
4561
4562 017342 004437 033566          JSR    R4,CLRIBF      ;GO CLEAR INPUT BUFFER
4563
4564 017346 004437 033612          JSR    R4,BLDDAT      ;GO BUILD DATA
4565 017352 000005          5              ;PATTERN 5
4566
4567
4568 017354 012700 000734          MOV    #734,R0      ;SET INDEX INTO BUFFER

```

NO7

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 91
DZR6EA.P11 28-SEP-76 15:31 T30 18 BIT READ DATA WITH ECH

SEQ 0091

4569	017360	012737	004020	052444	MOV	#4020,0BUFF+1000	;STORE ECC WORDS
4570	017366	012737	071720	052446	MOV	#71720,0BUFF+1002	
4571	017374	012737	000001	002344	MOV	#1,ECCSRC	;SET SOURCE TO INDICATE ECC IN BUFFER
4572	017402	042760	005000	051444	BIC	#BIT9!BIT11,0BUFF(R0)	;CREATE 11 BIT BURST ERROR
4573	017410	052760	050100	051444	BIS	#BIT6!BIT12!BIT14,0BUFF(R0)	
4574	017416	062700	000002		ADD	#2,R0	;BUMP TO NEXT WORD
4575	017422	052760	000001	051444	BIS	#BIT0,0BUFF(R0)	
4576	017430	042760	000002	051444	BIC	#BIT1,0BUFF(R0)	;MAKE IT A 12 BIT BURST ERROR
4577							
4578	017436	004437	034072		JSR	R4,OPSTRT	;START THE OPERATION
4579	017442	004437	025100		JSR	R4,DISEEK	;SIMULATE IMPLIED SEEK
4580	017446	104004			ERROR	4	;CS1 MISCOMPARE
4581	017450	104005			ERROR	5	;MR1 "
4582	017452	104006			ERROR	6	;MR2 "
4583	017454	104007			ERROR	7	;MR3 "
4584							
4585	017456	004437	025062		JSR	R4,DSECTR	;SIMULATE SECTOR PULSE
4586							
4587	017462	004437	025504		JSR	R4,DRSYNC	;SIMULATE HEADER PREAMBLE
4588	017466	104010			ERROR	10	;MR1 CONTENTS IN ERROR
4589	017470	104011			ERROR	11	;CS1 CONTENTS IN ERROR
4590							
4591	017472	004437	026042		JSR	R4,DHDCMP	;SIMULATE HEADER SEARCH
4592	017476	104012			ERROR	12	;MR1 CONTENTS IN ERROR
4593	017500	104013			ERROR	13	;CS1 IN ERROR AFTER SEARCH
4594	017502	104014			ERROR	14	;RKDCYL OR RKDA IN ERROR AFTER SEARCH
4595							
4596	017504	004437	030276		JSR	R4,DRGPSN	;GO READ GAP AND SYNC
4597	017510	104015			ERROR	15	;MR1 IN ERROR READING GAP
4598	017512	104016			ERROR	16	;CS1 IN ERROR AFTER READING GAP
4599	017514	104032			ERROR	32	;MR1 IN ERROR READING SYNC
4600	017516	104033			ERROR	33	;CS1 IN ERROR AFTER READING SYNC
4601							
4602	017520	012737	000001	002346	MOV	#1,ECPATX	;LOAD EXPECTED PATTERN
4603	017526	012737	005066	002350	MOV	#5066,ECPOSX	;LOAD EXPECTED POSITION
4604	017534	004437	030774		JSR	R4,DREAD	;GO SIMULATE DATA READ
4605	017540	000400			400		;NUMBER OF WORDS TO SIMULATE
4606	017542	104034			ERROR	34	;MR1 IN ERROR READING DATA OR ECC
4607	017544	104035			ERROR	35	;ECC ERROR READING DATA
4608	017546	104036			ERROR	36	;CS1 ERROR AFTER READING DATA OR ECC
4609	017550	104041			ERROR	41	;ECC REG INCORRECT AFTER ECC READ
4610	017552	104042			ERROR	42	;ERR IN ECC PAT CALC.
4611	017554	104043			ERROR	43	;ERR IN ECC POS COUNT
4612							
4613							
4614	017556	012737	000022	002326	MOV	#22,BITCNT	;SET BIT COUNT FOR CORRECTION COUNT
4615	017564	012700	000005		MOV	#5,R0	;SET COUNT FOR ECCPOS PASS THRU 0
4616	017570	012701	012672		MOV	#12672,R1	;SET COUNT FOR ECCPOS TO 0 FIRST PASS
4617	017574	005037	001214		CLR	\$TMP5	;CLEAR SWITCH
4618							
4619	017600	004737	033414		JSR	PC,RDBIT	;GO READ BIT
4620	017604	004737	032260		JSR	PC,ECCGEN	;GENERATE ECC
4621	017610	016237	000032	002214	MOV	RKECPT(R2),T.ECPT	;GET PATTERN
4622	017616	016237	000030	002212	MOV	RKECPS(R2),T.ECPS	;GET POSITION
4623	017624	005237	002350		INC	ECPOSX	;BUMP POSITION EXPECTED
4624	017630	013737	002350	002252	MOV	ECPOSX,E.ECPS	;SET FOR COMPARE AND REPORT

3\$:

```

4625 017636 023737 002254 002214      CMP      E.ECPT,T.ECPT      :CHECK IF PATTERN CORRECT
4626 017644 001401                      BEQ       1$              :YES - SKIP
4627 017646 104042                      ERROR    42
4628 017650 023737 002252 002212 1$:    CMP      E.ECPS,T.ECPS      :CHECK IF POSITION CORRECT
4629 017656 001401                      BEQ       2$              :YES - SKIP
4630 017660 104043                      ERROR    43
4631 017662 005237 002326      2$:      INC      BITCNT          :BUMP BIT COUNT
4632 017666 005301                      DEC      R1              :DEC COUNT TO ZERO IN ECCPOS
4633 017670 001343                      BNE      3$              :NOT YET ZERO - LOOP
4634 017672 005300                      DEC      R0              :DEC PASS THRU 0 COUNT
4635 017674 001406                      BEQ       4$              :IF 0 - SKIP
4636 017676 012737 177777 002350      MOV      #-1,ECP0SX      :PRESET EXPECTED POS TO GO TO ZERO
4637 017704 012701 020000      MOV      #20000,R1      :SET COUNT TO NEXT 0 IN ECC POS
4638 017710 000733                      BR        3$              :GO DO LOOP AGAIN
4639
4640 017712 005737 001214      4$:      TST      $TMP5          :TEST SWITCH
4641 017716 001012                      BNE      5$              :ALREADY SET - SKIP
4642 017720 012701 011041      MOV      #11041,R1      :SET COUNT TO FINAL POSITION
4643 017724 012700 000001      MOV      #1,R0        :SET R0 TO RETURN AFTER ONE PASS
4644 017730 012737 177777 002350      MOV      #-1,ECP0SX      :SET EXP POS TO GO TO 0 ON NEXT BIT
4645 017736 005237 001214      INC      $TMP5          :BUMP SWITCH
4646 017742 000716                      BR        3$              :GO DO LOOP AGAIN
4647
4648 017744 004737 033414      5$:      JSR      PC,RDBIT      :ONE MORE BIT TO SET READY
4649 017750 004437 034154      JSR      R4,GETREG      :GET '611 REGISTERS
4650 017754 013737 002260 002220      MOV      L.CS1,E.CS1    :GET EXPECTED CS1
4651 017762 052737 100200 002220      BIS      #RDY!CERR,E.CS1 :SET READY AND ERROR
4652 017770 042737 000001 002220      BIC      #GO,E.CS1      :CLEAR GO
4653 017776 023737 002160 002220      CMP      T.CS1,E.CS1    :CHECK IF CS1 CORRECT
4654 020004 001402                      BEQ       6$              :YES - SKIP
4655 020006 104044                      ERROR    44
4656 020010 000410                      BR        7$
4657 020012 032737 000100 002174 6$:    BIT      #ECH,T.ER      :TEST IF ECC HARD ERROR SET
4658 020020 001004                      BNE      7$              :YES - SKIP
4659 020022 012737 100100 002234      MOV      #DCK!ECH,E.ER    :SET EXPECTED ERROR REG
4660 020030 104045                      ERROR    45

```

4661
4662 020032

7\$:

.SBTTL **SPECIAL READ TESTS

```

*****
:TEST 31      PARTIAL SECTOR READ
:
:      CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
:      PUT CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A READ DATA
:      OF 103 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
:      CYLINDER 0, HEAD 0, SECTOR 0.  CLOCK THROUGH SEEK
:      AND DRIVE CLEAR MESSAGES.  SIMULATE A SECTOR PULSE,
:      A GOOD HEADER, 400 DATA WORDS, AND A GOOD 32 BIT ECC.
:      MAKE SURE ONLY 103 WORDS GOT TRANSFERRED TO MEMORY.
*****

```

```

4678 020032 000004      ST31:  SCOPE
4679 020034 012737 000001 001234      MOV      #1,$TIMES      ::DO 1 ITERATION
4680

```

C08

 RK511 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZRG6-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 93
 DZRG6A.P11 28-SEP-76 15:31 T31 PARTIAL SECTOR READ

SEQ 0093

4681	020042	005037	002352	CLR	BSEDES	:CLEAR BAD SEC DESIRED
4682	020046	012762	100000 000000	MOV	#0CLR,RKCS1,R2	:CLEAR CONTROLLER
4683						
4684	020054	005037	002344	CLR	ECCSRC	:CLEAR ECC SOURCE
4685	020060	005037	002354	CLR	HIBITS	:CLEAR HI ORDER BITS
4686	020064	004437	034026	JSR	R4,LOADRK	:LOAD "L" REGISTERS
4687	020070	000000		0		:CYLINDER 0
4688	020072	000		.BYTE	0	:SECTOR 0
4689	020073	000		.BYTE	0	:TRACK 0
4690	020074	050440		IBUFF		:BUFFER ADDRESS IBUFF
4691	020076	177675		-103		:WORD COUNT -103
4692	020100	000021		RDDATA		:COMMAND RDDATA
4693						
4694	020102	004437	033566	JSR	R4,CLRIBF	:GO CLEAR INPLT BUFFER
4695						
4696	020106	004437	033612	JSR	R4,BLDDAT	:GO BUILD DATA
4697	020112	000006		6		:PATTERN 6
4698						
4699						
4700	020114	004437	034072	JSR	R4,OPSTAT	:START THE OPERATION
4701	020120	004437	025100	JSR	R4,DISEEK	:SIMULATE IMPLIED SEEK
4702	020124	104004		ERROR	4	:CSI MISCOMPARE
4703	020126	104005		ERROR	5	:MR1
4704	020130	104006		ERROR	6	:MR2
4705	020132	104007		ERROR	7	:MR3
4706						
4707	020134	004437	025062	JSR	R4,DSECTR	:SIMULATE SECTOR PULSE
4708						
4709	020140	004437	025504	JSR	R4,DRSYNC	:SIMULATE HEADER PREAMBLE
4710	020144	104010		ERROR	10	:MR1 CONTENTS IN ERROR
4711	020146	104011		ERROR	11	:CSI CONTENTS IN ERROR
4712						
4713	020150	004437	026042	JSR	R4,DHDCMP	:SIMULATE HEADER SEARCH
4714	020154	104012		ERROR	12	:MR1 CONTENTS IN ERROR
4715	020156	104013		ERROR	13	:CSI IN ERROR AFTER SEARCH
4716	020160	104014		ERROR	14	:RKDCYL OR RKDA IN ERROR AFTER SEARCH
4717						
4718	020162	004437	030276	JSR	R4,DRGPSN	:GO READ GAP AND SYNC
4719	020166	104015		ERROR	15	:MR1 IN ERROR READING GAP
4720	020170	104016		ERROR	16	:CSI IN ERROR AFTER READING GAP
4721	020172	104032		ERROR	32	:MR1 IN ERROR READING SYNC
4722	020174	104033		ERROR	33	:CSI IN ERROR AFTER READING SYNC
4723						
4724	020176	012737	000000 002346	MOV	#0,ECPATX	:LOAD EXPECTED PATTERN
4725	020204	012737	004066 002350	MOV	#4066,ECP0SX	:LOAD EXPECTED POSITION
4726	020212	004437	030774	JSR	R4,DREAD	:GO SIMULATE DATA READ
4727	020216	000400		400		:NUMBER OF WORDS TO SIMULATE
4728	020220	104034		ERROR	34	:MR1 IN ERROR READING DATA OR ECC
4729	020222	104035		ERROR	35	:ECC ERROR READING DATA
4730	020224	104036		ERROR	36	:CSI ERROR AFTER READING DATA OR ECC
4731	020226	104041		ERROR	41	:ECC REG INCORRECT AFTER ECC READ
4732	020230	104042		ERROR	42	:ERR IN ECC PAT CALC.
4733	020232	104043		ERROR	43	:ERR IN ECC POS COUNT
4734						
4735						
4736	020234	013700	002164	MOV	T.BA,R0	:GET ACTUAL BA

D08

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZ6E-A MACY11 27 (1006) 05-OCT-76 09:12 PAGE 94
 DZ6EA.P11 28-SEP-76 15:31 T31 PARTIAL SECTOR READ

SEG 009-

```

4737 020240 162700 050440 SUB      #IBUFF,R0      ;SUB START VALUE OF BA
4738 020244 022700 000206 CMP      #206,R0      ;TEST IF CORRECT NUM OF WORDS XFER
4739 020250 001404 BEQ      48          ;YES - SKIP
4740 020252 012737 050646 002224 MOV      #IBUFF+206,E.BA ;SET EXPECTED BA
4741 020260 104046 ERROR    46          ;REPORT XFER LENGTH ERROR
4742
4743 020262 012701 000014 48:      MOV      #3,*4,R1      ;SET COUNT TO END OF OPERATION
4744 020266 012762 000440 000026 38:      MOV      #0MD!MCLK,RKMR1(R2) ;RUN THE CLOCK
4745 020274 012762 000040 000026 MOV      #0MD,RKMR1(R2)
4746 020302 005301 DEC      R1          ;DEC THE COUNT
4747 020304 001370 BNE      38          ;LOOP UNTIL ZERO
4748
4749 020306 004437 034154 18:      JSR      R4,GETREG      ;GET 611 REGS
4750 020312 013737 002260 002220 MOV      L,CS1,E.CS1    ;GET EXPECTED CS1
4751 020320 042737 000001 002220 BIC      #GO,E.CS1      ;CLEAR GO BIT
4752 020326 052737 000200 002220 BIS      #RDY,E.CS1      ;SET READY BIT
4753 020334 023737 002160 002220 CMP      T,CS1,E.CS1    ;CHECK IF CS1 CORRECT
4754 020342 001401 BEQ      28          ;YES - SKIP
4755 020344 104056 ERROR    56          ;"CS1 IN ERROR AFTER READ DATA"
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773 020346 000004 *****
4774 020350 012737 000001 001234 TEST 32      SILO UNLOAD BEFORE HEADER ERROR
4775
4776 020356 012762 100000 000000
4777
4778 020364 005037 002352
4779 020370 005037 002344
4780 020374 005037 002354
4781 020400 004437 034026
4782 020404 000000
4783 020406 000
4784 020407 000
4785 020410 050440
4786 020412 177377
4787 020414 000021
4788
4789 020416 004437 033566
4790
4791 020422 004437 033612
4792 020426 000002

```

 TEST 32: SCOPE
 MOV #1,STIMES ;DO 1 ITERATION
 MOV #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
 CLR BSEDES ;CLEAR BSE DESIRED
 CLR ECCSRC ;CLEAR ECC SOURCE
 CLR HIBITS ;CLEAR HI ORDER BITS
 JSR R4,LOADRK ;LOAD "L" REGISTERS
 0 ;CYLINDER 0
 .BYTE 0 ;SECTOR 0
 .BYTE 0 ;TRACK 0
 IBUFF ;BUFFER ADDRESS IBUFF
 -401 ;WORD COUNT -401
 RDATA ;COMMAND RDATA
 JSR R4,CLRIBF ;GO CLEAR INPUT BUFFER
 JSR R4,BLDDAT ;GO BUILD DATA
 2 ;PATTERN 2

E08

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MC-11-DZ6E-A MAC11 27(1006) 05-OCT-76 09:12 PAGE 95
 DZ6EA.P11 28-SEP-76 15:31 T32 SILO UNLOAD BEFORE HEADER ERROR

SEG 0095

4793								
4794								
4795	020430	004437	034072		JSR	R4,OPSTRT		:START THE OPERATION
4796	020434	004437	025100		JSR	R4,DISEEK		:SIMULATE IMPLIED SEEK
4797	020441	104004			ERROR	4		:CSI MISCOMPARE
4798	020442	104005			ERROR	5		:MR1
4799	020444	104006			ERROR	6		:MR2
4800	020446	104007			ERROR	7		:MR3
4801								
4802	020450	004437	025062		JSR	R4,DSECTR		:SIMULATE SECTOR PULSE
4803								
4804	020454	004437	025504		JSR	R4,DRSYNC		:SIMULATE HEADER PREAMBLE
4805	020460	104010			ERROR	10		:MR1 CONTENTS IN ERROR
4806	020462	104011			ERROR	11		:CSI CONTENTS IN ERROR
4807								
4808	020464	004437	026042		JSR	R4,DHDCMP		:SIMULATE HEADER SEARCH
4809	020470	104012			ERROR	12		:MR1 CONTENTS IN ERROR
4810	020472	104013			ERROR	13		:CSI IN ERROR AFTER SEARCH
4811	020474	104014			ERROR	14		:RKDCYL OR RKDA IN ERROR AFTER SEARCH
4812								
4813	020476	004437	030276		JSR	R4,DRGPSN		:GO READ GAP AND SYNC
4814	020502	104015			ERROR	15		:MR1 IN ERROR READING GAP
4815	020504	104016			ERROR	16		:CSI IN ERROR AFTER READING GAP
4816	020506	104032			ERROR	32		:MR1 IN ERROR READING SYNC
4817	020510	104033			ERROR	33		:CSI IN ERROR AFTER READING SYNC
4818								
4819	020512	012737	000000	002346	MOV	#0,ECPATX		:LOAD EXPECTED PATTERN
4820	020520	012737	004066	002350	MOV	#4066,ECPOSX		:LOAD EXPECTED POSITION
4821	020526	004437	030774		JSR	R4,DREAD		:DO SIMULATE DATA READ
4822	020532	000400			400			:NUMBER OF WORDS TO SIMULATE
4823	020534	104034			ERROR	34		:MR1 IN ERROR READING DATA OR ECC
4824	020536	104035			ERROR	35		:ECC ERROR READING DATA
4825	020540	104036			ERROR	36		:CSI ERROR AFTER READING DATA OR ECC
4826	020542	104041			ERROR	41		:ECC REG INCORRECT AFTER ECC READ
4827	020544	104042			ERROR	42		:ERR IN ECC PAT CALC.
4828	020546	104043			ERROR	43		:ERR IN ECC POS COUNT
4829								
4830								
4831	020550	012701	000040		MOV	#8,*4,R1		:SET COUNT TO END OF SECTOR
4832	020554	012762	000440	000026	MOV	#DMD!MCLK,RKMR1(R2)		:RUN THE CLOCK
4833	020562	012762	000040	000026	MOV	#DMD,RKMR1(R2)		
4834	020570	005301			DEC	R1		:DEC THE COUNT
4835	020572	001370			BNE	6\$		:LOOP UNTIL ZERO
4836								
4837	020574	052737	100000	002352	BIS	#BIT15,BSEDES		:SET FOR BAD SECTOR ERROR
4838								
4839	020602	004437	025062		JSR	R4,DSECTR		:ISSUE SECTOR PULSE
4840								
4841	020606	004437	025504		JSR	R4,DRSYNC		:READ SYNC
4842	020612	104010			ERROR	10		:MR1 CONTENTS IN ERROR
4843	020614	104011			ERROR	11		:CSI CONTENTS IN ERROR
4844								
4845	020616	004437	026042		JSR	R4,DHDCMP		:DO HEADER COMPARE WITH BSE ERROR
4846	020622	104012			ERROR	12		:MR1 CONTENTS IN ERROR
4847	020624	104013			ERROR	13		:CSI IN ERROR AFTER SEARCH
4848	020626	000411			BR	1\$		:EXPECTED RETURN (NO DA INCREMENT)

F08

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 96
 DZR6EA.F11 28-SEP-76 15:31 T32 SILO UNLOAD BEFORE HEADER ERROR

SEQ 0096

```

4849
4850      :      IF THIS CODE IS EXECUTED, THE CONTROLLER DID NOT INHIBIT
4851      :      THE ADDRESS INCREMENT BECAUSE OF THE BSE ERROR.
4852
4853 020630 005037 002352      CLR      BSEDES      ;CLEAR BSE DESIRED SWITCH
4854 020634 013737 002274 002240  MOV      L.DCYL,E.DCYL ;SET EXPECTED CYL
4855 020642 013737 002266 002226  MOV      L.DA,E.DA   ;SET EXPECTED SEC/TRACK
4856 020650 104047      ERROR  47      ;UNEXPECTED ADDRESS INCREMENT
4857
4858 020652 016237 000014 002174 1$:  MOV      RKER(R2),T.ER ;GET ERROR REG
4859 020660 032737 000200 002174  BIT      #BSE,T.ER   ;TEST IF BSE SET
4860 020666 001005      BNE      2$      ;YES - SKIP
4861 020670 012737 000200 002234  MOV      #BSE,E.ER   ;SET EXPECTED ERROR REG
4862 020676 104051      ERROR  51      ;"BSE DID NOT SET"
4863 020700 000453      BR       TST33   ;GO TO NEXT TEST
4864
4865 020702 004437 034154      JSR      R4,GETREG ;GET 611 REGS
4866 020706 013737 002260 002220  MOV      L.CS1,E.CS1 ;SET EXPECTED CS1
4867 020714 023737 002220 002160  CMP      E.CS1,T.CS1 ;CHECK IF CS1 OK
4868 020722 001402      BEQ      3$      ;YES - SKIP
4869 020724 104056      ERROR  56      ;"CS1 ERROR AFTER WRITE CHECK"
4870 020726 003440      BR       TST33   ;GO TO NEXT TEST
4871
4872 020730 012701 010540      MOV      #278.*16.,R1 ;SET COUNT TO NEXT ECC FIELD
4873 020734 005037 002320      CLR      PR.BIT   ;CLEAR PRESENT AND PREVIOUS BITS
4874 020740 005037 002322      CLR      M1.BIT
4875 020744 004737 033414 4$:  JSR      PC,RDBIT  ;SIMULATE READ BIT
4876 020750 005301      DEC      R1          ;DEC COUNT
4877 020752 001374      BNE      4$      ;LOOP UNTIL ZERO
4878
4879 020754 004437 034154      JSR      R4,GETREG ;GET 611 REGS
4880 020760 042737 000001 002220  BIC      #G0,E.CS1 ;SET UP EXPECTED CS1
4881 020766 052737 100200 002220  BIS      #RDY!CERR,E.CS1
4882 020774 023737 002220 002160  CMP      E.CS1,T.CS1 ;CHECK IF CS1 OK
4883 021002 001402      BEQ      7$      ;YES - SKIP
4884 021004 104056      ERROR  56      ;"CS1 ERROR AFTER READ DATA"
4885 021006 000410      BR       TST33   ;GO TO NEXT TEST
4886
4887 021010 012737 051440 002224 7$:  MOV      #IBUFF+1000,E.BA ;SET EXPECTED BA
4888 021016 023737 002224 002164  CMP      E.BA,T.BA   ;TEST IF BA CORRECT
4889 021024 001401      BEQ      5$      ;YES - SKIP
4890 021026 104052      ERROR  52      ;BUS ADDRESS INCORRECT
4891
4892 021030      5$:
4893
4894      .SBTTL  **WRITE CHECK TESTS
4895
4896      ;*****
4897      ;TEST 33      WRITE CHECK OF 400 WORDS
4898      ;
4899      ;
4900      ;      CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
4901      ;      PUT CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A WRITE
4902      ;      CHECK OF 400 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
4903      ;      CYLINDER 0, HEAD 0, SECTOR 0.  CLOCK THROUGH SEEK,
4904      ;      AND DRIVE CLEAR MESSAGES.  SIMULATE A SECTOR PULSE,
4905      ;      A GOOD HEADER, THE 400 WORDS TO BE WRITE CHECKED,

```

G08

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 97
 DZR6EA.P11 28-SEP-76 15:31 T33 WRITE CHECK OF 400 WORDS

SEG 0097

```

4905      ;*      AND A GOOD 32 BIT ECC.  MAKE SURE NO ERRORS SET.
4906      ;*
4907      ;*****
4908      ST33: SCOPE
4909      MOV     #10, $TIMES      ;DO 10. ITERATIONS
4910      CLR     HIBITS          ;CLEAR HI ORDER BITS
4911      CLR     BSEDES          ;CLEAR BSE DESIRED
4912
4913      CLR     ECCSRC           ;CLEAR ECC SOURCE FLAG
4914      CLR     ECCSRC           ;CLEAR ECC SOURCE
4915      JSR     R4, LOADRK       ;LOAD "L" REGISTERS
4916      0                ;CYLINDER 0
4917      .BYTE    0              ;SECTOR 0
4918      .BYTE    0              ;TRACK 0
4919      Ibuff      050440        ;BUFFER ADDRESS Ibuff
4920      -400        177400      ;WORD COUNT -400
4921      WRTCHK              ;COMMAND WRTCHK
4922
4923      JSR     R4, CLRIBF       ;GO CLEAR INPUT BUFFER
4924
4925      JSR     R4, BLDDAT       ;GO BUILD DATA
4926      4                ;PATTERN 4
4927
4928      MOV     #400, R0         ;SET A COUNT FOR BUFFER SIZE
4929      MOV     #IBuff, R1       ;SET ADDRESS OF Ibuff
4930      MOV     #OBuf, R3       ;SET ADDRESS OF OBuf
4931      75$:    MOV     (R3)+, (R1)+ ;MOV PATTERN TO Ibuff
4932      DEC     R0              ;DEC COUNT
4933      BNE     75$             ;LOOP UNTIL PATTERN IS MOVED
4934
4935      110$:    MOV     #CLR, RKCS1(R2) ;CLEAR CONTROLLER
4936      JSR     R4, OPSTRT       ;START THE OPERATION
4937      JSR     R4, DISEEK       ;SIMULATE IMPLIED SEEK
4938      ERROR   4                ;CS1 MISCOMPARE
4939      ERROR   5                ;MR1
4940      ERROR   6                ;MR2
4941      ERROR   7                ;MR3
4942
4943      JSR     R4, DSECTR       ;SIMULATE SECTOR PULSE
4944
4945      JSR     R4, DRSYNC       ;SIMULATE HEADER PREAMBLE
4946      ERROR   10              ;MR1 CONTENTS IN ERROR
4947      ERROR   11              ;CS1 CONTENTS IN ERROR
4948
4949      JSR     R4, DHDCMP       ;SIMULATE HEADER SEARCH
4950      ERROR   12              ;MR1 CONTENTS IN ERROR
4951      ERROR   13              ;CS1 IN ERROR AFTER SEARCH
4952      ERROR   14              ;RKDCYL OR RKDA IN ERROR AFTER SEARCH
4953
4954      JSR     R4, DRGPSN       ;GO READ GAP AND SYNC
4955      ERROR   15              ;MR1 IN ERROR READING GAP
4956      ERROR   16              ;CS1 IN ERROR AFTER READING GAP
4957      ERROR   32              ;MR1 IN ERROR READING SYNC
4958      ERROR   33              ;CS1 IN ERROR AFTER READING SYNC
4959
4960      MOV     #0, ECPATX       ;LOAD EXPECTED PATTERN

```

H08

 RA611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZRA6-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 98
 DZRA6A.F11 28-SEP-76 15:31 T33 WRITE CHECK OF 400 WORDS

SEQ 0098

```

4961 021230 012737 004066 002350      MOV      #4066,ECPOSX      ;LOAD EXPECTED POSITION
4962 021236 004437 030774              JSR      R4,DWRCHK      ;GO SIMULATE WRITE CHECK
4963 021242 000400                      400              ;NUMBER OF WORDS TO SIMULATE
4964 021244 104053                      ERROR    53          ;MR1 IN ERROR ON WRT CK OR ECC READ
4965 021246 104054                      ERROR    54          ;ECC ERROR IN WRITE CHECK OP
4966 021250 104055                      ERROR    55          ;CS1 ERROR AFTER WRT CHK OR ECC READ
4967 021252 104041                      ERROR    41          ;ECC REG INCORRECT AFTER ECC READ
4968 021254 104042                      ERROR    42          ;ERR IN ECC PAT CALC.
4969 021256 104043                      ERROR    43          ;ERR IN ECC POS COUNT
4970
4971
4972 021260 012700 002710      MOV      #40.*37.,R0      ;SET COUNT TO EMPTY SILO
4973
4974 021264 012762 000440 000026 15:    MOV      #DMD!MCLK,RKMR1(R2) ;CLOCK CONTROLLER
4975 021272 012762 000040 000026      MOV      #DMD,RKMR1(R2)
4976 021300 005300                      DEC      R0              ;DEC COUNT
4977 021302 001370                      BNE     15              ;LOOP UNTIL 0
4978
4979 021304 004437 034154      JSR      R4,GETREG      ;GET 611 REGS
4980 021310 013737 002260 002220      MOV      L.CS1,E.CS1    ;GET EXPECTED CS1
4981 021316 042737 000001 002220      BIC      #GO,E.CS1      ;CLEAR GO BIT
4982 021324 052737 000200 002220      BIS      #RDY,E.CS1     ;SET READY BIT
4983 021332 023737 002160 002220      CMP      T.CS1,E.CS1    ;CHECK IF CS1 CORRECT
4984 021340 001401                      BEQ     25              ;YES - SKIP
4985 021342 104057                      ERROR    57              ;"CS1 IN ERROR AFTER WRITE CHECK"
4986 021344
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000
5001
5002
5003 021344 000004      ;*****
5004 021346 012737 000001 001234 15:  *TEST 34      WRITE CHECK WITH DCK
5005
5006
5007
5008
5009
5010
5011
5012
5013
5014
5015
5016

```

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
 PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
 CHECK OF 400 WORDS TO AN RK06 IN 26 SECTOR FORMAT,
 CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
 AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE,
 A GOOD HEADER, THE 400 WORDS TO BE WRITE CHECKED,
 AND A 32 BIT ECC INDICATING A 1 BIT BURST ERROR.
 VERIFY THAT THE POSITION REGISTER INCREMENTS PROPERLY
 AND THAT THE FINAL PATTERN AND POSITION ARE CORRECT.
 CHECK THAT DATA CHECK AND CONTROLLER ERROR ARE SET.

```

5003 021344 000004      ;*****
5004 021346 012737 000001 001234 15:  *TEST 34: SCOPE
5005
5006 021354 005037 002352      MOV      #1,$TIMES      ;;DO 1 ITERATION
5007 021360 005037 002354      CLR      BSEDES        ;CLEAR BAD SECTOR DESIRED
5008 021364 004437 034026      CLR      HIBITS        ;CLEAR HI ORDER BITS
5009 021370 000000                      JSR      R4,LOADRK      ;LOAD "L" REGISTERS
5010 021372 000                      0              ;CYLINDER 0
5011 021373 000                      .BYTE    0              ;SECTOR 0
5012 021374 050440                      .BYTE    0              ;TRACK 0
5013 021376 177400                      Ibuff     ;BUFFER ADDRESS Ibuff
5014 021400 000031                      -400      ;WORD COUNT -400
5015
5016 021402 004437 033566      WRTCHK      ;COMMAND WRTCHK
5017
5018
5019
5020
5021
5022
5023
5024
5025
5026
5027
5028
5029
5030
5031
5032
5033
5034
5035
5036
5037
5038
5039
5040
5041
5042
5043
5044
5045
5046
5047
5048
5049
5050
5051
5052
5053
5054
5055
5056
5057
5058
5059
5060
5061
5062
5063
5064
5065
5066
5067
5068
5069
5070
5071
5072
5073
5074
5075
5076
5077
5078
5079
5080
5081
5082
5083
5084
5085
5086
5087
5088
5089
5090
5091
5092
5093
5094
5095
5096
5097
5098
5099
5100
5101
5102
5103
5104
5105
5106
5107
5108
5109
5110
5111
5112
5113
5114
5115
5116
5117
5118
5119
5120
5121
5122
5123
5124
5125
5126
5127
5128
5129
5130
5131
5132
5133
5134
5135
5136
5137
5138
5139
5140
5141
5142
5143
5144
5145
5146
5147
5148
5149
5150
5151
5152
5153
5154
5155
5156
5157
5158
5159
5160
5161
5162
5163
5164
5165
5166
5167
5168
5169
5170
5171
5172
5173
5174
5175
5176
5177
5178
5179
5180
5181
5182
5183
5184
5185
5186
5187
5188
5189
5190
5191
5192
5193
5194
5195
5196
5197
5198
5199
5200
5201
5202
5203
5204
5205
5206
5207
5208
5209
5210
5211
5212
5213
5214
5215
5216
5217
5218
5219
5220
5221
5222
5223
5224
5225
5226
5227
5228
5229
5230
5231
5232
5233
5234
5235
5236
5237
5238
5239
5240
5241
5242
5243
5244
5245
5246
5247
5248
5249
5250
5251
5252
5253
5254
5255
5256
5257
5258
5259
5260
5261
5262
5263
5264
5265
5266
5267
5268
5269
5270
5271
5272
5273
5274
5275
5276
5277
5278
5279
5280
5281
5282
5283
5284
5285
5286
5287
5288
5289
5290
5291
5292
5293
5294
5295
5296
5297
5298
5299
5300
5301
5302
5303
5304
5305
5306
5307
5308
5309
5310
5311
5312
5313
5314
5315
5316
5317
5318
5319
5320
5321
5322
5323
5324
5325
5326
5327
5328
5329
5330
5331
5332
5333
5334
5335
5336
5337
5338
5339
5340
5341
5342
5343
5344
5345
5346
5347
5348
5349
5350
5351
5352
5353
5354
5355
5356
5357
5358
5359
5360
5361
5362
5363
5364
5365
5366
5367
5368
5369
5370
5371
5372
5373
5374
5375
5376
5377
5378
5379
5380
5381
5382
5383
5384
5385
5386
5387
5388
5389
5390
5391
5392
5393
5394
5395
5396
5397
5398
5399
5400
5401
5402
5403
5404
5405
5406
5407
5408
5409
5410
5411
5412
5413
5414
5415
5416
5417
5418
5419
5420
5421
5422
5423
5424
5425
5426
5427
5428
5429
5430
5431
5432
5433
5434
5435
5436
5437
5438
5439
5440
5441
5442
5443
5444
5445
5446
5447
5448
5449
5450
5451
5452
5453
5454
5455
5456
5457
5458
5459
5460
5461
5462
5463
5464
5465
5466
5467
5468
5469
5470
5471
5472
5473
5474
5475
5476
5477
5478
5479
5480
5481
5482
5483
5484
5485
5486
5487
5488
5489
5490
5491
5492
5493
5494
5495
5496
5497
5498
5499
5500
5501
5502
5503
5504
5505
5506
5507
5508
5509
5510
5511
5512
5513
5514
5515
5516
5517
5518
5519
5520
5521
5522
5523
5524
5525
5526
5527
5528
5529
5530
5531
5532
5533
5534
5535
5536
5537
5538
5539
5540
5541
5542
5543
5544
5545
5546
5547
5548
5549
5550
5551
5552
5553
5554
5555
5556
5557
5558
5559
5560
5561
5562
5563
5564
5565
5566
5567
5568
5569
5570
5571
5572
5573
5574
5575
5576
5577
5578
5579
5580
5581
5582
5583
5584
5585
5586
5587
5588
5589
5590
5591
5592
5593
5594
5595
5596
5597
5598
5599
5600
5601
5602
5603
5604
5605
5606
5607
5608
5609
5610
5611
5612
5613
5614
5615
5616
5617
5618
5619
5620
5621
5622
5623
5624
5625
5626
5627
5628
5629
5630
5631
5632
5633
5634
5635
5636
5637
5638
5639
5640
5641
5642
5643
5644
5645
5646
5647
5648
5649
5650
5651
5652
5653
5654
5655
5656
5657
5658
5659
5660
5661
5662
5663
5664
5665
5666
5667
5668
5669
5670
5671
5672
5673
5674
5675
5676
5677
5678
5679
5680
5681
5682
5683
5684
5685
5686
5687
5688
5689
5690
5691
5692
5693
5694
5695
5696
5697
5698
5699
5700
5701
5702
5703
5704
5705
5706
5707
5708
5709
5710
5711
5712
5713
5714
5715
5716
5717
5718
5719
5720
5721
5722
5723
5724
5725
5726
5727
5728
5729
5730
5731
5732
5733
5734
5735
5736
5737
5738
5739
5740
5741
5742
5743
5744
5745
5746
5747
5748
5749
5750
5751
5752
5753
5754
5755
5756
5757
5758
5759
5760
5761
5762
5763
5764
5765
5766
5767
5768
5769
5770
5771
5772
5773
5774
5775
5776
5777
5778
5779
5780
5781
5782
5783
5784
5785
5786
5787
5788
5789
5790
5791
5792
5793
5794
5795
5796
5797
5798
5799
5800
5801
5802
5803
5804
5805
5806
5807
5808
5809
5810
5811
5812
5813
5814
5815
5816
5817
5818
5819
5820
5821
5822
5823
5824
5825
5826
5827
5828
5829
5830
5831
5832
5833
5834
5835
5836
5837
5838
5839
5840
5841
5842
5843
5844
5845
5846
5847
5848
5849
5850
5851
5852
5853
5854
5855
5856
5857
5858
5859
5860
5861
5862
5863
5864
5865
5866
5867
5868
5869
5870
5871
5872
5873
5874
5875
5876
5877
5878
5879
5880
5881
5882
5883
5884
5885
5886
5887
5888
5889
5890
5891
5892
5893
5894
5895
5896
5897
5898
5899
5900
5901
5902
5903
5904
5905
5906
5907
5908
5909
5910
5911
5912
5913
5914
5915
5916
5917
5918
5919
5920
5921
5922
5923
5924
5925
5926
5927
5928
5929
5930
5931
5932
5933
5934
5935
5936
5937
5938
5939
5940
5941
5942
5943
5944
5945
5946
5947
5948
5949
5950
5951
5952
5953
5954
5955
5956
5957
5958
5959
5960
5961
5962
5963
5964
5965
5966
5967
5968
5969
5970
5971
5972
5973
5974
5975
5976
5977
5978
5979
5980
5981
5982
5983
5984
5985
5986
5987
5988
5989
5990
5991
5992
5993
5994
5995
5996
5997
5998
5999
6000
6001
6002
6003
6004
6005
6006
6007
6008
6009
6010
6011
6012
6013
6014
6015
6016
6017
6018
6019
6020
6021
6022
6023
6024
6025
6026
6027
6028
6029
6030
6031
6032
6033
6034
6035
6036
6037
6038
6039
6040
6041
6042
6043
6044
6045
6046
6047
6048
6049
6050
6051
6052
6053
6054
6055
6056
6057
6058
6059
6060
6061
6062
6063
6064
6065
6066
6067
6068
6069
6070
6071
6072
6073
6074
6075
6076
6077
6078
6079
6080
6081
6082
6083
6084
6085
6086
6087
6088
6089
6090
6091
6092
6093
6094
6095
6096
6097
6098
6099
6100
6101
6102
6103
6104
6105
6106
6107
6108
6109
6110
6111
6112
6113
6114
6115
6116
6117
6118
6119
6120
6121
6122
6123
6124
6125
6126
6127
6128
6129
6130
6131
6132
6133
6134
6135
6136
6137
6138
6139
6140
6141
6142
6143
6144
6145
6146
6147
6148
6149
6150
6151
6152
6153
6154
6155
6156
6157
6158
6159
6160
6161
6162
6163
6164
6165
6166
6167
6168
6169
6170
6171
6172
6173
6174
6175
6176
6177
6178
6179
6180
6181
6182
6183
6184
6185
6186
6187
6188
6189
6190
6191
6192
6193
6194
6195
6196
6197
6198
6199
6200
6201
6202
6203
6204
6205
6206
6207
6208
6209
6210
6211
6212
6213
6214
6215
6216
6217
6218
6219
6220
6221
6222
6223
6224
6225
6226
6227
6228
6229
6230
6231
6232
6233
6234
6235
6236
6237
6238
6239
6240
6241
6242
6243
6244
6245
6246
6247
6248
6249
6250
6251
6252
6253
6254
6255
6256
6257
6258
6259
6260
6261
6262
6263
6264
6265
6266
6267
6268
6269
6270
6271
6272
6273
6274
6275
6276
6277
6278
6279
6280
6281
6282
6283
6284
6285
6286
6287
6288
6289
6290
6291
6292
6293
6294
6295
6296
6297
6298
6299
6300
6301
6302
6303
6304
6305
6306
6307
6308
6309
6310
6311
6312
6313
6314
6315
6316
6317
6318
6319
6320
6321
6322
6323
6324
6325
6326
6327
6328
6329
6330
6331
6332
6333
6334
6335
6336
6337
6338
6339
6340
6341
6342
6343
6344
6345
6346
6347
6348
6349
6350
6351
6352
6353
6354
6355
6356
6357
6358
6359
6360
6361
6362
6363
6364
6365
6366
6367
6368
6369
6370
6371
6372
6373
6374
6375
6376
6377
6378
6379
6380
6381
6382
6383
6384
6385
6386
6387
6388
6389
6390
6391
6392
6393
6394
6395
6396
6397
6398
6399
6400
6401
6402
6403
6404
6405
6406
6407
6408
6409
6410
6411
6412
6413
6414
6415
6416
6417
6418
6419
6420
6421
6422
6423
6424
6425
6426
6427
6428
6429
6430
6431
6432
6433
6434
6435
6436
6437
6438
6439
6440
6441
6442
6443
6444
6445
6446
6447
6448
6449
6450
6451
6452
6453
6454
6455
6456
6457
6458
6459
6460
6461
6462
6463
6464
6465
6466
6467
6468
6469
6470
6471
6472
6473
6474
6475
6476
6477
6478
6479
6480
6481
6482
6483
6484
6485
6486
6487
6488
6489
6490
6491
6492
6493
6494
6495
6496
6497
6498
6499
6500
6501
6502
6503
6504
6505
6506
6507
6508
6509
6510
6511
6512
6513
6514
6515
6516
6517
6518
6519
6520
6521
6522
6523
6524
6525
6526
6527
6528
6529
6530
6531
6532
6533
6534
6535
6536
6537
6538
6539
6540
6541
6542
6543
6544
6545
6546
6547
6548
6549
6550
6551
6552
6553
6554
6555
6556
6557
6558
6559
6560
6561
6562
6563
6564
6565
6566
6567
6568
6569
6570
6571
6572
6573
6574
6575
6576
6577
6578
6579
6580
6581
6582
6583
6584
6585
6586
6587
6588
6589
6590
6591
6592
6593
6594
6595
6596
6597
6598
6599
6600
6601
6602
6603
6604
6605
6606
6607
6608
6609
6610
6611
6612
6613
6614
6615
6616
6617
6618
6619
6620
6621
6622
6623
6624
6625
6626
6627
6628
6629
6630
6631
6632
6633
6634
6635
6636
6637
6638
6639
6640
6641
6642
6643
6644
6645
6646
6647
6648
6649
6650
6651
6652
6653
6654
6655
6656
6657
6658
6659
6660
6661
6662
6663
6664
6665
6666
6667
6668
6669
6670
6671
6672
6673
6674
6675
6676
6677
6678
6679
6680
6681
6682
6683
6684
6685
6686
6687
6688
6689
6690
6691
6692
6693
6694
6695
6696
6697
6698
6699
6700
6701
6702
6703
6704
6705
6706
6707
6708
6709
6710
6711
6712
6713
6714
6715
6716
6717
6718
6719
6720
6721
6722
6723
6724
6725
6726
6727
6728
6729
6730
6731
6732
6733
6734
6735
6736
6737
6738
6739
6740
6741
6742
6743
6744
6745
6746
6747
6748
6749
6750
6751
6752
6753
6754
6755
6756
6757
6758
6759
6760
6761
6762
6763
6764
6765
6766
6767
6768
6769
6770
6771
6772
6773
6774
6775
6776
6777
6778
6779
6780
6781
6782
6783
6784
6785
6786
6787
6788
6789
6790
6791
6792
6793
6794
6795
6796
6797
6798
6799
6800
6801
6802
6803
6804
6805
6806
6807
6808
6809
6810
6811
6812
6813
6814
6815
6816
6817
6818
6819
6820
6821
6822
6823
6824
6825
6826
6827
6828
6829
6830
6831
6832
6833
6834
6835
6836
6837
6838
6839
6840
6841
6842
6843
6844
6845
6846
6847
6848
6849
6850
6851
6852
6853
6854
6855
6856
6857
6858
6859
6860
6861
6862
6863
6864
6865
6866
6867
6868
6869
6870
6871
6872
6873
6874
6875
6876
6877
6878
6879
6880
6881
6882
6883
6884
6885
6886
6887
6888
6889
6890
6891
6892
6893
6894
6895
6896
6897
6898
6899
6900
6901
6902
6903
6904
6905
6906
6907
6908
6909
6910
6911
6912
6913
6914
6915
6916
6917
6918
6919
6920
6921
6922
6923
6924
6925
6926
6927
6928
6929
6930
6931
6932
6933
6934
6935
6936
6937
6938
6939
6940
6941
6942
6943
6944
6945
6946
6947
6948
6949
6950
6951
6952
6953
6954
6955
6956
6957
6958
6959
6960
6961
6962
6963
6964
6965
6966
6967
6968
6969
6970
6971
6972
6973
6974
6975
6976
6977
6978
6979
6980
6981
6982
6983
6984
6985
6986
6987
6988
6989
6990
6991
6992
6993
6994
6995
6996
6997
6998
6999
7000
7001
7002
7003
7004
7005
7006
7007
7008
7009
7010
7011
7012
7013
7014
7015
7016
7017
7018
7019
7020
7021
7022
7023
7024
7025
7026
7027
7028
7029
7030
7031
7032
7033
7034
7035
7036
7037
7038
7039
7040
7041
7042
7043
7044
7045
7046
7047
7048
7049
7050
7051
7052
7053
7054
7055
7056
7057
7058
7059
7060
7061
7062
7063
7064
7065
7066
7067
7068
7069
7070
7071
7072
7073
7074
7075
7076
7077
7078
7079
7080
7081
7082
7083
7084
7085
7086
7087
7088
7089
7090
7091
7092
7093
7094
7095
7096
7097
7098
7099
7100
7101
7102
7103
7104
7105
710
```

```

5017
5018 021406 004437 033612      JSR      R4,BLDDAT      ;GO BUILD DATA
5019 021412 000005              S              ;PATTERN 5
5020
5021 021414 012700 000400      MOV      #400,R0      ;SET A COUNT FOR BUFFER SIZE
5022 021420 012701 050440      MOV      #IBUFF,R1     ;SET ADDRESS OF IBUFF
5023 021424 012703 051444      MOV      #OBUFF,R3     ;SET ADDRESS OF OBUFF
5024 021430 012321              75$: MOV      (R3)+,(R1)+    ;MOV PATTERN TO IBUFF
5025 021432 005300              DEC      R0              ;DEC COUNT
5026 021434 001375              BNE      75$            ;LOOP UNTIL PATTERN IS MOVED
5027
5028 021436 012700 001002      MOV      #1002,R0     ;SET INDEX INTO BUFFER
5029 021442 012737 126073 052444 MOV      #126073,OBUFF+1000 ;STORE ECC WORDS
5030 021450 012737 151052 052446 MOV      #151052,OBUFF+1002
5031 021456 012737 000001 002344 MOV      #1,ECCSRC     ;SET SOURCE TO INDICATE ECC IN BUFFER
5032 021464 032760 000100 051444 BIT      #BIT6,OBUFF(R0) ;TEST IF BIT SET
5033 021472 001004              BNE      100$          ;YES - SKIP
5034 021474 052760 000100 051444 BIS      #BIT6,OBUFF(R0) ;ELSE SET BIT
5035 021502 000403              BR       101$
5036 021504 042760 000100 051444 100$: BIC      #BIT6,OBUFF(R0) ;CLEAR BIT
5037 021512              101$:
5038
5039 021512 012762 100000 000000 110$: MOV      #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
5040 021520 004437 034072      JSR      R4,OPSTRAT    ;START THE OPERATION
5041 021524 004437 025100      JSR      R4,DISEEK     ;SIMULATE IMPLIED SEEK
5042 021530 104004              ERROR     4              ;CS1 MISCOMPARE
5043 021532 104005              ERROR     5              ;MR1
5044 021534 104006              ERROR     6              ;MR2
5045 021536 104007              ERROR     7              ;MR3
5046
5047 021540 004437 025062      JSR      R4,DSECTR      ;SIMULATE SECTOR PULSE
5048
5049 021544 004437 025504      JSR      R4,DRSYNC     ;SIMULATE HEADER PREAMBLE
5050 021550 104010              ERROR     10             ;MR1 CONTENTS IN ERROR
5051 021552 104011              ERROR     11             ;CS1 CONTENTS IN ERROR
5052
5053 021554 004437 026042      JSR      R4,DHDCMP     ;SIMULATE HEADER SEARCH
5054 021560 104012              ERROR     12             ;MR1 CONTENTS IN ERROR
5055 021562 104013              ERROR     13             ;CS1 IN ERROR AFTER SEARCH
5056 021564 104014              ERROR     14             ;RKDCYL OR RKDA IN ERROR AFTER SEARCH
5057
5058 021566 004437 030276      JSR      R4,DRGPSN     ;GO READ GAP AND SYNC
5059 021572 104015              ERROR     15             ;MR1 IN ERROR READING GAP
5060 021574 104016              ERROR     16             ;CS1 IN ERROR AFTER READING GAP
5061 021576 104032              ERROR     32             ;MR1 IN ERROR READING SYNC
5062 021600 104033              ERROR     33             ;CS1 IN ERROR AFTER READING SYNC
5063
5064 021602 012737 000001 002346 MOV      #1,ECPATX     ;LOAD EXPECTED PATTERN
5065 021610 012737 004066 002350 MOV      #4066,ECPOSX   ;LOAD EXPECTED POSITION
5066 021616 004437 030774      JSR      R4,DWRTCK     ;GO SIMULATE WRITE CHECK
5067 021622 000400              400              ;NUMBER OF WORDS TO SIMULATE
5068 021624 104053              ERROR     53             ;MR1 IN ERROR ON WRT CK OR ECC READ
5069 021626 104054              ERROR     54             ;ECC ERROR IN WRITE CHECK OP
5070 021630 104055              ERROR     55             ;CS1 ERROR AFTER WRT CHK OR ECC READ
5071 021632 104041              ERROR     41             ;ECC REG INCORRECT AFTER ECC READ
5072 021634 104042              ERROR     42             ;ERR IN ECC PAT CALC.

```

J08

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 100
 DZR6EA.P11 28-SEP-76 15:31 T34 WRITE CHECK WITH DCK

SEQ 0100

```

5073 021636 104043          ERROR 43          ;ERR IN ECC POS COUNT
5074
5075
5076 021640 012737 000020 002326      MOV    #20,BITCNT      ;SET BIT COUNT FOR CORRECTION COUNT
5077 021646 012700 000005              MOV    #5,R0          ;SET COUNT FOR ECCPOS PASS THRU 0
5078 021652 012701 013672              MOV    #13672,R1       ;SET COUNT FOR ECCPOS TO 0 FIRST PASS
5079 021656 005037 001214              CLR     $TMP5         ;CLEAR SWITCH
5080
5081 021662 004737 033414          3$:    JSR    PC,RDBIT      ;GO READ BIT
5082 021666 004737 032260              JSR    PC,ECCGEN      ;GENERATE ECC
5083 021672 016237 000032 002214      MOV    RKECPT(R2),T.ECPT ;GET PATTERN
5084 021700 016237 000030 002212      MOV    RKECPS(R2),T.ECPS ;GET POSITION
5085 021706 005237 002350              INC     ECPOSX        ;BUMP POSITION EXPECTED
5086 021712 013737 002350 002252      MOV    ECPOSX,E.ECPS   ;SET FOR COMPARE AND REPORT
5087 021720 023737 002254 002214      CMP     E.ECPT,T.ECPT  ;CHECK IF PATTERN CORRECT
5088 021726 001401                  BEQ     1$            ;YES - SKIP
5089 021730 104042                  ERROR 42
5090 021732 023737 002252 002212 1$:    CMP     E.ECPS,T.ECPS  ;CHECK IF POSITION CORRECT
5091 021740 001401                  BEQ     2$            ;YES - SKIP
5092 021742 104043                  ERROR 43
5093 021744 005237 002326          2$:    INC     BITCNT        ;BUMP BIT COUNT
5094 021750 005301                  DEC     R1            ;DEC COUNT TO ZERO IN ECCPOS
5095 021752 001343                  BNE     3$            ;NOT YET ZERO - LOOP
5096 021754 005300                  DEC     R0            ;DEC PASS THRU 0 COUNT
5097 021756 001406                  BEQ     4$            ;IF 0 - SKIP
5098 021750 012737 177777 002350      MOV    #-1,ECPOSX      ;PRESET EXPECTED POS TO GO TO ZERO
5099 021766 012701 020000              MOV    #20000,R1       ;SET COUNT TO NEXT 0 IN ECC POS
5100 021772 000733                  BR      3$            ;GO DO LOOP AGAIN
5101
5102 021774 005737 001214          4$:    TST     $TMP5         ;TEST SWITCH
5103 022000 001012                  BNE     5$            ;ALREADY SET - SKIP
5104 022002 012701 010016              MOV    #10016,R1      ;SET COUNT TO FINAL POSITION
5105 022006 012700 000001              MOV    #1,R0          ;SET R0 TO RETURN AFTER ONE PASS
5106 022012 012737 177777 002350      MOV    #-1,ECPOSX      ;SET EXP POS TO GO TO 0 ON NEXT BIT
5107 022020 005237 001214              INC     $TMP5         ;BUMP SWITCH
5108 022024 000716                  BR      3$            ;GO DO LOOP AGAIN
5109
5110 022026 004737 033414          5$:    JSR    PC,RDBIT      ;ONE MORE BIT TO SET READY
5111 022032 004437 034154              JSR    R4,GETREG      ;GET '611 REGISTERS
5112 022036 013737 002260 002220      MOV    L.CS1,E.CS1    ;GET EXPECTED CS1
5113 022044 052737 100200 002220      BIS    #RDY!CERR,E.CS1 ;SET READY AND ERROR
5114 022052 042737 000001 002220      BIC    #GO,E.CS1      ;CLEAR GO
5115 022060 023737 002160 002220      CMP     T.CS1,E.CS1   ;CHECK IF CS1 CORRECT
5116 022066 001402                  BEQ     6$            ;YES - SKIP
5117 022070 104044                  ERROR 44
5118 022072 000410                  BR      7$
5119 022074 032737 000100 002174 6$:    BIT     #ECH,T.ER      ;TEST IF ECC HARD ERROR SET
5120 022102 001404                  BEQ     7$            ;YES - SKIP
5121 022104 012737 100000 002234      MOV    #DCK,E.ER      ;SET EXPECTED ERROR REG
5122 022112 104045                  ERROR 45
5123
5124 022114          7$:
5125
5126 ;*****
5127 ;*TEST 35      WRITE CHECK OF 18 BIT WORDS
5128 ;*
```

K08

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 101
 DZR6EA.P11 28-SEP-76 15:31 T35 WRITE CHECK OF 18 BIT WORDS

SEQ 0101

```

5129      ;*      CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
5130      ;*      PUT CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A WRITE
5131      ;*      CHECK OF 400 WORDS TO AN RK06 IN 24 SECTOR FORMAT,
5132      ;*      CYLINDER 0, HEAD 0, SECTOR 0.  CLOCK THROUGH SEEK,
5133      ;*      AND DRIVE CLEAR MESSAGES.  SIMULATE A SECTOR PULSE,
5134      ;*      A GOOD HEADER, THE 400 WORDS TO BE WRITE CHECKED,
5135      ;*      AND A GOOD 32 BIT ECC.  MAKE SURE NO ERRORS SET.
5136      ;*
5137      ;*****
5138 022114 000004      T35: SCOPE
5139 022116 012737 000001 001234      MOV      #1, $TIMES      ;;DO 1 ITERATION
5140
5141 022124 005037 002352      CLR      BSEDES      ;CLEAR BAD SECTOR DESIRED
5142 022130 005037 002354      CLR      HIBITS      ;CLEAR HI ORDER BITS
5143 022134 005037 002344      CLR      ECCSRC      ;CLEAR ECC SOURCE
5144 022140 004437 034026      JSR      R4, LOADRK      ;LOAD "L" REGISTERS
5145 022144 000000      0      ;CYLINDER 0
5146 022146      000      .BYTE 0      ;SECTOR 0
5147 022147      000      .BYTE 0      ;TRACK 0
5148 022150 050440      Ibuff      ;BUFFER ADDRESS Ibuff
5149 022152 177400      -400      ;WORD COUNT -400
5150 022154 013031      WRTCHK!CFMT      ;COMMAND WRTCHK!CFMT
5151
5152 022156 004437 033566      JSR      R4, CLRIBF      ;GO CLEAR INPUT BUFFER
5153
5154 022162 004437 033612      JSR      R4, BLDDAT      ;GO BUILD DATA
5155 022166 000004      4      ;PATTERN 4
5156
5157 022170 012700 000400      MOV      #400, R0      ;SET A COUNT FOR BUFFER SIZE
5158 022174 012701 050440      MOV      #IBUFF, R1      ;SET ADDRESS OF Ibuff
5159 022200 012703 051444      MOV      #OBUFF, R3      ;SET ADDRESS OF OBUFF
5160 022204 012321      75$: MOV      (R3)+, (R1)+      ;MOV PATTERN TO Ibuff
5161 022206 005300      DEC      R0      ;DEC COUNT
5162 022210 001375      BNE      75$      ;LOOP UNTIL PATTERN IS MOVED
5163
5164 022212 012762 100000 000000 110$: MOV      #CCLR, RKCS1(R2) ;CLEAR CONTROLLER
5165 022220 004437 034072      JSR      R4, OPSTRT      ;START THE OPERATION
5166 022224 004437 025100      JSR      R4, DISEEK      ;SIMULATE IMPLIED SEEK
5167 022230 104004      ERROR 4      ;CS1 MISCOMPARE
5168 022232 104005      ERROR 5      ;MR1
5169 022234 104006      ERROR 6      ;MR2
5170 022236 104007      ERROR 7      ;MR3
5171
5172 022240 004437 025062      JSR      R4, DSECTR      ;SIMULATE SECTOR PULSE
5173
5174 022244 004437 025504      JSR      R4, DRSYNC      ;SIMULATE HEADER PREAMBLE
5175 022250 104010      ERROR 10      ;MR1 CONTENTS IN ERROR
5176 022252 104011      ERROR 11      ;CS1 CONTENTS IN ERROR
5177
5178 022254 004437 026042      JSR      R4, DHDCMP      ;SIMULATE HEADER SEARCH
5179 022260 104012      ERROR 12      ;MR1 CONTENTS IN ERROR
5180 022262 104013      ERROR 13      ;CS1 IN ERROR AFTER SEARCH
5181 022264 104014      ERROR 14      ;RKDCYL OR RKDA IN ERROR AFTER SEARCH
5182
5183 022266 004437 030276      JSR      R4, DRGPSN      ;GO READ GAP AND SYNC
5184 022272 104015      ERROR 15      ;MR1 IN ERROR READING GAP

```

L08

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 102
DZR6EA.F11 28-SEP-76 15:31 T35 WRITE CHECK OF 18 BIT WORDS

SEQ 0102

5185	022274	104016			ERROR	16		;CS1 IN ERROR AFTER READING GAP
5186	022276	104032			ERROR	32		;MR1 IN ERROR READING SYNC
5187	022300	104033			ERROR	33		;CS1 IN ERROR AFTER READING SYNC
5188								
5189	022302	012737	000000	002346	MOV	#0,ECPTX		;LOAD EXPECTED PATTERN
5190	022310	012737	005066	002350	MOV	#5066,ECPOSX		;LOAD EXPECTED POSITION
5191	022316	004437	030774		JSR	R4,DWRTCK		;GO SIMULATE WRITE CHECK
5192	022322	000400			400			;NUMBER OF WORDS TO SIMULATE
5193	022324	104053			ERROR	53		;MR1 IN ERROR ON WRT CK OR ECC READ
5194	022326	104054			ERROR	54		;ECC ERROR IN WRITE CHECK OP
5195	022330	104055			ERROR	55		;CS1 ERROR AFTER WRT CHK OR ECC READ
5196	022332	104041			ERROR	41		;ECC REG INCORRECT AFTER ECC READ
5197	022334	104042			ERROR	42		;ERR IN ECC PAT CALC.
5198	022336	104043			ERROR	43		;ERR IN ECC POS COUNT
5199								
5200								
5201	022340	012700	003100		MOV	#40.*40.,R0		;SET COUNT TO EMPTY SILO
5202								
5203	022344	012762	000440	000026	1\$: MOV	#DMD!MCLK,RKMR1(R2)		;CLOCK CONTROLLER
5204	022352	012762	000040	000026	MOV	#DMD,RKMR1(R2)		
5205	022360	005300			DEC	R0		;DEC COUNT
5206	022362	001370			BNE	1\$		;LOOP UNTIL 0
5207								
5208	022364	004437	034154		JSR	R4,GETREG		;GET 611 REGS
5209	022370	013737	002260	002220	MOV	L.CS1,E.CS1		;GET EXPECTED CS1
5210	022376	042737	000001	002220	BIC	#GO,E.CS1		;CLEAR GO BIT
5211	022404	052737	000200	002220	BIS	#RDY,E.CS1		;SET READY BIT
5212	022412	023737	002160	002220	CMP	T.CS1,E.CS1		;CHECK IF CS1 CORRECT
5213	022420	001401			BEQ	2\$		;YES - SKIP
5214	022422	104057			ERROR	57		; "CS1 IN ERROR AFTER WRITE CHECK"
5215	022424							
5216								
5217								
5218								
5219								
5220								
5221								
5222								
5223								
5224								
5225								
5226								
5227								
5228								
5229	022424	000004			1\$T36: SCOPE			
5230	022426	012737	000001	001234	MOV	#1,\$TIMES		;DO 1 ITERATION
5231								
5232	022434	005037	002352		CLR	BSEDES		;CLEAR BAD SECTOR DESIRED
5233	022440	005037	002354		CLR	HIBITS		;CLEAR HI ORDER BITS
5234	022444	005037	002344		CLR	ECCSRC		;CLEAR ECC SOURCE
5235	022450	004437	034026		JSR	R4,LOADRK		;LOAD "L" REGISTERS
5236	022454	000000			0			;CYLINDER 0
5237	022456	000			.BYTE	0		;SECTOR 0
5238	022457	000			.BYTE	0		;TRACK 0
5239	022460	050440			IBUFF			;BUFFER ADDRESS IBUFF
5240	022462	177777			-1			;WORD COUNT -1

M08

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 103
DZR6EA.P11 28-SEP-76 15:31 T36 WRITE CHECK OF A PARTIAL SECTOR

SEQ 0103

5241	022464	000031			WRTCHK	;COMMAND WRTCHK
5242						
5243	022466	004437	033566		JSR R4,CLRIBF	;GO CLEAR INPUT BUFFER
5244						
5245	022472	004437	033612		JSR R4,BLDDAT	;GO BUILD DATA
5246	022476	000006			6	;PATTERN 6
5247						
5248	022500	012700	000400		MOV #400,R0	;SET A COUNT FOR BUFFER SIZE
5249	022504	012701	050440		MOV #IBUFF,R1	;SET ADDRESS OF IBUFF
5250	022510	012703	051444		MOV #OBUFF,R3	;SET ADDRESS OF OBUFF
5251	022514	012321		75\$:	MOV (R3)+,(R1)+	;MOV PATTERN TO IBUFF
5252	022516	005300			DEC R0	;DEC COUNT
5253	022520	001375			BNE 75\$	;LOOP UNTIL PATTERN IS MOVED
5254						
5255	022522	012762	100000	000000	110\$:	MOV #CCLR,RKCS1(R2)
5256	022530	004437	034072		JSR R4,OPSTRT	;CLEAR CONTROLLER
5257	022534	004437	025100		JSR R4,DISEEK	;START THE OPERATION
5258	022540	104004			ERROR 4	;SIMULATE IMPLIED SEEK
5259	022542	104005			ERROR 5	;CS1 MISCOMPARE
5260	022544	104006			ERROR 6	;MR1 "
5261	022546	104007			ERROR 7	;MR2 "
5262						;MR3 "
5263	022550	004437	025062		JSR R4,DSECTR	;SIMULATE SECTOR PULSE
5264						
5265	022554	004437	025504		JSR R4,DRSYNC	;SIMULATE HEADER PREAMBLE
5266	022560	104010			ERROR 10	;MR1 CONTENTS IN ERROR
5267	022562	104011			ERROR 11	;CS1 CONTENTS IN ERROR
5268						
5269	022564	004437	026042		JSR R4,DHDCMP	;SIMULATE HEADER SEARCH
5270	022570	104012			ERROR 12	;MR1 CONTENTS IN ERROR
5271	022572	104013			ERROR 13	;CS1 IN ERROR AFTER SEARCH
5272	022574	104014			ERROR 14	;RKDCYL OR RKDA IN ERROR AFTER SEARCH
5273						
5274	022576	004437	030276		JSR R4,DRGPSN	;GO READ GAP AND SYNC
5275	022602	104015			ERROR 15	;MR1 IN ERROR READING GAP
5276	022604	104016			ERROR 16	;CS1 IN ERROR AFTER READING GAP
5277	022606	104032			ERROR 32	;MR1 IN ERROR READING SYNC
5278	022610	104033			ERROR 33	;CS1 IN ERROR AFTER READING SYNC
5279						
5280	022612	012737	000000	002346	MOV #0,ECPATX	;LOAD EXPECTED PATTERN
5281	022620	012737	004066	002350	MOV #4066,ECPOSX	;LOAD EXPECTED POSITION
5282	022626	004437	030774		JSR R4,DWRTCK	;GO SIMULATE WRITE CHECK
5283	022632	000400			400	;NUMBER OF WORDS TO SIMULATE
5284	022634	104053			ERROR 53	;MR1 IN ERROR ON WRT CK OR ECC READ
5285	022636	104054			ERROR 54	;ECC ERROR IN WRITE CHECK OP
5286	022640	104055			ERROR 55	;CS1 ERROR AFTER WRT CHK OR ECC READ
5287	022642	104041			ERROR 41	;ECC REG INCORRECT AFTER ECC READ
5288	022644	104042			ERROR 42	;ERR IN ECC PAT CALC.
5289	022646	104043			ERROR 43	;ERR IN ECC POS COUNT
5290						
5291						
5292	022650	013700	002164		MOV T.BA,R0	;GET ACTUAL BA
5293	022654	162700	050440		SUB #IBUFF,R0	;SUB START VALUE OF BA
5294	022660	022700	000002		CMP #2,R0	;TEST IF CORRECT NUM OF WORDS XFER
5295	022664	001404			BEG 4\$	;YES - SKIP
5296	022666	012737	050442	002224	MOV #IBUFF+2,E.BA	;SET EXPECTED BA

N08

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 104
DZR6EA.P11 28-SEP-76 15:31 T36 WRITE CHECK OF A PARTIAL SECTOR

SEQ 0104

```

5297 022674 104046          ERROR 46          ;REPORT XFER LENGTH ERROR
5298
5299 022676 012701 000040    4$: MOV #8,*4,R1    ;SET COUNT TO END OF OPERATION
5300 022702 012762 000440    3$: MOV #DMD!MCLK,RKMR1(R2) ;RUN CLOCK
5301 022710 012762 000040    3$: MOV #DMD,RKMR1(R2)
5302 022716 005301          DEC R1              ;DEC COUNT
5303 022720 001370          BNE 3$              ;LOOP UNTIL ZERO
5304
5305 022722 004437 034154    1$: JSR R4,GETREG    ;GET 611 REGS
5306 022726 013737 002260    MOV L.CS1,E.CS1    ;GET EXPECTED CS1
5307 022734 042737 000001    BIC #GO,E.CS1     ;CLEAR GO BIT
5308 022742 052737 000200    BIS #RDY,E.CS1     ;SET READY BIT
5309 022750 023737 002160    CMP T.CS1,E.CS1    ;CHECK IF CS1 CORRECT
5310 022756 001401          BEQ 2$              ;YES - SKIP
5311 022760 104057          ERROR 57              ;"CS1 IN ERROR AFTER WRITE CHECK"
5312 022762
5313
5314
5315
5316
5317
5318
5319
5320
5321
5322
5323
5324
5325
5326
5327
5328
5329
5330
5331
5332
5333

```

```

*****
;TEST 37      WRITE CHECK ERROR (PART 1)
;
; CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
; PUT CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A WRITE
; CHECK OF 1 WORD TO AN RK06 IN 26 SECTOR FORMAT.
; CLINDER 0, HEAD 0, SECTOR 0.  CLOCK THROUGH SEEK
; AND DRIVE CLEAR MESSAGES.  SIMULATE A SECTOR PULSE,
; A GOOD HEADER, AND ONE WORD OF DATA CONSISTING OF 000000.
; WRITE CHECKED DATA IS 000001.  MAKE SURE WRITE CHECK ERROR SETS.
; REPEAT USING EACH OF THE FOLLOWING CONFIGURATIONS
; AS WRITE CHECKED DATA:
;
; 000002 000040 001000 020000
; 000004 000100 002000 040000
; 000010 000200 004000 100000
; 000020 000400 010000
;
*****

```

```

5334 022762 000004          TST37: SCOPE
5335 022764 012737 000001    001234      MOV #1,$TIMES    ;;DO 1 ITERATION
5336
5337 022772 005037 002352    CLR BSEDES        ;CLEAR BAD SECTOR DESIRED
5338 022776 005037 002354    CLR HIBITS        ;CLEAR HI ORDER BITS
5339 023002 005037 002344    CLR ECCSRC        ;CLEAR ECC SOURCE
5340 023006 004437 034026    JSR R4,LOADRK     ;LOAD "L" REGISTERS
5341 023012 000000          0                ;CYLINDER 0
5342 023014          .BYTE 0                ;SECTOR 0
5343 023015          .BYTE 0                ;TRACK 0
5344 023016 050440          IBUFF            ;BUFFER ADDRESS IBUFF
5345 023020 177777          -1              ;WORD COUNT -1
5346 023022 000031          WRTCHK           ;COMMAND WRTCHK
5347
5348 023024 012705 000020    MOV #16,R5        ;SET PATTERN COUNT
5349 023030 005037 051444    CLR OBUF         ;CLEAR FIRST WORD OF OBUF
5350 023034 012737 000001    MOV #1,IBUFF      ;SET IN FIRST PATTERN
5351 023042 032737 000001    BIT #BIT0,IBUFF   ;IS BIT 0 SET
5352 023050 001003          BNE 112$          ;YES - PATTERN SHOULD BE 0

```

5353	023052	012737	177777	051444		MOV	#-1,0BUFF	:ELSE PATTERN SHOULD BE ALL ONES
5354	023060	012737	023066	001110	1125:	MOV	#1105,\$LPERR	:SET LOCAL LOOP ON ERROR
5355								
5356	023066	012762	100000	000000	1108:	MOV	#CCLR,RKCS1(R2)	:CLEAR CONTROLLER
5357	023074	004437	034072			JSR	R4,OPSTR	:START THE OPERATION
5358	023100	004437	025100			JSR	R4,DISEEK	:SIMULATE IMPLIED SEEK
5359	023104	104004				ERROR	4	:CS1 MISCOMPARE
5360	023106	104005				ERROR	5	:MR1
5361	023110	104006				ERROR	6	:MR2
5362	023112	104007				ERROR	7	:MR3
5363								
5364	023114	004437	025062			JSR	R4,DSECTR	:SIMULATE SECTOR PULSE
5365								
5366	023120	004437	025504			JSR	R4,DRSYNC	:SIMULATE HEADER PREAMBLE
5367	023124	104010				ERROR	10	:MR1 CONTENTS IN ERROR
5368	023126	104011				ERROR	11	:CS1 CONTENTS IN ERROR
5369								
5370	023130	004437	026042			JSR	R4,DHDCMP	:SIMULATE HEADER SEARCH
5371	023134	104012				ERROR	12	:MR1 CONTENTS IN ERROR
5372	023136	104013				ERROR	13	:CS1 IN ERROR AFTER SEARCH
5373	023140	104014				ERROR	14	:RKDCYL OR RKDA IN ERROR AFTER SEARCH
5374								
5375	023142	004437	030276			JSR	R4,DRGPSN	:GO READ GAP AND SYNC
5376	023146	104015				ERROR	15	:MR1 IN ERROR READING GAP
5377	023150	104016				ERROR	16	:CS1 IN ERROR AFTER READING GAP
5378	023152	104032				ERROR	32	:MR1 IN ERROR READING SYNC
5379	023154	104033				ERROR	33	:CS1 IN ERROR AFTER READING SYNC
5380								
5381	023156	012737	000000	002346		MOV	#0,ECPATX	:LOAD EXPECTED PATTERN
5382	023164	012737	004066	002350		MOV	#4066,ECP0SX	:LOAD EXPECTED POSITION
5383	023172	004437	030774			JSR	R4,DWATCK	:GO SIMULATE WRITE CHECK
5384	023176	000001				1		:NUMBER OF WORDS TO SIMULATE
5385	023200	104053				ERROR	53	:MR1 IN ERROR ON WRT CK OR ECC READ
5386	023202	104054				ERROR	54	:ECC ERROR IN WRITE CHECK OP
5387	023204	104055				ERROR	55	:CS1 ERROR AFTER WRT CHK OR ECC READ
5388	023206	104041				ERROR	41	:ECC REG INCORRECT AFTER ECC READ
5389	023210	104042				ERROR	42	:ERR IN ECC PAT CALC.
5390	023212	104043				ERROR	43	:ERR IN ECC POS COUNT
5391								
5392	023214	012701	000040			MOV	#9,*4,R1	:SET COUNT
5393	023220	012762	000440	000026	15:	MOV	#DMD,MCLV,RKMR1(R2)	:RUN CLOCK
5394	023226	012762	000040	000026		MOV	#DMD,RKMR1(R2)	
5395	023234	005301				DEC	R1	:DEC COUNT
5396	023236	001370				BNE	15	:LOOP UNTIL ZERO
5397								
5398	023240	004437	034154			JSR	R4,GETREG	:GET REGS
5399	023244	032737	040000	002170		BIT	#WCE,T.CS2	:TEST IF WCE SET
5400	023252	001012				BNE	25	:YES - SKIP
5401	023254	013737	002170	001162		MOV	T.CS2,\$REG0	:SET CS2 FOR REPORT
5402	023262	013737	050440	001164		MOV	IBUFF,\$REG1	:SET WORD READ
5403	023270	013737	051444	001166		MOV	OBUFF,\$REG2	:SET WORD FROM BUFFER
5404	023276	104060				ERROR	60	:WCE FAILED TO SET
5405								
5406	023300	104415			25:	SCOP1		:LOCAL LOOP ON ERROR
5407								
5408	023302	006337	050440			ASL	IBUFF	:SHIFT FOR NEXT TEST PATTERN

C09

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MC-11-DZB6E-A MACY11 27 (1006) 05-OCT-76 09:12 PAGE 106
 DZB6E.A.P11 28-SEP-76 15:31 T37 WRITE CHECK ERROR (PART 1)

SEQ 0106

```

5409 023306 005305 DEC RS :DEC PATTERN COUNT
5410 023310 001266 BNE 1105 :LOOP UNTIL ALL PATTERNS TESTED

*****
*ES* 40 WRITE CHECK ERROR (PART 2)
*
* CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
* PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
* CHECK OF 1 WORD TO AN RK06 IN 26 SECTOR FORMAT
* CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
* AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE
* A GOOD HEADER, AND ONE WORD OF DATA CONSISTING OF 177777.
* WRITE CHECKED DATA IS 177776. MAKE SURE WRITE
* CHECK ERROR SETS. REPEAT USING EACH OF THE FOLLOWING
* CONFIGURATION AS WRITE CHECKED DATA:
*
* 177775 177737 176777 157777
* 177773 177677 175777 137777
* 177767 177577 173777 077777
* 177757 177377 167777
*
*****
*ES* 40: SCOPE
MOV #1,5TIMES ::DO 1 ITERATION

023312 000004 CLR BSEDES :CLEAR BAD SECTOR DESIRED
023314 012737 000001 001234 CLR HIBITS :CLEAR HI ORDER BITS
023322 005037 002352 CLR ECCSRC :CLEAR ECC SOURCE
023326 005037 002354 JSR R4,LOADRK :LOAD "L" REGISTERS
023332 005037 002344 0 CYLINDER 0
023336 004437 034026 0 SECTOR 0
023342 000000 0 TRACK 0
023344 000 0 BUFFER ADDRESS Ibuff
023346 000 0 WORD COUNT -1
023348 050440 -1 COMMAND WATCHK
023350 177777
023352 000031

023354 012705 000020 MOV #16,RS :SET PATTERN COUNT
023360 005037 051444 CLR OBUFF :CLEAR FIRST WORD OF OBUFF
023364 012737 177776 050440 MOV #177776,IBUFF :SET IN FIRST PATTERN
023372 032737 000001 050440 BIT #BIT0,IBUFF :IS BIT 0 SET
023400 001003 BNE 1125 :YES - PATTERN SHOULD BE 0
023402 012737 177777 051444 MOV #-1,OBUFF :ELSE PATTERN SHOULD BE ALL ONES
023410 012737 023416 001110 1125: MOV #1105,$LPERA :SET LOCAL LOOP ON ERROR

023416 012762 100000 000000 1105: MOV #CLR,RKCS1(R2) :CLEAR CONTROLLER
023424 004437 034072 JSR R4,OPSTAT :START THE OPERATION
023430 004437 025100 JSR R4,DISEEK :SIMULATE IMPLIED SEEK
023434 104004 ERROR 4 :CS1 MISCOMPARE
023436 104005 ERROR 5 :MR1
023440 104006 ERROR 6 :MR2
023442 104007 ERROR 7 :MR3

023444 004437 025062 JSR R4,DSECTR :SIMULATE SECTOR PULSE

023450 004437 025504 JSR R4,DRSYNC :SIMULATE HEADER PREAMBLE
023454 104010 ERROR 10 :MR1 CONTENTS IN ERROR

```

023456	104011			ERROR	11	:CSI CONTENTS IN ERROR
023460	004437	026242		JSR	R4,DHOCMP	:SIMULATE HEADER SEARCH
023464	004437			ERROR	12	:MRI CONTENTS IN ERROR
023468	004437			ERROR	13	:CSI IN ERROR AFTER SEARCH
023472	004437			ERROR	14	:RKDCYL OR RKDA IN ERROR AFTER SEARCH
023476	004437	030276		JSR	R4,DRGPSN	:GO READ GAP AND SYNC
023480	004437			ERROR	15	:MRI IN ERROR READING GAP
023484	004437			ERROR	16	:CSI IN ERROR AFTER READING GAP
023488	004437			ERROR	32	:MRI IN ERROR READING SYNC
023492	004437			ERROR	33	:CSI IN ERROR AFTER READING SYNC
023506	004437	000000	002346	MOV	#0,ECPATX	:LOAD EXPECTED PATTERN
023510	004437	004066	002350	MOV	#4066,ECPOSX	:LOAD EXPECTED POSITION
023514	004437	000000		JSR	R4,DWATCH	:GO SIMULATE WRITE CHECK
023518	004437			ERROR	53	:NUMBER OF WORDS TO SIMULATE
023522	004437			ERROR	54	:MRI IN ERROR ON WRT CK OR ECC READ
023526	004437			ERROR	55	:ECC ERROR IN WRITE CHECK OP
023530	004437			ERROR	41	:CSI ERROR AFTER WRT CHK OR ECC READ
023534	004437			ERROR	42	:ECC REG INCORRECT AFTER ECC READ
023538	004437			ERROR	43	:ERR IN ECC PAT CALC
023542	004437			ERROR	43	:ERR IN ECC POS COUNT
023546	012701	000040		MOV	#8,*4,R1	:SET COUNT
023550	012762	000440	000026	MOV	#DMD,MCLK,RKMR1,R2	:RUN CLOCK
023554	012762	000040	000026	MOV	#DMD,RKMR1,R2	
023558	005301			DEC	R1	:DEC COUNT
023562	001370			BNE	18	:LOOP UNTIL ZERO
023570	004437	034154		JSR	R4,GETREG	:GET REGS
023574	032737	040000	002170	BIT	#ACE,T.CS2	:TEST IF WCE SET
023602	001012			BNE	28	:YES - SKIP
023604	013737	002170	001162	MOV	T.CS2,\$REG0	:SET CS2 FOR REPORT
023612	013737	050440	001164	MOV	IBUFF,\$REG1	:SET WORD READ
023620	013737	051444	001166	MOV	OBUFF,\$REG2	:SET WORD FROM BUFFER
023626	104060			ERROR	60	:WCE FAILED TO SET
023630	104415		28:	SCOP1		:LOCAL LOOP ON ERROR
023632	006337	050440		ASL	IBUFF	:SHIFT FOR NEXT TEST PATTERN
023636	005537	050440		ADC	IBUFF	:PUT CARRY BACK IN
023642	005305			DEC	R5	:DEC PATTERN COUNT
023644	001264			BNE	1108	:LOOP UNTIL ALL PATTERNS TESTED

 *TEST 41 WRITE CHECK ERROR (PART 3)

 *
 * CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.
 * PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
 * CHECK OF 1 WORD TO AN RK06 IN 24 SECTOR FORMAT.
 * CYLINDER 0, HEAD 0, SECTOR 0. CLOCK THROUGH SEEK
 * AND DRIVE CLEAR MESSAGES. SIMULATE A SECTOR PULSE.
 * A GOOD HEADER, AND ONE WORD OF DATA CONSISTING
 * OF 200000. WRITE CHECK DATA IS 000000. MAKE SURE
 * WRITE CHECK ERROR SETS. REPEAT USING 400000 AS

E09

RK611: DISKLESS CONTROLLER DIAGNOSTIC: PS MC-11-D2R6E-A MACY11 27 1006 05-OCT-76 09:12 PAGE 108
 D2R6E-A.F11 28-SEP-76 15.31 741 WRITE CHECK ERROR (PART 3)

SEQ 0108

```

5521:          : SIMULATED DATA.
5522:          :
5523:          :*****
5524:          :
5525: 023646 000024 000001 001234 5541: SCOPE
5526: 023650 012737 000001 001234 MOV #1,STIMES ;:DO 1 ITERATION
5527:
5528: 023656 005037 002352 CLR BSEDES ;CLEAR BAD SECTOR DESIRED
5529: 023662 005037 002344 CLR ECCSRC ;CLEAR ECC SOURCE
5530: 023666 004437 034026 JSR R4,LOADRK ;LOAD "L" REGISTERS
5531: 023672 000000 0 CYLINDER 0
5532: 023674 000 .BYTE 0 ;SECTOR 0
5533: 023675 000 .BYTE 0 ;TRACK 0
5534: 023676 050440 IBUFF ;BUFFER ADDRESS IBUFF
5535: 023700 000000 -1 ;WORD COUNT -1
5536: 023702 010031 WRTCHK!CFMT ;COMMAND WRTCHK!CFMT
5537:
5538: 023704 012737 040000 002354 MOV #40000,HIBITS ;SET HIBITS FOR PATTERN OF 200000
5539: 023712 005037 050440 CLR IBUFF ;SET LOW BITS FOR READ
5540: 023716 005037 051444 CLR OBUFF ;SET 0 IN BUFFER
5541: 023722 012737 023730 001110 1125: MOV #1105,$LPERR ;SET LOCAL LOOP ON ERROR
5542: 023730 012762 100000 000000 1105: MOV #0CLR,RKCSI(R2) ;CLEAR CONTROLLER
5543:
5544: 023736 012700 002000 MOV #2000,R0 ;SET STALL COUNT
5545: 023742 005300 45: DEC R0 ;DEC COUNT
5546: 023744 001376 BNE 45 ;LOOP UNTIL ZERO
5547:
5548: 023746 004437 034072 JSR R4,OPSTRT ;START THE OPERATION
5549: 023752 004437 025100 JSR R4,DISEEK ;SIMULATE IMPLIED SEEK
5550: 023756 104004 ERROR 4 ;CS1 MISCOMPARE
5551: 023760 104005 ERROR 5 ;MR1
5552: 023762 104006 ERROR 6 ;MR2
5553: 023764 104007 ERROR 7 ;MR3
5554:
5555: 023766 004437 025062 JSR R4,DSECTR ;SIMULATE SECTOR PULSE
5556:
5557: 023772 004437 025504 JSR R4,DRSYNC ;SIMULATE HEADER PREAMBLE
5558: 023776 104010 ERROR 10 ;MR1 CONTENTS IN ERROR
5559: 024000 104011 ERROR 11 ;CS1 CONTENTS IN ERROR
5560:
5561: 024002 004437 026042 JSR R4,DHDCMP ;SIMULATE HEADER SEARCH
5562: 024006 104012 ERROR 12 ;MR1 CONTENTS IN ERROR
5563: 024010 104013 ERROR 13 ;CS1 IN ERROR AFTER SEARCH
5564: 024012 104014 ERROR 14 ;RKDCYL OR RKDA IN ERROR AFTER SEARCH
5565:
5566: 024014 004437 030276 JSR R4,DRGPSN ;GO READ GAP AND SYNC
5567: 024020 104015 ERROR 15 ;MR1 IN ERROR READING GAP
5568: 024022 104016 ERROR 16 ;CS1 IN ERROR AFTER READING GAP
5569: 024024 104032 ERROR 32 ;MR1 IN ERROR READING SYNC
5570: 024026 104033 ERROR 33 ;CS1 IN ERROR AFTER READING SYNC
5571:
5572: 024030 012737 000000 002346 MOV #0,ECPATX ;LOAD EXPECTED PATTERN
5573: 024036 012737 005066 002350 MOV #5066,ECP0SX ;LOAD EXPECTED POSITION
5574: 024044 004437 030774 JSR R4,DWRTCK ;GO SIMULATE WRITE CHECK
5575: 024050 000001 1 ;NUMBER OF WORDS TO SIMULATE
5576: 024052 104053 ERROR 53 ;MR1 IN ERROR ON WRT CK OR ECC READ
5577: 024054 104054 ERROR 54 ;ECC ERROR IN WRITE CHECK CP

```

F09

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZ6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 109
 DZ6EA.P11 28-SEP-76 15:31 T4: WRITE CHECK ERROR (PART 3)

SEQ 0109

```

5577 024056 104055          ERROR 55          :CS1 ERROR AFTER WRT CHK OR ECC READ
5578 024060 104041          ERROR 41          :ECC REG INCORRECT AFTER ECC READ
5579 024062 104042          ERROR 42          :ERR IN ECC PAT CALC.
5580 024064 104043          ERROR 43          :ERR IN ECC POS COUNT
5581
5582 024066 012701 000040          MOV  #8,*4,R1          :SET COUNT
5583 024072 012762 000440 000026 1S:  MOV  #DMD!MCLK,RKMR1(R2) :RUN CLOCK
5584 024100 012762 000040 000026          MOV  #DMD,RKMR1(R2)
5585 024106 005301          DEC  R1          :DEC COUNT
5586 024110 001370          BNE  1S          :LOOP UNTIL ZERO
5587
5588 024112 004437 034154          JSR  R4,GETREG          :GET REGS
5589 024116 032737 040000 002170          BIT  #WCE,T.CS2          :TEST IF WCE SET
5590 024124 001025          BNE  2S          :YES - SKIP
5591 024126 013737 002170 001162          MOV  T.CS2,$REG0          :SET CS2 FOR REPORT
5592 024134 013737 050440 001164          MOV  Ibuff,$REG1          :SET WORD READ
5593 024142 013737 051444 001166          MOV  Obuff,$REG2          :SET WORD FROM BUFFER
5594 024150 005037 001170          CLR  $REG3          :SET UP FOR REPORT
5595 024154 013737 002354 001172          MOV  HIBITS,$REG4
5596 024162 012700 000016          MOV  #14,R0
5597 024166 006037 001172          3S:  ROR  $REG4
5598 024172 005300          DEC  R0
5599 024174 001374          BNE  3S
5600 024176 104063          ERROR 63          :WCE FAILED TO SET
5601
5602 024200 104415          2S:  SCOP1          :LOCAL LOOP ON ERROR
5603
5604 024202 023727 002354 100000          CMP  HIBITS,#100000          :TEST IF SECOND PATTERN DONE
5605 024210 001404          BEQ  5S          :YES - EXIT
5606 024212 012737 100000 002354          MOV  #100000,HIBITS          :SET FOR PATTERN 400000
5607 024220 000643          BR   110S          :LOOP UNTIL ALL PATTERNS TESTED
5608
5609
5610
5611
5612
5613
5614
5615
5616
5617
5618
5619
5620
5621
5622
5623 024222 000004          *****
5624 024224 012737 000001 001234 1ST42: SCOPE *****
5625
5626 024232 005037 002354          MOV  #1,$TIMES          :DO 1 ITERATION
5627 024236 005037 002352          CLR  HIBITS          :CLEAR HI BITS
5628 024242 004437 034026          CLR  BSEDES          :CLEAR BAD SECTOR DESIRED
5629 024246 000000          JSR  R4,LOADRK          :LOAD "L" REGISTERS
5630 024250 000          0          :CYLINDER 0
5631 024251 000          .BYTE 0          :SECTOR 0
5632 024252 050440          .BYTE 0          :TRACK 0
          Ibuff          :BUFFER ADDRESS Ibuff

```

G09

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 110
 DZR6EA.P11 28-SEP-76 15:31 T42 WRITE CHECK ERROR (PART 4)

SEQ 0110

5633	024254	177400				-400		;WORD COUNT -400
5634	024256	000031				WRTCHK		;COMMAND WRTCHK
5635								
5636	024260	004437	033566			JSR	R4,CLRIBF	;GO CLEAR INPUT BUFFER
5637								
5638	024264	004437	033612			JSR	R4,BLDDAT	;GO BUILD DATA
5639	024270	000005				S		;PATTERN 5
5640								
5641	024272	012700	000400			MOV	#400,R0	;SET A COUNT FOR BUFFER SIZE
5642	024276	012701	050440			MOV	#IBUFF,R1	;SET ADDRESS OF IBUFF
5643	024302	012703	051444			MOV	#OBUFF,R3	;SET ADDRESS OF OBUFF
5644	024306	012321			75\$:	MOV	(R3)+,(R1)+	;MOV PATTERN TO IBUFF
5645	024310	005300				DEC	R0	;DEC COUNT
5646	024312	001375				BNE	75\$	;LOOP UNTIL PATTERN IS MOVED
5647	024314	012700	000530			MOV	#530,R0	;SET INDEX INTO BUFFER
5648								
5649	024320	012737	126073	052444		MOV	#126073,OBUFF+1000	;STORE ECC WORDS
5650	024326	012737	151052	052446		MOV	#151052,OBUFF+1002	
5651	024334	012737	000001	002344		MOV	#1,ECCSRC	;SET SOURCE TO INDICATE ECC IN BUFFER
5652	024342	032760	000100	051444		BIT	#BIT6,OBUFF(R0)	;TEST IF BIT SET
5653	024350	001004				BNE	100\$	;YES - SKIP
5654	024352	052760	000100	051444		BIS	#BIT6,OBUFF(R0)	;ELSE SET BIT
5655	024360	000403				BR	101\$	
5656	024362	042760	000100	051444	100\$:	BIC	#BIT6,OBUFF(R0)	;CLEAR BIT
5657	024370				101\$:			
5658								
5659	024370	012762	100000	000000	110\$:	MOV	#CCLR,RKCS1(R2)	;CLEAR CONTROLLER
5660	024376	004437	034072			JSR	R4,OPSTRT	;START THE OPERATION
5661	024402	004437	025100			JSR	R4,DISEEK	;SIMULATE IMPLIED SEEK
5662	024406	104004				ERROR	4	;CS1 MISCOMPARE
5663	024410	104005				ERROR	5	;MR1
5664	024412	104006				ERROR	6	;MR2
5665	024414	104007				ERROR	7	;MR3
5666								
5667	024416	004437	025062			JSR	R4,DSECTR	;SIMULATE SECTOR PULSE
5668								
5669	024422	004437	025504			JSR	R4,DRSYNC	;SIMULATE HEADER PREAMBLE
5670	024426	104010				ERROR	10	;MR1 CONTENTS IN ERROR
5671	024430	104011				ERROR	11	;CS1 CONTENTS IN ERROR
5672								
5673	024432	004437	026042			JSR	R4,DHDCMP	;SIMULATE HEADER SEARCH
5674	024436	104012				ERROR	12	;MR1 CONTENTS IN ERROR
5675	024440	104013				ERROR	13	;CS1 IN ERROR AFTER SEARCH
5676	024442	104014				ERROR	14	;RKDCYL OR RKDA IN ERROR AFTER SEARCH
5677								
5678	024444	004437	030276			JSR	R4,DRGPSN	;GO READ GAP AND SYNC
5679	024450	104015				ERROR	15	;MR1 IN ERROR READING GAP
5680	024452	104016				ERROR	16	;CS1 IN ERROR AFTER READING GAP
5681	024454	104032				ERROR	32	;MR1 IN ERROR READING SYNC
5682	024456	104033				ERROR	33	;CS1 IN ERROR AFTER READING SYNC
5683								
5684	024460	012737	000000	002346		MOV	#0,ECPATX	;LOAD EXPECTED PATTERN
5685	024466	012737	004066	002350		MOV	#4066,ECPOSX	;LOAD EXPECTED POSITION
5686	024474	004437	030774			JSR	R4,DWRTCK	;GO SIMULATE WRITE CHECK
5687	024500	000310				310		;NUMBER OF WORDS TO SIMULATE
5688	024502	104053				ERROR	53	;MR1 IN ERROR ON WRT CK OR ECC READ

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 111
 DZR6EA.F11 28-SEP-76 15:31 T42 WRITE CHECK ERROR (PART 4)

RK611 DISKLESS CONTROLLER DIA
DZR6EA.F11 28-SEP-76 15:31

T42

WRITE CHECK ERROR (PART 4)

Address	Hex	Hex	Hex	Hex	Label	Instruction	Comment
5689	024504	104054				ERROR	54 ;ECC ERROR IN WRITE CHECK OP
5690	024506	104055				ERROR	55 ;CS1 ERROR AFTER WRT CHK OR ECC REAC
5691	024510	104041				ERROR	41 ;ECC REG INCORRECT AFTER ECC REAC
5692	024512	104042				ERROR	42 ;ERR IN ECC PAT CALC.
5693	024514	104043				ERROR	43 ;ERR IN ECC POS COUNT
5694							
5695							
5696	024516	012701	000020			MOV	#4.*4.,R1 ;SET COUNT TO END OF OPERATION
5697	024522	012762	000440	000026	1\$:	MOV	#DMD!MCLK,RKMR1(R2) ;RUN CLOCK
5698	024530	012762	000040	000026		MOV	#DMD,RKMR1(R2)
5699	024536	005301				DEC	R1 ;DEC COUNT
5700	024540	001370				BNE	1\$;LOOP UNTIL ZERO
5701							
5702	024542	004437	034154			JSR	R4,GETREG ;GET '611 REGS
5703	024546	032737	040000	002170		BIT	#WCE,T.CS2 ;TEST IF WRITE CHECK SET
5704	024554	001013				BNE	2\$;YES - SKIP
5705	024556	013737	002170	001162		MOV	T.CS2,\$REG0 ;SET UP FOR REPORT
5706	024564	016037	050440	001164		MOV	IBUFF(R0),\$REG1 ;BUFFER WORD
5707	024572	016037	051444	001166		MOV	OBUFF(R0),\$REG2 ;SIMULATED WORD
5708	024600	104060				ERROR	60 ;WCE NOT SET ERROR
5709	024602	000423				BR	4\$;EXIT
5710							
5711	024604	012737	050442	002224	2\$:	MOV	#IBUFF+2,E.BA ;SET UP EXPECTED BA
5712	024612	060037	002224			ADD	R0,E.BA ;ADD IN THE NUMBER OF WORDS TO THE ERROR
5713	024616	023737	002224	002164		CMP	E.BA,T.BA ;CHECK IF CORRECT
5714	024624	001402				BEQ	3\$;YES - SKIP
5715	024626	104061				ERROR	61 ;BUS ADDRESS ERROR
5716	024630	000410				BR	4\$;EXIT
5717							
5718	024632	012737	177655	002222	3\$:	MOV	#-123,E.WC ;SET EXPECTED WORD COUNT
5719	024640	023737	002162	002222		CMP	T.WC,E.WC ;CHECK IF CORRECT
5720	024646	001401				BEQ	4\$;YES - SKIP
5721	024650	104062				ERROR	62 ;WORD COUNT ERROR
5722							
5723	024652				4\$:		
5724							

```

5725      .SBTTL  END OF PASS ROUTINE
5726
5727      ;;*****
5728      ;;*INCREMENT THE PASS NUMBER ($PASS)
5729      ;;*TYPE "END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYY"
5730      ;;*WHERE XXXXX AND YYYYY ARE DECIMAL NUMBERS
5731      ;;*IF THERES A MONITOR GO TO IT
5732      ;;*IF THERE ISN'T JUMP TO NEWPAS
5733
5734      024652      $EOP:
5735      024652      000004      SCOPE
5736      024654      005037      001102      CLR      $TSTNM      ;;ZERO THE TEST NUMBER
5737      024660      005037      001234      CLR      $TIMES      ;;ZERO THE NUMBER OF ITERATIONS
5738      024664      005237      001256      INC      $PASS      ;;INCREMENT THE PASS NUMBER
5739      024670      042737      100000      001256      BIC      #100000,$PASS      ;;DON'T ALLOW A NEG. NUMBER
5740      024676      005327      DEC      (PC)+      ;;LOOP?
5741      024700      000001      $EOPCT: .WORD      1
5742      024702      003063      BGT      $DOAGN      ;;YES
5743      024704      012737      MOV      (PC)+,2(PC)+      ;;RESTORE COUNTER
5744      024706      000001      $ENDCT: .WORD      1
5745      024710      024700      $EOPCT
5746      024712      104401      024720      TYPE      65$      ;;TYPE ASCIZ STRING
5747      024716      000407      BR      64$      ;;GET OVER THE ASCIZ
5748
5749      024736      013746      001256      ;;65$: .ASCIZ      <12><15>/END PASS #/
5750      024736      013746      001256      64$: MOV      $PASS,-(SP)      ;;SAVE $PASS FOR TYPEOUT
5751
5752      024742      104405      024752      TYPDS      ;;TYPE PASS NUMBER
5753      024744      104401      024752      TYPE      67$      ;;GO TYPE--DECIMAL ASCII WITH SIGN
5754      024750      000421      BR      66$      ;;TYPE ASCIZ STRING
5755      025014      013746      001112      ;;67$: .ASCIZ      / TOTAL ERRORS SINCE LAST REPORT /
5756      025014      013746      001112      66$: MOV      $ERTTL,-(SP)      ;;SAVE $ERTTL FOR TYPEOUT
5757
5758      025020      104405      001245      TYPDS      ;;TOTAL NUMBER OF ERRORS
5759      025022      104401      001112      TYPE      $CRLF      ;;GO TYPE--DECIMAL ASCII WITH SIGN
5760      025026      005037      001112      CLR      $ERTTL      ;;TYPE CARRIAGE RETURN, LINE FEED
5761      025032      013700      000042      $GET42: MOV      2*42,RO      ;;CLEAR ERROR TOTAL
5762      025036      001405      BEQ      $DOAGN      ;;GET MONITOR ADDRESS
5763      025040      000005      RESET      ;;BRANCH IF NO MONITOR
5764      025042      004710      $ENDAD: JSR      PC,(RO)      ;;CLEAR THE WORLD
5765      025044      000240      NOP      ;;GO TO MONITOR
5766      025046      000240      NOP      ;;SAVE ROOM
5767      025050      000240      NOP      ;;FOR
5768      025052      000137      $DOAGN: JMP      2(PC)+      ;;ACT11
5769      025054      003376      $RTNAD: .WORD      NEWPAS      ;;RETURN
5770      025056      377      000      $ENULL: .BYTE      -1,-1,0      ;;NULL CHARACTER STRING
5771
5772      025062      012762      000140      000026      .SBTTL  DIAGNOSTIC MODE SECTOR PULSE
5773      025062      012762      000040      000026      ;;*
5774      025062      012762      000040      000026      DSECTR: MOV      #DMD:MSP,RKMRI(R2)      ;;SET SECTOR PULE
5775      025062      012762      000040      000026      MOV      #DMD,RKMRI(R2)      ;;RESET SECTOR PULSE
5776      025062      012762      000040      000026      RTS      R4
5777
5778      025076      000204
5779
5780

```

5781			.SBTTL	DIAGNOSTIC MODE IMPLIED SEEK
5782			:*	THE CONTROLLER IS CLOCKED THROUGH THE SEEK MESSAGES,
5783			:*	GETTING STATUS, AND CLEARING THE DRIVE. SINCE THIS REQUIRES A
5784			:*	DIFFERENT COUNT FOR WRITE DATA AND READ DATA/WRITE CHECK,
5785			:*	THE ROUTINE LOOKS AT THE COMMAND THAT WAS
5786			:*	STARTED (L.CS1) AND SETS THE CLOCK COUNT ACCORDINGLY.
5787			:*	
5788			:*	AT THE END OF THE IMPLIED SEEK THE CONTENTS OF CS1, MR1, MR2
5789			:*	AND MR3 ARE CHECKED TO VERIFY THE OPERATION. ALL DISCREPENCIES
5790			:*	ARE REPORTED IN THE ORDER LISTED.
5791			:*	
5792			:*	CALL:
5793			:*	JSR R4,DISEEK
5794			:*	ERROR 4 ;CS1 MISCOMPARE ERROR
5795			:*	ERROR 5 ;MR1 MISCOMPARE ERROR
5796			:*	ERROR 6 ;MR2 MISCOMPARE ERROR
5797			:*	ERROR 7 ;MR3 MISCOMPARE ERRON
5798				
5799	025100		DISEEK:	
5800	025100	010046	MOV	R0,-(SP) ;: PUSH R0 ON STACK
5801	025102	010346	MOV	R3,-(SP) ;: PUSH R3 ON STACK
5802	025104	032737 000002 002260	BIT	#BIT1,L.CS1 ;:CHECK IF WRITE DATA
5803	025112	001003	BNE	10\$;YES - SKIP
5804	025114	012700 000316	MOV	#51.*4+2,R0 ;:SET COUNT FOR READ OR WRITE CHECK
5805	025120	000412	BR	1\$
5806				
5807	025122	012700 004612	MOV	#66.*37.,R0 ;:COUNT FOR CLOCK THROUGH SILO LOAD
5808	025126	032737 010000 002260	BIT	#CFMT,L.CS1 ;:TEST IF 22 SECTOR MODE
5809	025134	001402	BEQ	11\$;NO - SKIP
5810	025136	012700 005222	MOV	#66.*41.,R0 ;:ELSE CHANGE FOR 18 BIT WORDS
5811	025142	062700 000016	ADD	#4*3+2,R0 ;:AND SHIFT REGISTER LOAD
5812				
5813	025146	012762 000440 000026	MOV	#DMD!MCLK,RKMR1(R2) ;:SET CLOCK
5814	025154	012762 000040 000026	MOV	#DMD,RKMR1(R2) ;:CLEAR CLOCK
5815	025162	005300	DEC	R0 ;:LOOP UNTIL R0 IS ZERO
5816	025164	001370	BNE	1\$
5817				
5818	025166	013703 002274	MOV	L.DCYL,R3 ;:GET DESIRED CYLINDER
5819	025172	004437 034242	JSR	R4,FSBLVV ;:REVERSE ORDER OF BITS
5820	025176	010337 002246	MOV	R3,E.MR2 ;:STORE AS EXPECTED MR2
5821	025202	113703 002267	MOVB	L.DT,R3 ;:GET DESIRED TRACK/SECTOR
5822	025206	042703 177400	BIC	#177400,R3 ;:CLEAR UNUSED BITS
5823	025212	012700 000005	MOV	#5,R0 ;:SET SHIFT COUNT
5824	025216	006303	ASL	R3 ;:SHIFT FOR TRACK ALIGNMENT
5825	025220	005300	DEC	R0 ;:DEC COUNT
5826	025222	001375	BNE	15\$;:LOOP UNTIL ZERO
5827	025224	153703 002266	BISB	L.DS,R3 ;:INSERT SECTOR NUMBER
5828	025230	032737 010000 002260	BIT	#CFMT,L.CS1 ;:TEST IF 18 BIT MODE
5829	025236	001402	BEQ	16\$;NO - SKIP
5830	025240	052703 001000	BIS	#BIT9,R3 ;:ELSE SET FORMAT BIT IN HDR WRD
5831	025244	004437 034242	JSR	R4,FSBLVV ;:REVERSE BIT ORDER
5832	025250	010337 002250	MOV	R3,E.MR3 ;:STORE AS EXPECTED MR3
5833				
5834	025254	013737 002260 002220	MOV	L.CS1,E.CS1 ;:GET EXPECTED CS1
5835	025262	012737 022040 002244	MOV	#DMD!ECCW!MEWD,E.MR1 ;:SET UP EXPECTED MR1
5836	025270	004437 034154	JSR	R4,GETREG ;:GET RK REGS

K09

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 114
 DZR6EA.F11 28-SEP-76 15:31 DIAGNOSTIC MODE IMPLIED SEEK

SEQ 0114

```

5837
5838 025274 023737 002160 002220      CMP      T.CS1,E.CS1      ;CHECK CS1
5839 025302 001411                      BEQ      2$              ; OK - SKIP
5840 025304 012737 025326 001236      MOV      #2$,$ESCAPE    ;SET ESCAPE TO RETURN FOR
5841                                ;REMAINING TESTS
5842 025312 032777 001000 153620      BIT      #BIT9,$SWR      ;CHECK IF LOOP ON ERROR
5843 025320 001056                      BNE      9$              ;YES - SKIP
5844
5845 025322 010446                      4$:  MOV      R4,-(SP)      ;SET STACK FOR RETURN TO SUB-ROUTINE
5846 025324 000204                      RTS
5847
5848 025326 005724                      2$:  TST      (R4)+          ;BUMP TO NEXT ERROR CALL
5849 025330 023737 002244 002204      CMP      E.MR1,T.MR1      ;CHECK MR1
5850 025336 001410                      BEQ      3$              ;OK - SKIP
5851 025340 032777 001000 153572      BIT      #BIT9,$SWR      ;CHECK IF LOOP ON ERROR
5852 025346 001043                      BNE      9$              ;YES - SKIP
5853 025350 012737 025360 001236      MOV      #3$,$ESCAPE    ;SET ESCAPE FOR RETURN TO SUB-ROUTINE
5854 025356 000761                      BR       4$
5855
5856 025360 005724                      3$:  TST      (R4)+          ;SAME OPERATION AS ABOVE TO
5857 025362 023737 002246 002206      CMP      E.MR2,T.MR2      ;CHECK MR2
5858 025370 001410                      BEQ      5$
5859 025372 032777 001000 153540      BIT      #BIT9,$SWR
5860 025400 001026                      BNE      9$
5861 025402 012737 025412 001236      MOV      #5$,$ESCAPE
5862 025410 000744                      BR       4$
5863
5864 025412 005724                      5$:  TST      (R4)+          ;SAME OPERATION AS ABOVE TO
5865 025414 023737 002250 002210      CMP      E.MR3,T.MR3      ;CHECK MR3
5866 025422 001414                      BEQ      8$
5867 025424 032777 001000 153506      BIT      #BIT9,$SWR
5868 025432 001011                      BNE      9$
5869 025434 012737 025444 001236      MOV      #6$,$ESCAPE
5870 025442 000727                      BR       4$
5871
5872 025444 032777 001000 153466      6$:  BIT      #BIT9,$SWR      ;CHECK IF LOOP ON ERROR
5873 025452 001001                      BNE      9$              ;YES - SKIP
5874 025454 005724                      8$:  TST      (R4)+          ;BUMP TO NO ERROR RETURN
5875 025456 005037 001236      9$:  CLR      $ESCAPE      ;CLEAR ESCAPE
5876 025462 012603                      MOV      (SP)+,R3        ;POP STACK INTO R3
5877 025464 012600                      MOV      (SP)+,R0        ;POP STACK INTO R0
5878 025466 000204                      RTS      R4
5879
5880 025470 005037 001236      7$:  CLR      $ESCAPE      ;CLEAR ESCAPE
5881 025474 012706 001100      MOV      #STACK,SP      ;CLEAN OFF STACK
5882 025500 000177 153404      JMP      @SLPERR      ;GO TO LOOP START
5883

```

```

5884      .SBTTL  DIAGNOSTIC MODE READ SYNC PATTERN (PREAMBLE)
5885      ;*      THE 255 "0" BITS AND 1 "1" BIT ARE CLOCKED THROUTH THE READ.
5886      ;*      THE CONTENTS OF MR1 ARE MONITORED TO WATCH FOR AN ERROR.
5887      ;*      AT THE END OF THE SYNC READ, CS1 IS CHECKED TO INSURE IT
5888      ;*      DID NOT GET CHANGED.
5889      ;*
5890      ;*      CALL:
5891      ;*      JSR      R4,DRSYNC
5892      ;*      ERROR    10      ;MR1 CONTENTS IN ERROR

```

```

5893      ;*      ERROR 11      ;CS1 CONTENTS IN ERROR
5894      ;*
5895      ;*      IF AN ERROR IS DETECTED. THE SUBROUTINE WILL SET UP TO ABORT
5896      ;*      THE PRESENT TEST UNLESS LOOP ON ERROR IS SET.
5897
5898      025504      DRSYNC:
5899      025504      010046      MOV      RD, -(SP)      ;: PUSH RD ON STACK
5900      025506      005037      002326      CLR      BITCNT      ;: CLEAR BIT COUNT
5901      025512      005037      001236      CLR      $ESCAPE      ;: CLEAR ESCAPE
5902      025516      012700      000200      MOV      #128, RD      ;: SET COUNT TO CLOCK TO READ GATE
5903      025522      005037      002320      CLR      PR.BIT      ;: CLEAR PRESENT BIT
5904      025526      005037      002322      CLR      M1.BIT      ;: CLEAR PREVIOUS BIT
5905      025532      012737      022040      002244      MOV      #DMD!MEWD!ECCW.E.MR1 ;: SET UP EXPECTED MR1
5906
5907      025540      004737      033414      002204      3$:      JSR      PC,RDBIT      ;: GO READ A BIT
5908      025544      016237      000026      002204      MOV      RKMRI(R2),T.MR1 ;: GET MR1
5909      025552      023737      002204      002244      CMP      T.MR1,E.MR1      ;: CHECK IF IT IS CORRECT
5910      025560      001424      BEQ      2$      ;: YES - SKIP
5911
5912      025562      012737      025632      001236      MOV      #2$, $ESCAPE      ;: SET ESCAPE FOR CONTINUE TEST
5913
5914      025570      005037      001236      4$:      CLR      $ESCAPE      ;: CLEAR ESCAPE
5915      025574      032777      001000      153336      BIT      #BIT9,$SWR      ;: CHECK IF LOOP ON ERROR
5916      025602      001011      BNE      1$      ;: YES SKIP
5917      025604      013700      001254      MOV      $TESTN,RD      ;: ELSE SET UP TO SKIP TO NEXT TEST
5918      025610      006300      ASL      RD
5919      025612      016037      035270      001236      MOV      $SW08TB(RD),$ESCAPE
5920      025620      162737      000002      001236      SUB      #2,$ESCAPE
5921
5922      025626      012600      1$:      MOV      (SP)+,RD      ;: POP STACK INTO RD
5923      025630      000204      RTS      R4
5924
5925      025632      005237      002326      2$:      INC      BITCNT      ;: INCREMENT THE BIT COUNT
5926      025636      005300      DEC      RD      ;: LOOP UNTIL READY FOR READ GATE
5927      025640      001337      BNE      3$
5928
5929      025642      012700      000177      MOV      #127,RD      ;: SET COUNT FOR REST OF SYNC
5930      025646      012737      122040      002244      MOV      #DMD!MEWD!ECCW!RDGATE.E.MR1 ;: SET NEW EXPECTED MR1
5931
5932      025654      004737      033414      002204      5$:      JSR      PC,RDBIT      ;: GO READ BIT
5933      025660      016237      000026      002204      MOV      RKMRI(R2),T.MR1 ;: GET MR1
5934      025666      023737      002204      002244      CMP      T.MR1,E.MR1      ;: CHECK IF CORRECT
5935      025674      001404      BEQ      6$      ;: YES - SKIP
5936      025676      012737      025706      001236      MOV      #6$, $ESCAPE      ;: SET FOR CONTINUE TESTING
5937      025704      000445      BR      50$      ;: GO TO ERROR EXIT
5938      025706      005237      002326      6$:      INC      BITCNT      ;: BUMP BIT COUNTER
5939      025712      005300      DEC      RD      ;: LOOP UNTIL READY FOR SYNC 1
5940      025714      001357      BNE      5$
5941
5942      025716      013737      002320      002322      MOV      PR.BIT,M1.BIT ;: MOV PRESENT TO PREVIOUS
5943      025724      012737      000001      002320      MOV      #1,PR.BIT ;: SET SYNC 1 BIT
5944      025732      004737      033414      JSR      PC,RDBIT      ;: GO READ BIT
5945      025736      016237      000026      002204      MOV      RKMRI(R2),T.MR1 ;: GET MR1
5946      025744      023737      002204      002244      CMP      T.MR1,E.MR1      ;: CHECK IF CORRECT
5947      025752      001404      BEQ      7$      ;: YES - SKIP
5948

```

```

5949 025754 012737 025764 001236      MOV    #7$, $ESCAPE      ;SET RETURN FOR CONTINUE TEST
5950 025762 000416                      BR     50$              ;GO TO ERROR EXIT
5951
5952 025764 004437 034154      7$:   JSR    R4, GETREG      ;GO GET RK611 REGISTERS
5953 025770 005237 002326      INC    BITCNT          ;BUMP BIT COUNTER
5954 025774 005724              TST    (R4)+            ;BUMP TO NEXT ERROR CALL
5955 025776 013737 002260 002220  MOV    L.CS1, E.CS1      ;GET EXPECTED CS1
5956 026004 023737 002160 002220  CMP    T.CS1, E.CS1      ;CHECK IF CORRECT
5957 026012 001266                      BNE    4$              ;NO - SKIP
5958
5959 026014 005724              TST    (R4)+            ;BUMP TO NO ERROR RETURN
5960 026016 000703                      BR     1$              ;SKIP TO RETURN
5961
5962 026020 032777 001000 153112 50$:  BIT    #BIT9, $SWR      ;TEST IF LOOP ON ERROR
5963 026026 001403                      BEQ    51$              ;NO - SKIP
5964 026030 005037 001236      CLR    $ESCAPE          ;CLEAR FOR SKIP TO NEXT TEST
5965 026034 000674                      BR     1$
5966 026036 010446      51$:  MOV    R4, -(SP)        ;SET UP STACK FOR RETURN
5967 026040 000204                      RTS    R4
5968
5969      .SBTTL  DIAGNOSTIC MODE HEADER READ AND COMPARE
5970      ;* THIS ROUTINE SIMULATES THE THREE HEADER WORDS READ AND
5971      ;* COMPARE THAT IS PART OF ALL DATA TRANSFER OPERATIONS.
5972      ;*
5973      ;* THE EXPECTED HEADER IS CALCULATED FROM THE CYLINDER, TRACK,
5974      ;* AND SECTOR SPECIFIED IN THE "L" REGISTERS. THE READING OF THE
5975      ;* 3 WORDS IS DONE AND MR1 IS MONITORED TO INSURE IT DOES
5976      ;* NOT CHANGE. AT THE END OF THE READ/COMPARE, THE CONTENTS
5977      ;* OF RKCS1 IS CHECKED AND THE INCREMENT OF RKDCYL AND RKDA IS
5978      ;* VERIFIED. IF THE INCREMENT IS GOOD, THE "L" REGISTERS ARE
5979      ;* UPDATED TO THE NEW PACK ADDRESS IN CASE OF MULTI-SECTOR
5980      ;* OPERATIONS. THE BAD SECTOR ERROR DESIRED (BSEDES) SWITCH
5981      ;* IS TESTED AND IF IT IS NON-ZERO, THE 2ND AND 3RD HEADER
5982      ;* WORDS ARE CHANGED ACCORDING TO BSEDES CONTENTS. THIS
5983      ;* CAN BE USED TO FORCE HEADER TYPE ERROR.
5984      ;*
5985      ;* CALL:
5986      ;* JSR    R4, DHDCMP
5987      ;* ERROR 12      ;ERROR IN MR1
5988      ;* ERROR 13      ;ERROR IN CS1
5989      ;* ERROR 14      ;ERROR IN PACK ADDRESS
5990
5991      DHDCMP:
5992 026042 010046      MOV    R0, -(SP)      ;; PUSH R0 ON STACK
5993 026044 010146      MOV    R1, -(SP)      ;; PUSH R1 ON STACK
5994 026046 010346      MOV    R3, -(SP)      ;; PUSH R3 ON STACK
5995 026050 010546      MOV    R5, -(SP)      ;; PUSH R5 ON STACK
5996 026052 005037 002326  CLR    BITCNT          ;CLEAR BIT COUNTER
5997 026056 012700 000005  MOV    #5, R0          ;SET A SHIFT COUNT
5998 026062 113701 002267  MOVB   L.DT, R1          ;GET DESIRED TRACK
5999 026066 042701 177400  BIC    #177400, R1      ;CLEAR UNUSED BITS
6000
6001 026072 006301      1$:   ASL    R1              ;SHIFT TRACK TO ALIGN TO
6002 026074 005300      DEC    R0              ;HEADER WORD FORMAT
6003 026076 001375      BNE    1$
6004

```

6005	026100	153701	002266		BISB	L.DS,R1	; INSERT DESIRED SECTOR
6006	026104	052701	140000		BIS	#BIT15:BIT14,R1	; SET BS BITS
6007	026110	013700	002274		MOV	L.DCYL,R0	; GET DESIRED CYLINDER
6008	026114	032737	010000	002260	BIT	#CFMT,L.CS1	; TEST IF 22 SECTOR FORMAT
6009	026122	001402			BEQ	2\$	; NO - SKIP
6010	026124	052701	001000		BIS	#BIT9,R1	; ELSE SET FORMAT BIT
6011	026130	010003		2\$:	MOV	R0,R3	; COMPUTE VRC
6012	026132	010105			MOV	R1,R5	
6013	026134	040005			BIC	R0,R5	
6014	026136	040103			BIC	R1,R3	
6015	026140	050503			BIS	R5,R3	
6016	026142	005737	002352		TST	BSEDES	; TEST IF BSE DESIRED
6017	026146	001404			BEQ	23\$	; NO - SKIP
6018	026150	043701	002352		BIC	BSEDES,R1	; CLEAR BIT IN WORD 2
6019	026154	043703	002352		BIC	BSEDES,R3	; CORRECT HVRC
6020	026160			23\$:			
6021							
6022							
6023							
6024	026160	005037	001204		CLR	\$TMP1	; CLEAR A PASS COUNTER
6025	026164	012737	122040	002244	MOV	#DMD!MEWD!ECCW!RDGATE,E.MR1	; SET EXPECTED MR1
6026							
6027	026172	012705	000001		MOV	#BIT0,R5	; ESTABLISH BEGINNING OF DATA WORD
6028	026176	013737	002320	002322	MOV	PR.BIT,M1.BIT	; STORE PREVIOUS BIT
6029	026204	005037	002320		CLR	PR.BIT	; PRESET FOR PRESENT BIT 0
6030	026210	030500			BIT	R5,R0	; TEST IF BIT TO BE A ONE
6031	026212	001403			BEQ	3\$	; NO - SKIP
6032	026214	012737	000001	002320	MOV	#BIT0,PR.BIT	; ELSE CHANGE TO ONE
6033	026222	004737	033414		JSR	PC,RDBIT	; GO READ A BIT
6034	026226	016237	000026	002204	MOV	RKMR1(R2),T.MR1	; GET MR1 FOR TESTING
6035	026234	023737	002204	002244	CMP	T.MR1,E.MR1	; CHECK IF CORRECT
6036	026242	001431			BEQ	6\$	; YES - SKIP
6037	026244	012737	026326	001236	MOV	#6\$,\$ESCAPE	; SET RETURN TO CONTINUE TEST
6038	026252	000137	026756		JMP	50\$	
6039							
6040	026256	005037	001236		CLR	\$ESCAPE	; CLEAR TO JUMP TO NEXT TEST
6041	026262	032777	001000	152650	BIT	#BIT9,\$SWR	; CHECK IF LOOP ON ERROR
6042	026270	001011			BNE	5\$	; YES - SKIP
6043	026272	013705	001254		MOV	\$TESTN,R5	; GET CURRENT TEST NUMBER
6044	026276	006305			ASL	R5	; AND SET UP TO ESCAPE TO NEXT TEST
6045	026300	016537	035270	001236	MOV	\$SW08TB(R5),\$ESCAPE	
6046	026306	162737	000002	001236	SUB	#2,\$ESCAPE	
6047							
6048	026314						
6049	026314	012605		5\$:	MOV	(SP)+,R5	; POP STACK INTO R5
6050	026316	012603			MOV	(SP)+,R3	; POP STACK INTO R3
6051	026320	012601			MOV	(SP)+,R1	; POP STACK INTO R1
6052	026322	012600			MOV	(SP)+,R0	; POP STACK INTO R0
6053	026324	000204			RTS	R4	
6054							
6055	026326	005237	002326		INC	BITCNT	; BUMP BIT COUNT
6056	026332	005705		6\$:	TST	R5	; CHECK IF LAST BIT OF THIS WORD
6057	026334	100402			BMI	8\$	; HAS BEEN SIMULATED - YES - SKIP
6058	026336	006305			ASL	R5	; SHIFT TO NEXT BIT
6059	026340	000716			BR	7\$	; GO DO THIS BIT
6060							

B10

RK611: DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZ6E-A MACY11 271006 05-OCT-76 09:12 PAGE 118
 DZ6EA.P11 28-SEP-76 15:31 DIAGNOSTIC MODE HEADER READ AND COMPARE

SEQ 0118

```

6061 026342 005737 001204      8$:  TST      STMP1      :TEST IF PASS FOR 1ST WORD JUST DONE
6062 026346 001004              BNE      9$      :NO - SKIP
6063 026350 010100              MOV      R1,R0      :PUT WORD 2 IN R0 FOR SIMULATION
6064 026352 005237 001204      INC      STMP1      :BUMP PASS COUNT
6065 026356 000705              BR       11$
6066
6067 026360 023727 001204 000002 9$:  CMP      STMP1, #2      :JUST DONE WITH WORD 2?
6068 026366 002004              BGE      10$      :NO SKIP
6069 026370 010300              MOV      R3,R0      :PUT 3RD WORD IN R0
6070 026372 005237 001204      INC      STMP1      :BUMP PASS COUNT
6071 026376 000675              BR       11$
6072
6073 026400 023727 001204 000003 10$:  CMP      STMP1, #3      :JUST DONE WITH WORD 3?
6074 026406 002006              BGE      16$      :NO - SKIP
6075 026410 005000              CLR      R0      :PUT IN WORD 0 OF GAP
6076 026412 012705 100000      MOV      #BIT15,R5      :SET TO GIVE IT 1ST GAP BIT
6077 026416 005237 001204      INC      STMP1      :BUMP PASS COUNT
6078 026422 000665              BR       7$
6079
6080 026424 023727 001204 000004 16$:  CMP      STMP1, #4      :TEST IF READY FOR GAP
6081 026432 001445              BEQ      12$      :YES - SKIP
6082
6083      :
6084      : THE FOLLOWING BLOCK OF CODE COMPUTES THE NUMBER OF CLOCK
6085      : PULSES (WHICH ARE ACTUALLY CALLS TO RDBIT) THE OPERATION
6086      : BEING PERFORMED REQUIRES TO UPDATE THE ADDRESS COUNTERS
6087      : (RKDC AND RKDA). THIS CALCULATION IS BASED ON THE WHAT WAS
6088      : IN THESE COUNTERS WHEN THE OPERATION STARTED (L.DC AND L.DA).
6089
6089 026434 012701 000025      MOV      #25,R1      :PRESET R1 FOR 24 SECTOR MODE
6090 026440 032737 010000 002260      BIT      #CFMT,L.CS1      :IS IT 24 SECTOR MODE OPERATION?
6091 026446 001402              BEQ      22$      :YES - SKIP
6092 026450 012701 000023      MOV      #23,R1      :ELSE CHANGE FOR 22 SECTOR MODE
6093 026454 032737 002000 002260 22$:  BIT      #CDT,L.CS1      :RK06?
6094 026462 001403              BEQ      21$      :YES - SKIP
6095 026464 052701 002000      BIS      #BIT10,R1      :ELSE SET FOR MAX HEAD OF 4
6096 026470 000402              BR       20$      :SKIP
6097 026472 052701 001000      BIS      #BIT9,R1      :SET FOR MAX HEAD OF 2
6098 026476 023701 002266      CMP      L.DS,R1      :TEST IF TRACK/SECTOR ADDRESS WAS MAX
6099 026502 001003      BNE      17$      :NO - SKIP
6100 026504 012705 000100      MOV      #BIT6,R5      :SET FOR 8 BITS OF GAP TO UPDATE
6101 026510 000410              BR       19$
6102 026512 123701 002266      CMPB     L.DS,R1      :TEST IF SECTOR WAS MAX
6103 026516 001003      BNE      18$      :NO - SKIP
6104 026520 012705 000400      MOV      #BIT8,R5      :SET FOR 6 BITS OF GAP TO UPDATE
6105 026524 000402              BR       19$
6106 026526 012705 010000      MOV      #BIT12,R5      :SET TO GIVE 1ST 4 BITS OF GAP
6107 026532 005237 001204      INC      STMP1      :BUMP PASS COUNT
6108 026536 042737 100000 002244      BIC      #RDGATE,E.MR1      :CLEAR READ GATE
6109 026544 000614              BR       7$
6110      :
6111      : END OF BLOCK
6112
6112 026546 004437 034154      12$:  JSR      R4,GETREG      :GET RK611 REGISTERS
6113 026552 005724              TST      (R4)+      :BUMP TO NEXT ERROR RETURN
6114 026554 013737 002260 002220      MOV      L.CS1,E.CS1      :SET EXPECTED CS1
6115 026562 023737 002160 002220      CMP      T.CS1,E.CS1      :CHECK IF CORRECT
6116 026570 001232              BNE      4$      :NO SKIP

```

C10

PK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-D2R6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 119
 C2R6E.A.P11 29-SEP-76 15:31 DIAGNOSTIC MODE HEADER READ AND COMPARE

SEG 0119

```

6117
6118 026572 005724          TST      (R4)+          :BUMP TO NEXT ERROR RETURN
6119 026574 013737 002274 002240      MOV      L.DCYL,E.DCYL      :GET STARTING CYLINDER
6120 026602 013737 002266 002226      MOV      L.DA,E.DA          :TRACK/SECTOR
6121 026610 105237 002226          INCB      E.DA          :BUMP SECTOR
6122 026614 012700 000026          MOV      #26,R0          :PRESET FOR 26 SEC/TRACK
6123 026620 032737 010000 002260      BIT      #CFMT,L.CS1      :TEST IF IN 26 SECTOR MODE
6124 026626 001402          BEQ      13$          :YES - SKIP
6125 026630 012700 000024          MOV      #24,R0          :ELSE CHANGE FOR 24 SECTOR MODE
6126 026634 120037 002226          CMPB     R0,E.DA          :CHECK IF SECTOR TO LARGE
6127 026640 001023          BNE      15$          :NO - SKIP
6128 026642 105037 002226          CLRB     E.DA          :SET SECTOR TO 0
6129 026646 105237 002227          INCB      E.DA+1          :BUMP TRACK
6130 026652 012700 000003          MOV      #3,R0          :PRESET FOR 3 TRACKS (RK06)
6131 026656 032737 002000 002260      BIT      #COT,L.CS1      :TEST IF IT IS RK06
6132 026664 001402          BEQ      14$          :YES - SKIP
6133 026666 012700 000005          MOV      #5,R0          :ELSE CHANGE
6134 026672 123700 002227          CMPB     E.DA+1,R0      :CHECK IF TRACK INC TO LARGE
6135 026676 001004          BNE      15$          :NO - SKIP
6136
6137 026700 105037 002227          CLRB     E.DA+1          :CLEAR TRACK TO 0
6138 026704 005237 002240          INC      E.DCYL          :BUMP CYLINDER
6139
6140 026710 023737 002240 002200 15$:  CMP      E.DCYL,T.DCYL      :CHECK IF CYL OK
6141 026716 001004          BNE      25$          :NO - SKIP
6142 026720 023737 002226 002166      CMP      E.DA,T.DA          :CHECK IF DA OK
6143 026726 001402          BEQ      26$          :YES - SKIP
6144 026730 000137 026256          JMP      4$          :NO - SKIP
6145
6146 026734 013737 002240 002274 26$:  MOV      E.DCYL,L.DCYL      :STORE COMPUTED NEXT CYL
6147 026742 013737 002226 002266      MOV      E.DA,L.DA          :AND NEXT DA
6148
6149 026750 005724          TST      (R4)+          :BUMP TO NO ERROR RETURN
6150 026752 000137 026314          JMP      5$          :GO TO GOOD EXIT
6151
6152 026756 032777 001000 152154 50$:  BIT      #BIT9,DSWR      :TEST IF LOOP ON ERROR
6153 026764 001404          BEQ      51$          :NO - SKIP
6154 026766 005037 001236          CLR      $ESCAPE      :CLEAR TO LOOP ON ERROR
6155 026772 000137 026314          JMP      5$
6156
6157 026776 010446          51$:  MOV      R4,-(SP)          :SET STACK TO CONTINUE TEST
6158 027000 000204          RTS      R4
6159          .SBTTL      DIAGNOSTIC MODE GAP READ AND SYNC WRITE
6160          :*          THE READING OF THE GAP AND WRITING OF THE SYNC (OR PREAMBLE)
6161          :*          IS HANDLED IN THIS ROUTINE.
6162          :*
6163          :*          THE FIRST GAP BITS ARE PROCESSED IN THE HEADER COMPARE
6164          :*          ROUTINE AND THE REMAINDER ARE PROCESSED HERE. THEN A
6165          :*          CHECK IS MADE THAT WRITE GATE CAME ON AND 255 "0" BITS
6166          :*          AND A SINGLE "1" BIT IS WRITTEN.
6167          :*
6168          :*          CALL:
6169          :*          JSR      R4,DWGPN
6170          :*          ERROR 15      :MR1 IN ERROR IN READ GAP
6171          :*          ERROR 16      :CS1 IN ERROR AFTER READ GAP
6172          :*          ERROR 17      :MR1 IN ERROR IN WRITE SYNC

```


E10

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZ6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 121
 DZ6EA.P11 28-SEP-76 15:31 DIAGNOSTIC MODE GAP READ AND SYNC WRITE

SEQ 0121

```

6229
6230 027232 005724          TST      (R4)+          ;BUMP RETURN POINTER
6231
6232 027234 005037 002326    CLR      BITCNT        ;CLEAR BIT COUNTER
6233 027240 005037 002322    CLR      M1.BIT        ;CLEAR ALL WRITE BIT INDICATORS
6234 027244 005037 002324    CLR      M2.BIT
6235 027250 005037 002316    CLR      P1.BIT
6236 027254 005037 002320    CLR      PR.BIT
6237 027260 012737 062040 002244  MOV     #DMD!ECCW!MEWD!WRTGATE,E.MR1 ;SET EXPECTED MR1
6238 027266 012700 000400    MOV     #256.,R0        ;SET COUNT TO WRITE 256 "0"
6239
6240 027272 004737 032450    5$:  JSR      PC,WRTBIT      ;CALL WRITE BIT
6241 027276 000405          BR        6$              ;ERROR RETURN - SKIP TO HANDLER
6242 027300 005237 002326    INC      BITCNT        ;BUMP BIT COUNT
6243 027304 005300          DEC      R0              ;LOOP UNTIL ALL 256
6244 027306 001371          BNE      5$              ;BITS ARE WRITTEN
6245 027310 000402          BR        7$
6246
6247 027312 010446          6$:  MOV     R4,-(SP)        ;WRITE ERROR HANDLER-SET
6248 027314 000204          RTS      R4              ;STACK FOR REPORTING SUBSEQUENT ERRORS
6249
6250 027316 012737 000001 002316 7$:  MOV     #1,P1.BIT      ;SET SYNC BIT OF 1 AS NEXT BIT
6251 027324 004737 032450    JSR      PC,WRTBIT      ;GO WRITE IT
6252 027330 000770          BR        6$              ;ERROR RETURN
6253 027332 005237 002326    INC      BITCNT        ;BUMP BIT COUNT
6254
6255 027336 004437 034154    JSR      R4,GETREG      ;GET RK611 REGS
6256 027342 005724          TST      (R4)+          ;BUMP TO NEXT ERROR RETURN
6257 027344 023737 002160 002220  CMP     T.CS1,E.CS1    ;CHECK IF CORRECT
6258 027352 001274          BNE      2$              ;NO - SKIP
6259
6260 027354 005724          TST      (R4)+          ;BUMP PAST ERROR RETURNS
6261 027356 000711          BR        3$              ;GO TO EXIT
6262
6263 027360 032777 001000 151552 50$: BIT     #BIT9,2SWP      ;TEST IF LOOP ON ERROR
6264 027366 001403          BEQ      51$              ;YES - SKIP
6265 027370 005037 001236    CLR      $ESCAPE        ;ELSE SET TO LOOP
6266 027374 000702          BR        3$
6267 027376 010446          51$:  MOV     R4,-(SP)        ;PRESET STACK FOR RETURN
6268 027400 000204          RTS      R4
6269
6270          .SBTTL  DIAGNOSTIC WRITE DATA AND ECC ROUTINE
6271          :*      THE DATA IN THE OUPUT BUFFER IS WRITE SIMULATED, ECC IS
6272          :*      GENERATED AND WRITTEN, AND THE POSTAMBLE IS WRITTEN.
6273          :*
6274          :*      CALL:
6275          :*      JSR      R4,DWRITE
6276          :*      ERROR    17          ;MR1 IN ERROR IN WRITING
6277          :*      ERROR    21          ;ECC ERROR IN WRITING
6278          :*      ERROR    22          ;CS1 ERROR AFTER WRITE
6279          :*      NO ERROR RETURN
6280          :*
6281          :*      ROUTINE CALLED:
6282          :*      JSR      PC,WRTBIT
6283          :*      RETURN POINT IF ERROR IN WRTBIT
6284          :*      NO ERROR RETURN FROM WRTBIT

```

F10

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZ6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 122
 DZ6EA.P11 28-SEP-76 15:31 DIAGNOSTIC WRITE DATA AND ECC ROUTINE

SEQ 0122

```

6285 027402          DWRITE:
6286 027402 010046      MOV    R0,-(SP)      ;;PUSH R0 ON STACK
6287 027404 010146      MOV    R1,-(SP)      ;;PUSH R1 ON STACK
6288 027406 010346      MOV    R3,-(SP)      ;;PUSH R3 ON STACK
6289 027410 010546      MOV    R5,-(SP)      ;;PUSH R5 ON STACK
6290 027412 005037 002326 CLR    BITCNT      ;;CLEAR BIT COUNTER
6291 027416 005037 002342 CLR    ECCLO      ;;CLEAR ECC LOW
6292 027422 005037 002340 CLR    ECCHI      ;;CLEAR ECC HI
6293 027426 012700 000400 MOV    #400,R0      ;;SET WORD COUNT FOR ONE SECTOR
6294 027432 012703 051444 MOV    #0BUFF,R3      ;;GET POINTER TO OUTPUT DATA
6295 027436 010437 001216 MOV    R4,$TMP6      ;;STORE MRI ERROR RETURN
6296 027442 005724      TST    (R4)+      ;;BUMP TO NEXT ERROR RETURN
6297 027444 010437 001220 MOV    R4,$TMP7      ;;STORE ECC ERROR RETURN
6298
6299
6300
6301
6302
6303 027450 005037 001214      CLR    $TMP5      ;;CLEAR SWITCH
6304 027454 012701 000001 3$: MOV    #BIT0,R1      ;;LOAD BIT POINTER
6305 027460 012305      MOV    (R3)+,R5      ;;GET A WORD FOR WRITING
6306
6307 027462 013737 002322 002324 4$: MOV    M1.BIT,M2.BIT      ;;SHIFT CURRENT-1 TO CURRENT-2
6308 027470 013737 002320 002322      MOV    PR.BIT,M1.BIT      ;;PRESENT TO CURRENT-1
6309 027476 013737 002316 002320      MOV    P1.BIT,PR.BIT      ;;NEXT TO PRESENT
6310 027504 005037 002316      CLR    P1.BIT      ;;CLEAR NEXT BIT
6311 027510 030105      BIT    R1,R5      ;;TEST IF NEXT BIT IS ONE
6312 027512 001403      BEQ    $S      ;;NO - SKIP
6313 027514 012737 000001 002316      MOV    #1,P1.BIT      ;;ELSE SET NEXT BIT
6314 027522 004737 032450 5$: JSR    PC,WRTBIT      ;;GO WRITE BIT
6315 027526 000441      BR    33$      ;;ERROR RETURN - SKIP
6316 027530 004737 032260 43$: JSR    PC,ECCGEN      ;;GO GENERATE ECC FOR THIS BIT
6317 027534 016237 000032 002214      MOV    RKECPT(R2),T.ECPT      ;;GET PATTERN
6318 027542 023737 002254 002214      CMP    E.ECPT,T.ECPT      ;;CHECK IF ECC CORRECT
6319 027550 001402      BEQ    40$      ;;YES - SKIP
6320 027552 000137 030150      JMP    13$      ;;JUMP TO ERROR EXIT
6321
6322 027556 005237 002326 40$: INC    BITCNT      ;;BUMP BIT COUNT
6323 027562 005701      TST    R1      ;;TEST BIT POINTER
6324 027564 100424      BMI    7$      ;;IF NEGATIVE - SKIP
6325 027566 006301      ASL    R1      ;;ELSE SHIFT LEFT AND LOOP
6326 027570 020027 000001      CMP    R0,#1      ;;TEST IF WRITING LAST WORD
6327 027574 001332      BNE    4$      ;;NO - GO TO LOOP
6328 027576 032737 010000 002260      BIT    #CFMT,L.CS1      ;;TEST IF 22 SECTOR MODE
6329 027604 001404      BEQ    32$      ;;NO - SKIP
6330 027606 005737 001214      TST    $TMP5      ;;TEST IF 18 BIT SWITCH SET
6331 027612 001001      BNE    32$      ;;YES - SKIP
6332 027614 000722      BR    4$      ;;ELSE LOOP
6333 027616 005701 32$: TST    R1      ;;ELSE TEST IF LAST BIT OF LAST WORD
6334 027620 100320      BPL    4$      ;;NO - GO TO LOOP
6335 027622 042737 020000 002244      BIC    #ECCW,E.MR1      ;;ELSE SET MR1 FOR ECC WRITE
6336 027630 000714      BR    4$
6337 027632 000137 030172 33$: JMP    20$
6338
6339 027636 032737 010000 002260 7$: BIT    #CFMT,L.CS1      ;;TEST IF 22 SECTOR MODE
6340 027644 001413      BEQ    31$      ;;NO - SKIP

```

G10

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 123
 DZR6EA.P11 28-SEP-76 15:31 DIAGNOSTIC WRITE DATA AND ECC ROUTINE

SEQ 0123

6341	027646	005737	001214		TST	\$TMP5		; TEST IF 18 BIT SWITCH SET
6342	027652	001006			BNE	30\$		; YES - SKIP
6343	027654	012701	040000		MOV	#BIT14,R1		; SET TO SUPPLY 2 MORE BITS (16 & 17)
6344	027660	005005			CLR	R5		; MAKE SURE THEY ARE 0'S
6345	027662	005137	001214		COM	\$TMP5		; SET THE 18 BIT SWITCH
6346	027666	000675			BR	4\$		; GO WRITE THE BITS
6347								
6348	027670	005037	001214	30\$:	CLR	\$TMP5		; CLEAR THE SWITCH
6349	027674	005300		31\$:	DEC	R0		; DEC WORD COUNT
6350	027676	001266			BNE	3\$		; IF NOT ZERO - GO GET NEXT WORD
6351	027700	005037	001214		CLR	\$TMP5		; CLEAR FOR PASS CONTROL
6352	027704	013705	002340		MOV	ECCHI,R5		; GET FIRST ECC WORD
6353	027710	012701	000001		MOV	#BIT0,R1		; SET BIT POINTER
6354								
6355	027714	013737	002322	002324	8\$:	MOV	M1.BIT,M2.BIT	; SHIFT BIT INDICATORS AGAIN
6356	027722	013737	002320	002322		MOV	PR.BIT,M1.BIT	
6357	027730	013737	002316	002320		MOV	P1.BIT,PR.BIT	
6358	027736	005037	002316		CLR	P1.BIT		; CLEAR NEXT BIT
6359	027742	030105			BIT	R1,R5		; TEST IF BIT IS ZERO
6360	027744	001403			BEQ	9\$		; YES - SKIP
6361	027746	012737	000001	002316	MOV	#1,P1.BIT		; ELSE CHANGE TO ONE
6362	027754	004737	032450	9\$:	JSR	PC,WRTBIT		; GO WRITE BIT
6363	027760	000501			BR	14\$		; WRTBIT ERROR - SKIP
6364								
6365	027762	005237	002326	41\$:	INC	BITCNT		; BUMP BIT COUNT
6366	027766	005701			TST	R1		; TEST BIT POINTER
6367	027770	100413			BMI	10\$		; IF NEGATIVE - SKIP
6368	027772	006301			ASL	R1		; ELSE SHIFT AND LOOP
6369	027774	023727	001214	000001	CMP	\$TMP5,#1		; TEST IF WRITING LAST ECC WORD
6370	030002	001344			BNE	8\$		; NO - SKIP TO LOOP
6371	030004	005701			TST	R1		; ELSE TEST IF WRITING LAST BIT OF LAST ECC
6372	030006	100342			BPL	8\$		; NO - SKIP TO LOOP
6373	030010	052737	020000	002244	BIS	#ECCW,E.MR1		; ELSE SET ECCW IN MR1
6374	030016	000736			BR	8\$		
6375								
6376	030020	005737	001214	10\$:	TST	\$TMP5		; TEST PASS 0 JUST DONE
6377	030024	001007			BNE	11\$		; NO - SKIP
6378	030026	005237	001214		INC	\$TMP5		; BUMP PASS COUNT
6379	030032	012701	000001		MOV	#BIT0,R1		; PRESET BIT POINTER
6380	030036	013705	002342		MOV	ECCL0,R5		; GET SECOND ECC WORD
6381	030042	000724			BR	8\$		; LOOP
6382								
6383	030044	012701	000017	11\$:	MOV	#15.,R1		; SET A BIT COUNT
6384								
6385	030050	013737	002322	002324	12\$:	MOV	M1.BIT,M2.BIT	; SHIFT BIT INDICATORS
6386	030056	013737	002320	002322		MOV	PR.BIT,M1.BIT	
6387	030064	013737	002316	002320		MOV	P1.BIT,PR.BIT	
6388	030072	005037	002316		CLR	P1.BIT		; SET ZERO FOR WRITE POSTAMBLE
6389	030076	004737	032450		JSR	PC,WRTBIT		; GO WRITE BIT
6390	030102	000436			BR	15\$		; WRITE BIT ERROR - SKIP
6391	030104	005237	002326	42\$:	INC	BITCNT		; BUMP BIT COUNT
6392	030110	005301			DEC	R1		; DEC BIT COUNT
6393	030112	001356			BNE	12\$		; LOOP UNTIL BIT COUNT 0
6394								
6395	030114	004437	034154		JSR	R4,GETREG		; GET RK611 REGS
6396	030120	013704	001220		MOV	\$TMP7,R4		; SET R4 FOR ECC ERROR RETURN

H10

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 124
 DZR6EA.P11 28-SEP-76 15:31 DIAGNOSTIC WRITE DATA AND ECC ROUTINE

SEQ 0124

```

6397 030124 005724          TST      (R4)+      ;BUMP TO CS1 ERROR RETURN
6398 030126 013737 002260 002220      MOV      L.CS1,E.CS1      ;GET EXPECTED CS1
6399
6400 030134 023737 002160 002220      CMP      T.CS1,E.CS1      ;CHECK IF CORRECT
6401 030142 001031          BNE      18$          ;NO - SKIP
6402 030144 005724          TST      (R4)+      ;BUMP TO GOOD RETURN
6403 030146 000446          BR       19$          ;GO TO RETURN
6404
6405 030150 013704 001220          13$:      MOV      $TMP7,R4      ;SET FOR ECC ERROR RETURN
6406 030154 012737 027556 001236      MOV      #40$,$ESCAPE      ;STORE RETURN POINT FOR NO LOOP ON ERROR
6407 030162 000410          BR       16$
6408
6409 030164 013704 001216          14$:      MOV      $TMP6,R4      ;SET FOR MR1 ERROR RETURN
6410 030170 000405          BR       16$
6411
6412 030172 013704 001216          20$:      MOV      $TMP6,R4      ;SET FOR MR1 ERROR RETURN
6413 030176 000402          BR       16$
6414
6415 030200 013704 001216          15$:      MOV      $TMP6,R4      ;SET FOR MR1 ERROR RETURN
6416
6417 030204 032777 001000 150726 16$:      BIT      #BIT9,$SWR      ;CHECK IF LOOP ON ERROR
6418 030212 001403          BEQ      17$          ;NO - SKIP
6419
6420 030214 005037 001236          CLR      $ESCAPE      ;CLEAR ESCAPE
6421 030220 000421          BR       19$          ;SKIP TO EXIT
6422
6423 030222 010446          17$:      MOV      R4,-(SP)      ;CONDITION STACK
6424 030224 000204          RTS      R4          ;GO REPORT ERROR
6425
6426 030226 005037 001236          18$:      CLR      $ESCAPE      ;CLEAR FOR LOOP OR EXIT
6427 030232 032777 001000 150700      BIT      #BIT9,$SWR      ;CHECK IF LOOP ON ERROR
6428 030240 001011          BNE      19$          ;YES - SKIP
6429 030242 013700 001254          MOV      $TESTN,R0      ;SET UP TO SKIP TO NEXT TEST
6430 030246 006300          ASL      R0
6431 030250 016037 035270 001236      MOV      $SW08TB(R0),$ESCAPE
6432 030256 162737 000002 001236      SUB      #2,$ESCAPE
6433 030264          19$:
6434 030264 012605          MOV      (SP)+,R5      ;;POP STACK INTO R5
6435 030266 012603          MOV      (SP)+,R3      ;;POP STACK INTO R3
6436 030270 012601          MOV      (SP)+,R1      ;;POP STACK INTO R1
6437 030272 012600          MOV      (SP)+,R0      ;;POP STACK INTO R0
6438 030274 000204          RTS      R4
6439
6440          .SBTTL      DIAGNOSTIC MODE READ GAP AND DATA SYNC
6441          ;*      THE READING OF THE GAP AND READING OF THE SYNC (OR PREAMBLE)
6442          ;*      IS HANDLED IN THIS ROUTINE.
6443          ;*
6444          ;*      THE FIRST GAP BITS ARE PROCESSED IN THE HEADER COMPARE
6445          ;*      ROUTINE AND THE REMAINING GAP BITS ARE PROCESSED HERE.
6446          ;*      THEN A CHECK IS MADE THAT READ GATE CAME ON
6447          ;*      AT THE 129TH BIT OF SYNC.*
6448          ;*      CALL:
6449          ;*      JSR      R4,DRGPSN
6450          ;*      ERROR    15      ;MR1 IN ERROR IN READ GAP
6451          ;*      ERROR    16      ;CS1 IN ERROR AFTER READ GAP
6452          ;*      ERROR    32      ;MR1 IN ERROR IN READ SYNC

```

I10

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 125
 DZR6EA.P11 28-SEP-76 15:31 DIAGNOSTIC MODE READ GAP AND DATA SYNC

SEQ 0125

```

6453      ;*      ERROR 33      ;CS1 IN ERROR AFTER SYNC READ
6454      ;*
6455      ;*      ROUTINE CALLED:
6456      ;*      JSR      PC,RDBIT
6457
6458      030276      DRGPSN:
6459      030276      010046      MOV      RD,-(SP)      ;;PUSH RD ON STACK
6460      ;      THE FOLLOWING CODE ADJUST THE COUNT FOR THE REMAINDER OF
6461      ;      THE GAP. THE ADJUSTMENT DEPENDS ON HOW MANY GAP BITS WERE
6462      ;      READ IN THE HEADER COMPARE ROUTINE FOR ADDRESS REGISTER UPDATE.
6463      030300      005737      002266      TST      L.DA      ;TEST IF DESIRED TRACK/SECTOR NOW 0
6464      030304      001006      BNE      8$      ;NO - SKIP
6465      030306      012700      000066      MOV      #54.,RO      ;ELSE ADJUST COUNT FOR CYL CHANGE
6466      030312      012737      000011      002326      MOV      #11,BITCNT      ;PRESET BIT COUNT FOR REST OF GAP
6467      030320      000416      BR      10$
6468      030322      105737      002266      8$:      TSTB     L.DS      ;TEST IF NEW SECTOR IS 0
6469      030326      001006      BNE      9$      ;NO - SKIP
6470      030330      012700      000070      MOV      #56.,RO      ;ELSE ADJUST COUNT FOR TRACK CHANGE
6471      030334      012737      000007      002326      MOV      #7,BITCNT      ;PRESET BIT COUNT FOR REST OF GAP
6472      030342      000405      BR      10$
6473      030344      012737      000005      002326      9$:      MOV      #5,BITCNT      ;PRESET BIT COUNTER FOR REST OF GAP
6474      030352      012700      000074      MOV      #60.,RO      ;SET COUNT FOR 60 BITS
6475      ;
6476      ;      END OF BLOCK
6477      030356      012737      022040      002244      10$:      MOV      #DMD!ECCW!MEWD,E.MR1 ;SET EXPECTED MR1
6478      030364      005037      002320      CLR      PR.BIT      ;CLEAR PRESENT BIT
6479      030370      005037      002322      CLR      M1.BIT      ;CLEAR PREVIOUS BIT
6480      030374      004737      033414      1$:      JSR      PC,RDBIT      ;GO READ BIT
6481      030400      016237      000026      002204      MOV      RKMR1(R2),T.MR1 ;GET MR1
6482      030406      023737      002244      002204      CMP      E.MR1,T.MR1 ;CHECK IF CORRECT
6483      030414      001404      BEQ      11$      ;YES - SKIP
6484      030416      012737      030440      001236      MOV      #2,$ESCAPE ;SET RETURN TO CONT TEST
6485      030424      000552      BR      50$
6486      030426      005237      002326      11$:      INC      BITCNT      ;BUMP BIT COUNT
6487      030432      005300      DEC      RO      ;DEC LOOP COUNT
6488      030434      001357      BNE      1$      ;LOOP IF NOT ZERO
6489      030436      000421      BR      4$      ;ELSE SKIP
6490
6491      030440      005037      001236      2$:      CLR      $ESCAPE ;CLEAR TO EXIT OR LOOP
6492      030444      022777      001000      150466      CMP      #BIT9,$SWR ;ERROR RETURN: CHECK IF LOOP ON ERROR
6493      030452      001011      BNE      3$      ;YES - SKIP
6494      030454      013700      001254      MOV      $TESTN,RO ;GET NUMBER OF NEXT TEST
6495      030460      006300      ASL      RO      ;SHIFT FOR INDEXING
6496      030462      016037      035270      001236      MOV      $SWOBTB(RO),$ESCAPE ;SET ESCAPE FOR GOT TO NEXT TEST
6497      030470      162737      000002      001236      SUB      #2,$ESCAPE ;SET TO FIRST INST (SCOPE)
6498      030476      3$:
6499      030476      012600      MOV      (SP)+,RO      ;;POP STACK INTO RO
6500      030500      000204      RTS      R4
6501
6502      030502      004437      034154      4$:      JSR      R4,GETREG ;GET RK611 REGS
6503      030506      005724      TST      (R4)+ ;BUMP TO NEXT ERROR RETURN
6504      030510      013737      002260      002220      MOV      L.CS1,E.CS1 ;SET EXPECTED CS1
6505      030516      023737      002160      002220      CMP      T.CS1,E.CS1 ;CHECK IF CS1 OK
6506      030524      001345      BNE      2$      ;NO - SKIP
6507
6508      030526      005724      TST      (R4)+ ;BUMP RETURN POINTER

```

J10

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 126
 DZR6EA.P11 28-SEP-76 15:31 DIAGNOSTIC MODE READ GAP AND DATA SYNC

SEQ 0126

```

6509
6510 030530 005037 002326          CLR      BITCNT          ;CLEAR BIT COUNTER
6511 030534 012737 022040 002244    MOV      #DMD!ECCW!MEWD,E.MR1 ;SET EXPECTED MR1
6512 030542 012700 000200          MOV      #128.,RO          ;SET COUNT TO READ 128 "0"
6513
6514 030546 004737 033414          5$:    JSR      PC,RDBIT          ;CALL READ BIT
6515 030552 016237 000026 002204    MOV      RKMRI(R2),T.MR1      ;GET MR1 FOR TESTING
6516 030550 023737 002244 002204    CMP      E.MR1,T.MR1          ;TEST IF CORRECT
6517 030566 001404          BEQ      22$          ;YES - SKIP
6518
6519 030570 012737 030600 001236    MOV      #22$, $ESCAPE          ;SET RETURN FOR CONTINUE TEST
6520 030576 000465          BR      50$          ;GO TO ERROR EXIT
6521 030600 005237 002326          22$:    INC      BITCNT          ;BUMP BIT COUNT
6522 030604 005300          DEC      RO          ;LOOP UNTIL ALL 128
6523 030606 001357          BNE      5$          ;BITS ARE READ
6524 030610 012700 000177          MOV      #127.,RO          ;SET COUNT FOR REST OF SYNC
6525 030614 052737 100000 002244    BIS      #RDGATE,E.MR1          ;SET EXPECTED MR1
6526 030622 004737 033414          6$:    JSR      PC,RDBIT          ;GO CALL READ BIT
6527 030626 016237 000026 002204    MOV      RKMRI(R2),T.MR1      ;GET MR1 FOR TESTING
6528 030634 023737 002244 002204    CMP      E.MR1,T.MR1          ;TEST IF CORRECT
6529 030642 001404          BEQ      23$          ;YES - SKIP
6530 030644 012737 030654 001236    MOV      #23$, $ESCAPE          ;ELSE SET ESCAPE
6531 030652 000437          BR      50$          ;GO TO ERROR REPORT EXIT
6532
6533 030654 005237 002326          23$:    INC      BITCNT          ;BUMP BIT COUNTER
6534 030660 005300          DEC      RO          ;DEC BIT COUNT
6535 030662 001357          BNE      6$          ;LOOP UNTIL ALL 127 READ
6536
6537 030664 012737 000001 002320    MOV      #1,PR.BIT          ;SET SYNC BIT OF 1 AS NEXT BIT
6538 030672 004737 033414          JSR      PC,RDBIT          ;GO READ IT
6539 030676 016237 000026 002244    MOV      RKMRI(R2),E.MR1      ;GET MR1 FOR TESTING
6540 030704 023737 002244 002204    CMP      E.MR1,T.MR1          ;CHECK IF CORRECT
6541 030712 001404          BEQ      24$          ;YES - SKIP
6542 030714 012737 030724 001236    MOV      #24$, $ESCAPE          ;ELSE SET RETURN FOR CONTINUE TEST
6543 030722 000413          BR      50$
6544
6545 030724 005237 002326          24$:    INC      BITCNT          ;BUMP BIT COUNT
6546
6547 030730 004437 034154          JSR      R4,GETREG          ;GET RK611 REGS
6548 030734 005724          TST      (R4)+          ;BUMP TO NEXT ERROR RETURN
6549 030736 023737 002160 002220    CMP      T.CS1,E.CS1          ;CHECK IF CORRECT
6550 030744 001235          BNE      2$          ;NO - SKIP
6551
6552 030746 005724          TST      (R4)+          ;BUMP PAST ERROR RETURNS
6553 030750 000652          BR      3$          ;GO TO EXIT
6554
6555 030752 032777 001000 150160 50$:    BIT      #BIT9,$SWR          ;TEST IF LOOP ON ERROR
6556 030760 001403          BEQ      51$          ;YES - SKIP
6557 030762 005037 001236          CLR      $ESCAPE          ;ELSE SET TO LOOP
6558 030766 000643          BR      3$
6559 030770 010446          51$:    MOV      R4,-(SP)          ;PRESET STACK FOR RETURN
6560 030772 000204          RTS      R4
6561
6562      .SBTTL      DIAGNOSTIC READ DATA OR DIAGNOSTIC WRITE CHECK ROUTINE
6563      ;*          THIS ROUTINE IS USED TO SIMULATE THE READ OR WRITE CHECK
6564      ;*          OF DATA. THE ROUTINE WILL TRANSFER A FULL OR PARTIAL SECTOR.
        ;*          THE FIRST WORD AFTER THE CALL SPECIFIES HOW MANY WORDS

```

K10

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 127
 DZR6EA.P11 28-SEP-76 15:31 DIAGNOSTIC READ DATA OR DIAGNOSTIC WRITE CHECK ROUTINE

SEQ 0127

```

6565      ARE TO BE TRANSFERRED.
6566      :
6567      :
6568      : IF LESS THAN A FULL SECTOR, THE ECC AT THE END
6569      : OF THE TRANSFER IS NOT CHECKED. THE ECC IS CHECKED AT THE TIME
6570      : EACH BIT IS TRANSFERRED.
6571      :
6572      : EITHER 16 OR 18 BIT WORDS CAN BE SIMULATED
6573      : DEPENDING ON THE STATE OF CFMT BIT IN L.CS1. IF 18 BITS
6574      : ARE REQUIRED, BITS 16 AND 17 ARE ASSUMED TO BE ZERO.
6575      :
6576      : THE CONTENTS OF LOCATION ECPOSX MUST BE LOADED WITH THE ECC
6577      : COUNT EXPECTED AT THE END OF THE DATA TRANSFER. THIS
6578      : LOADING MUST OCCUR BEFORE THE ROUTINE IS CALLED.
6579      :
6580      : ANOTHER LOCATION, ECPATX, IS SET TO 0 IF THE ECC IS EXPECTED
6581      : TO BE CORRECT AND SET TO 1 IF AN ERROR IS BEING FORCED.
6582      :
6583      : A 3RD LOCATION, ECCSRC, SPECIFIES THE SOURCE OF THE ECC
6584      : WORDS TO BE USED IN CHECKING THE OPERATION. IF ECCSRC IS A
6585      : 1, THE ECC WORDS FED INTO THE DECODER ARE TAKEN FROM THE
6586      : LAST TWO LOCATIONS OF OBUFF. IF ECCSRC IS 0, THE COMPUTED
6587      : ECC WORDS FROM ECCHI AND ECCLO ARE USED.
6588      : CALL:
6589      : JSR      R4,DREAD
6590      : LENGTH OF TRANSFER
6591      : ERROR 34      ;MR1 IN ERROR READING DATA OR ECC
6592      : ERROR 35      ;ECC ERROR READING DATA
6593      : ERROR 36      ;CS1 ERROR AFTER READING DATA OR ECC
6594      : ERROR 41      ;ECC REGISTER INCORRECT AFTER READ ECC
6595      : ERROR 42      ;ERR IN ECC PAT CALCULATION
6596      : ERROR 43      ;ERR IN ECC POS COUNTING
6597      :
6598      : OR
6599      : JSR      R4,DWRTCK
6600      : LENGTH OF TRANSFER
6601      : ERROR 53      ;MR1 IN ERROR WRITE CHK OR ECC READ
6602      : ERROR 54      ;ECC ERROR IN WRITE CHECK
6603      : ERROR 55      ;CS1 ERROR AFTER IN WRT CHK OR ECC READ
6604      : ERROR 41      ;ECC REGISTER INCORRECT AFTER READ ECC
6605      : ERROR 42      ;ERR IN ECC PAT CALCULATION
6606      : ERROR 43      ;ERR IN ECC POS COUNTING
6607      :
6608      : ROUTINES CALLED:
6609      : RDBIT
6610      : ECCGEN
6611      :
6612      :
6613      : DWRTCK:
6614      : DREAD:
6615      : MOV      R0,-(SP)      ;;PUSH R0 ON STACK
6616      : MOV      R1,-(SP)      ;;PUSH R1 ON STACK
6617      : MOV      R3,-(SP)      ;;PUSH R3 ON STACK
6618      : MOV      R5,-(SP)      ;;PUSH R5 ON STACK
6619      : MOV      (R4)+,R0      ;STORE WORD COUNT
6620      : MOV      R4,$TMP6      ;STORE MR1 ERROR RETURN
  
```

030774
 030774
 030774 010046
 030776 010146
 031000 010346
 031002 010546
 031004 012400
 031006 010437 001216

L10

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 128
 DZR6EA.P11 28-SEP-76 15:31 DIAGNOSTIC READ DATA OR DIAGNOSTIC WRITE CHECK ROUTINE

SEG 0128

6621	031012	005724			TST	(R4)+	;BUMP TO NEXT RETURN
6622	031014	010437	001220		MOV	R4,\$TMP7	;STORE ECC ERROR RETURN
6623	031020	005724			TST	(R4)+	;BUMP TO NEXT ERROR RETURN
6624	031022	010437	001210		MOV	R4,\$TMP3	;STORE CS1 ERROR RETURN
6625	031026	005724			TST	(R4)+	;BUMP TO NEXT ERROR RETURN
6626	031030	010437	001212		MOV	R4,\$TMP4	;STORE ECC REG ERROR RETURN
6627	031034	005724			TST	(R4)+	;BUMP TO NEXT ERROR RETURN
6628	031036	010437	001222		MOV	R4,\$TMP10	;STORE ECC PAT ERR RETURN
6629	031042	005724			TST	(R4)+	;BUMP TO NEXT ERROR RETURN
6630	031044	010437	001224		MOV	R4,\$TMP11	;SOTER ECC POS ERROR RETURN
6631	031050	012703	051444		MOV	#0BUFF,R3	;GET ADDRESS OF BLUFFER
6632	031054	005037	002326		CLR	BITCNT	;CLEAR BIT COUNTER
6633	031060	005037	002340		CLR	ECCHI	;CLEAR ECC COMPUTED LOCATIONS
6634	031064	005037	002342		CLR	ECCL0	
6635	031070	005037	001214		CLR	\$TMP5	;CLEAR FOR 18 BIT MODE SWITCH
6636	031074	032737	010000	002260	BIT	#CFMT,L.CS1	;TEST IF 18 BIT MODE
6637	031102	001412			BEQ	32\$	;NO - SKIP
6638	031104	012737	011000	001226	MOV	#11000,\$TMP12	;ELSE SET UP BIT COUNTS FOR 18 BIT MODE
6639	031112	012737	011001	001230	MOV	#11001,\$TMP13	
6640	031120	012737	011041	001232	MOV	#11041,\$TMP14	
6641	031126	000411			BR	3\$	
6642	031130	012737	010000	001226	32\$: MOV	#10000,\$TMP12	;SET BIT COUNTS FOR 16 BIT MODE
6643	031136	012737	010001	001230	MOV	#10001,\$TMP13	
6644	031144	012737	010041	001232	MOV	#10041,\$TMP14	
6645							
6646	031152	012701	000001		3\$: MOV	#BIT0,R1	;PRESET BIT POINTER
6647	031156	012305			MOV	(R3)+,R5	;GET DATA WORD
6648							
6649	031160	013737	002320	002322	4\$: MOV	PR.BIT,M1.BIT	;SHIFT PRESENT INTO PREVIOUS
6650	031166	005037	002320		CLR	PR.BIT	;CLEAR PRESENT
6651	031172	030105			BIT	R1,R5	;TEST IF PRESENT BIT IS 0
6652	031174	001403			BEQ	1\$	;YES - SKIP
6653	031176	012737	000001	002320	MOV	#1,PR.BIT	;ELSE INSERT A 1
6654							
6655	031204	004737	033414		1\$: JSR	PC,RDBIT	;GO READ BIT
6656	031210	016237	000026	002204	MOV	RKMRI(R2),T.MRI	;GET MRI
6657	031216	023737	002204	002244	CMP	T.MRI,E.MRI	;CHECK IF CORRECT
6658	031224	001402			BEQ	43\$	;YES - SKIP
6659	031226	000137	032060		JMP	9\$	
6660							
6661	031232	005737	002326		43\$: TST	BITCNT	;TEST IF FIRST BIT OF DATA
6662	031236	001413			BEQ	44\$	;YES - SKIP ECC GEN
6663	031240	004737	032260		JSR	PC,ECCGEN	;GO GENERATE ECC
6664	031244	016237	000032	002214	MOV	RKECPT(R2),T.ECPT	;TEST IF ECC CORRECT
6665	031252	023737	002214	002254	CMP	T.ECPT,E.ECPT	
6666	031260	001402			BEQ	44\$	
6667	031262	000137	032074		JMP	10\$	
6668							
6669	031266	005237	002326		44\$: INC	BITCNT	;BUMP BIT COUNTER
6670	031272	005701			TST	R1	;TEST IF THIS WORD IS DONE
6671	031274	100402			BMI	2\$	;YES - SKIP
6672	031276	006301			ASL	R1	;ELSE SHIFT BIT POINTER
6673	031300	000727			BR	4\$	
6674							
6675	031302	032737	010000	002260	2\$: BIT	#CFMT,L.CS1	;TEST IF 18 BIT MODE
6676	031310	001412			BEQ	30\$	;NO - SKIP

M10

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 129
 DZR6EA.P11 28-SEP-76 15:31 DIAGNOSTIC READ DATA OR DIAGNOSTIC WRITE CHECK ROUTINE

SEQ 0129

6677	031312	005737	001214		TST	\$TMP5	;TEST IF PROCESSING BIT 16 OR 17 DONE
6678	031316	001007			BNE	30\$	;YES - SKIP
6679	031320	012701	040000		MOV	#BIT14,R1	;SET POINTER FOR 2 MORE BITS
6680	031324	013705	002354		MOV	HIBITS,R5	;INSERT HI ORDER BITS
6681	031330	005137	001214		COM	\$TMP5	;SET 18 BIT SWITCH
6682	031334	000711			BR	4\$	
6683	031336	005037	001214	30\$:	CLR	\$TMP5	;CLEAR 18 BIT SWITCH
6684							
6685	031342	005300		31\$:	DEC	R0	;DECREMENT WORD COUNT
6686	031344	001302			BNE	3\$	;IF NOT ZERO - LOOP
6687							
6688	031346	023737	002326	001226	CMP	BITCNT,\$TMP12	;ELSE TEST IF FULL SECTOR WAS READ
6689	031354	001402			BEQ	33\$	;YES - SKIP
6690	031356	000137	032032		JMP	8\$	;ELSE JUMP TO EXIT - DO NOT CHECK ;ECC WORDS
6691							
6692							
6693	031362	013737	002320	002322	33\$:	MOV	PR.BIT,M1.BIT ;MOVE LAST DATA BIT FOR ECC GEN
6694	031370	004737	032260		JSR	PC,ECCGEN	;GENERATE ECC
6695	031374	005037	001214		CLR	\$TMP5	;CLEAR FOR PASS COUNTING
6696	031400	005737	002344		TST	ECCSRC	;TEST WHERE ECC WORDS ARE
6697	031404	001006			BNE	36\$	;THEY ARE IN BUFFER - SKIP
6698	031406	013737	002340	052444	MOV	ECCHI,0BUFF+1000	;ELSE MOVE THEM INTO BUFFER
6699	031414	013737	002342	052446	MOV	ECCL0,0BUFF+1002	
6700							
6701	031422	012305		36\$:	MOV	(R3)+,R5	;GET ECC FROM BUFFER
6702	031424	012701	000001		MOV	#BIT0,R1	;SET BIT POINTER
6703							
6704	031430	013737	002320	002322	5\$:	MOV	PR.BIT,M1.BIT ;SHIFT PRESENT TO PREVIOUS
6705	031436	005037	002320		CLR	PR.BIT	;AND SET UP PRESENT
6706	031442	030105			BIT	R1,R5	
6707	031444	001403			BEQ	34\$	
6708	031446	012737	000001	002320	MOV	#1,PR.BIT	
6709	031454	004737	033414	34\$:	JSR	PC,RDBIT	;GO READ BIT
6710	031460	016237	000026	002204	MOV	RKMR1(R2),T.MR1	;GET MR1
6711	031466	023737	002204	002244	CMP	T.MR1,E.MR1	;CHECK IF CORRECT
6712	031474	001402			BEQ	38\$	;YES - SKIP
6713	031476	000137	032110		JMP	11\$	;ELSE JUMP TO REPORT EXIT
6714	031502	023737	002326	001226	38\$:	CMP	BITCNT,\$TMP12 ;TEST IF FIRST ECC BIT
6715	031510	001402			BEQ	49\$	;YES - SKIP ECC GEN (ALREADY DONE)
6716	031512	004737	032260		JSR	PC,ECCGEN	;GENERATE ECC FOR BIT
6717	031516	016237	000032	002214	49\$:	MOV	RKECPT(R2),T.ECPT ;GET PATTERN
6718	031524	023737	002214	002254	CMP	T.ECPT,E.ECPT	;TEST IF ECC PAT CORRECT
6719	031532	001402			BEQ	45\$	;YES SKIP
6720	031534	000137	032074		JMP	10\$	;ELSE GO REPORT ERROR
6721							
6722	031540	005237	002326	45\$:	INC	BITCNT	;BUMP BIT COUNT
6723	031544	023737	002326	001230	CMP	BITCNT,\$TMP13	;TEST IF 1ST BIT OF ECC
6724	031552	001003			BNE	46\$	;NO - SKIP
6725	031554	042737	020000	002244	BIC	#ECCW,E.MR1	;ELSE CLR ECCW FOR ECC READ
6726	031562	023737	002326	001232	46\$:	CMP	BITCNT,\$TMP14 ;TEST IF 1ST BIT OF POSTAMBLE
6727	031570	001027			BNE	47\$	;NO - SKIP
6728	031572	052737	020000	002244	BIS	#ECCW,E.MR1	;ELSE SET ECCW FOR ECC DONE
6729	031600	042737	100000	002244	BIC	#RDGATE,E.MR1	;CLEAR READ GATE
6730							
6731	031606	016237	000032	002214	MOV	RKECPT(R2),T.ECPT	;STORE ACTUAL PATTERN
6732	031614	016237	000030	002212	MOV	RKECPS(R2),T.ECPS	;STORE ACTUAL POSITION

N10

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 130
DZR6EA.P11 28-SEP-76 15:31 DIAGNOSTIC READ DATA OR DIAGNOSTIC WRITE CHECK ROUTINE

SEQ 0130

6733	031622	013737	002350	002252		MOV	ECPOSX,E.ECPS	:GET EXPECTED POSITION FOR REPORT
6734	031630	023737	002214	002254		CMP	T.ECPT,E.ECPT	:CHECK IF CORRECT
6735	031636	001163				BNE	20\$	:NO SKIP
6736	031640	023737	002212	002252		CMP	T.ECPS,E.ECPS	:CHECK IF POSITION CORRECT
6737	031646	001157				BNE	20\$	:NO - SKIP
6738								
6739	031650	023737	002326	001232	47\$:	CMP	BITCNT,\$TMP14	:TEST IF ECC DONE
6740	031656	002436				BLT	48\$	:NO - SKIP
6741	031660	005737	002346			TST	ECPATX	:TEST IF EXPECTED ECC IS 0
6742								: (NO ERROR)
6743	031664	001433				BEQ	48\$	:YES - SKIP
6744	031666	023737	001232	002326		CMP	\$TMP14,BITCNT	:TEST IF FIRST POSTAMBLE BIT
6745	031674	001402				BEQ	52\$	:YES - SKIP
6746	031676	005237	002350			INC	ECPOSX	:ELSE BUMP POS COUNT EXPECTED
6747	031702	016237	000032	002214	52\$:	MOV	RKECPT(R2),T.ECPT	:GET PATTERN
6748	031710	016237	000030	002212		MOV	RKECPS(R2),T.ECPS	:GET POSITION
6749	031716	013737	002350	002252		MOV	ECPOSX,E.ECPS	:GET EXPECTED POSITION
6750	031724	023737	002214	002254		CMP	T.ECPT,E.ECPT	:TEST IF PATTERN CORRECT
6751	031732	001402				BEQ	51\$	:YES - SKIP
6752	031734	000137	032124			JMP	22\$	:ELSE GO REPORT
6753	031740	023737	002212	002252	51\$:	CMP	T.ECPS,E.ECPS	:CHECK IF POSITION CORRECT
6754	031746	001402				BEQ	48\$	:YES - SKIP
6755	031750	000137	032140			JMP	23\$	:ELSE GO REPORT
6756								
6757	031754	005701			48\$:	TST	R1	:TEST IF WORD DONE
6758	031756	100403				BMI	6\$	:YES SKIP
6759	031760	006301				ASL	R1	:ELSE SHIFT BIT POINTER
6760	031762	000137	031430			JMP	5\$	:AND LOOP
6761								
6762	031766	005737	001214		6\$:	TST	\$TMP5	:TEST PASS COUNT. IF NOT 0
6763								: (NOT THE FIRST ECC WORD JUST WRITTE)
6764	031772	001004				BNE	7\$	: - SKIP
6765								
6766	031774	005237	001214			INC	\$TMP5	:BUMP PASS COUNT
6767	032000	000137	031422			JMP	36\$	
6768								
6769	032004	023727	001214	000001	7\$:	CMP	\$TMP5,#1	:TEST PASS COUNT. IF NOT 1 (NOT THE
6770	032012	001007				BNE	8\$	:2ND ECC WORD JUST READ) - SKIP
6771	032014	005005				CLR	R5	:CLEAR WORD (POSTAMBLE)
6772	032016	012701	000001			MOV	#BIT0,R1	:SET BIT POINTER
6773	032022	005237	001214			INC	\$TMP5	:BUMP PASS COUNT
6774	032026	000137	031430			JMP	5\$	
6775								
6776	032032	004437	034154		8\$:	JSR	R4,GETREG	:GET RK611 REGS
6777	032036	013737	002260	002220		MOV	L.CS1,E.CS1	:SET EXPECTED CS1
6778	032044	023737	002220	002160		CMP	E.CS1,T.CS1	:CHECK IF CORRECT
6779	032052	001060				BNE	21\$	:NO - SKIP
6780	032054	005724				TST	(R4)+	:ELSE BUMP RETURN POINTER TO NO ERR
6781	032056	000446				BR	13\$	
6782								
6783	032060	013704	001216		9\$:	MOV	\$TMP6,R4	:SET RETURN TOR MR1 ERR.
6784	032064	012737	031232	001236		MOV	#43\$,\$ESCAPE	:SET RETURN TO CONTINUE TEST
6785	032072	000427				BR	12\$	
6786								
6787	032074	013704	001220		10\$:	MOV	\$TMP7,R4	:SET RETURN FOR ECC ERROR
6788	032100	012737	031266	001236		MOV	#44\$,\$ESCAPE	:SET RETURN TO CONTINUE TEST

B11

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZ6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 131
 DZ6EA.P11 28-SEP-76 15:31 DIAGNOSTIC READ DATA OR DIAGNOSTIC WRITE CHECK ROUTINE

SEG 0131

```

6789 032106 000421 BR 12$
6790
6791 032110 013704 001216 11$: MOV STMP6,R4 ;SET RETURN FOR MR1 ERROR
6792 032114 012737 031540 001236 MOV #45$, $ESCAPE ;SET RETURN TO CONTINUE TEST
6793 032122 000413 BR 12$
6794
6795 032124 013704 001222 22$: MOV STMP10,R4 ;GET RETURN FOR PATTERN ERROR
6796 032130 012737 031740 001236 MOV #51$, $ESCAPE ;SET RETURN TO CONTINUE TEST
6797 032136 000405 BR 12$
6798
6799 032140 013704 001224 23$: MOV STMP11,R4 ;RET RETURN FOR POSITION ERROR
6800 032144 012737 031754 001236 MOV #48$, $ESCAPE ;SET RETURN TO CONTINUE TEST
6801
6802
6803 032152 013746 001224 12$: MOV STMP11, -(SP) ;PREP STACK FOR RETURN
6804 032156 032777 001000 146754 BIT #BIT9, $SWR ;TEST IF LOOP ON ERROR
6805 032164 001407 BEQ 14$ ;NO SKIP
6806
6807 032166 005726 TST (SP)+ ;REMOVE LAST ENTRY FROM STACK
6808 032170 005037 001236 CLR $ESCAPE ;NO LOOP - SLEAR ESCAPE
6809
6810 032174 13$:
6811 032174 012605 MOV (SP)+, R5 ;:POP STACK INTO R5
6812 032176 012603 MOV (SP)+, R3 ;:POP STACK INTO R3
6813 032200 012601 MOV (SP)+, R1 ;:POP STACK INTO R1
6814 032202 012600 MOV (SP)+, R0 ;:POP STACK INTO R0
6815 032204 000204 14$: RTS R4
6816
6817 032206 013704 001212 20$: MOV STMP4,R4 ;GET RETURN FOR ECC ERROR
6818 032212 000402 BR 15$
6819 032214 013704 001210 21$: MOV STMP3,R4 ;GET CSI ERROR RETURN
6820 032220 005037 001236 15$: CLR $ESCAPE ;TEST IF LOOP ON ERROR
6821 032224 032777 001000 146706 BIT #BIT9, $SWR ;TEST IF LOOP ON ERROR
6822 032232 001360 BNE 13$ ;YES - SKIP
6823
6824 032234 013700 001254 MOV $TESTN,R0 ;SET REGISTERS TO LOOP ON ERROR OR
6825 032240 006300 R0 ;ESCAPE TO NEXT TEST.
6826 032242 016037 035270 001236 MOV $SWD08TB,R0), $ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
6827
6828 032250 162737 000002 001236 SUB #2, $ESCAPE ;ADJUST RETURN
6829 032256 000746 BR 13$ ;GO TO EXIT
6830
6831 .SBTTL GENERATE ECC WORD
6832 ;*
6833 ;* EACH TIME THIS ROUTINE IS CALLED THE ECC IS CALCULATED FOR THE
6834 ;* NEXT BIT TO BE READ OR WRITTEN AND THE BITS PREVIOUSLY
6835 ;* READ OR WRITTEN (STORED IN ECCHI AND ECCLO). THE RESULTS OF
6836 ;* THE CALCULATION ARE PLACED IN ECCHI AND ECCLO. THE LOW ORDER
6837 ;* 11 BITS OF ECCHI ARE PLACED IN E.ECPT FOR EASY COMPARISON
6838 ;* TO RKECPT.
6839 ;*
6840 ;* CALL: JSR PC,ECCGEN
6841 ;* RETURN: RTS R4
6842 032260 010046 ECCGEN: MOV R0, -(SP) ;STORE R0
6843 032262 012700 002316 MOV #PI,BIT,R0 ;GET ADDRESS OF NEXT BIT
6844 032266 032737 000002 002260 BIT #BIT1,L.CS1 ;CHECK IF WRITING
6845 032274 001002 BNE 12$ ;YES - SKIP

```

```

6845 032276 012700 002322      MOV      #M1.BIT,RO      ;ELSE CHANGE TO PRESENT BIT
6846
6847 032302 005710      12$: TST      (RO)      ;CHECK IF NEXT BIT 0
6848 032304 001410      BEQ      3$      ;YES - SKIP
6849 032306 032737 000001 002340      BIT      #1,ECCHI      ;NO, CHECK IF BIT 31 = 1
6850 032314 001410      BEQ      5$      ;NO, SET ECC XORED BIT
6851 032316 005037 002336      1$: CLR      ECCXOR      ;CLEAR ECC XORED BIT
6852 032322 000241      CLC      ;CLEAR CARRY FLOP
6853 032324 000410      BR       7$      ;SHIFT IN ECC BIT
6854
6855 032326 032737 000001 002340 3$: BIT      #1,ECCHI      ;CHECK IF BIT 31 = 1
6856 032334 001770      BEQ      1$      ;NO, CLEAR ECC XORED BIT
6857 032336 012737 000001 002336 5$: MOV      #1,ECCXOR      ;SET ECC XOR
6858 032344 000261      SEC      ;SET CARRY FLOP
6859 032346 006037 002342      7$: ROR      ECCLO
6860 032352 006037 002340      ROR      ECCHI
6861 032356 005737 002336      TST      ECCXOR      ;CHECK IF XOR NEEDED
6862 032362 001422      BEQ      10$     ;NO, RETURN
6863 032364 012746 020020      MOV      #BIT13:BIT4,-(SP) ;DO XOR OF ECC BITS
6864 032370 043716 002342      BIC      ECCL0,(SP)      ;0, 2, 11, 21, 23
6865 032374 042737 020020 002342      BIC      #BIT13:BIT4,ECCL0
6866 032402 052637 002342      BIS      (SP)+,ECCL0
6867 032406 012746 002400      MOV      #BIT10:BIT8,-(SP)
6868 032412 043716 002340      BIC      ECCHI,(SP)
6869 032416 042737 002400 002340      BIC      #BIT10:BIT8,ECCHI
6870 032424 052637 002340      BIS      (SP)+,ECCHI
6871 032430 013737 002340 002254 10$: MOV      ECCHI,E.ECPT ;STORE ECC PATTERN
6872 032436 042737 174000 002254      BIC      #174000,E.ECPT ;MASK UNSEEN BITS
6873 032444 012600      MOV      (SP)+,RO      ;RESTORE RO
6874 032446 000207      RTS      PC      ;RETURN
6875
6876      .SBTTL      SIMULATE ONE BIT OF WRITE DATA IN MAINTANENCE MODE
6877      ;*      ONE BIT OF DATA IS WRITE SIMULATED. THE CONTENTS OF
6878      ;*      RKMRI IS COMPUTED (PRECOMP ADVANCE AND DELAY, MAINTANENCE
6879      ;*      ENCODED WRITE DATA) FOR THE PROPER TRANSITION OF THE
6880      ;*      MAINTANENCE CLOCK AND CHECKED AT EACH TRANSITION. IF MRI
6881      ;*      IS INCORRECT AT ANY TRANSITION, AN ERROR IS SET UP TO
6882      ;*      CONTAIN A MESSAGE THAT IDENTIFIES THE TRANSITION WHEN THE
6883      ;*      ERROR OCCURRED.
6884      ;*
6885      ;*      CALL: JSR      PC,WRTBIT
6886      ;*      RETURN: RTS      PC+2      FOR NO ERROR RETURN
6887      ;*      RTS      PC      FOR ERROR IN MRI
6888 032450 052737 002400 002244      WRTBIT: BIS      #MCLK!MEWD,E.MRI ;CREATE EXPECTED MAINT. REG. 1
6889 032456 012762 000440 000026      MOV      #DMD!MCLK,RKMRI(R2) ;PROVIDE 1ST UPWARD TRANSITION
6890 032464 016237 000026 002204      MOV      RKMRI(R2),T.MRI ;STORE MAINT. REG. 1
6891 032472 023737 002244 002204      CMP      E.MRI,T.MRI ;CHECK IF MAINT REG 1 CORRECT
6892 032500 001416      BEQ      3$      ;YES, PROVIDE DOWNWARD TRANSITION
6893 032502 012737 032522 001236      MOV      #1$, $ESCAPE ;LOAD ESCAPE FOR LOOP ON ERROR
6894 032510 012737 045452 001512      MOV      #EMW1,EMW+2 ;LOAD ERROR MESSAGE
6895 032516 011646      MOV      (SP),-(SP) ;SAVE RETURN
6896 032520 000207      RTS      PC ;MRI INCORRECT ON UPWARD TRANSITION
6897
6898 032522 032777 001000 146410 1$: BIT      #SW9,$SWR ;CHECK IF LOOP ON ERROR
6899 032530 001402      BEQ      3$      ;NO, CONTINUE
6900 032532 000137 033404      JMP      63$     ;YES, LOOP ON ERROR

```

D11

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZ6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 133
 C296EA.F11 28-SEP-76 15:31 SIMULATE ONE BIT OF WRITE DATA IN MAINTENANCE MODE

SEQ 0133

```

6901
6902 032536 042737 014400 002244 35: BIC #MCLK!PCA!PCD,E.MR1 ;INITIALIZE MAINTENANCE REG. 1
6903 032544 052737 042000 002244 BIC #MEWD!WRTGAT,E.MR1
6904 032552 005737 002320 TST PR.BIT ;CHECK IF ONE
6905 032556 001152 9NE 20$ ;YES, SIMULATE ONE
6906 032560 005737 002322 TST M1.BIT ;CHECK IF PREVIOUS ONE
6907 032564 001023 10$ ;YES, NO TRANSITION
6908 032566 042737 002000 002244 BIC #MEWD,E.MR1 ;INDICATE TRANSITION
6909 032574 005737 002316 TST P1.BIT ;CHECK IF NEXT BIT = 1
6910 032600 001007 5$ ;YES, CHECK FOR PRECOMP ADVANCE
6911 032602 005737 002324 TST M2.BIT ;CHECK FOR PRECOMP. ADVANCE
6912 032606 001412 BEQ 10$ ;NO, CLOCK IN ZERO
6913 032610 052737 010000 002244 BIS #PCD,E.MR1 ;SET PRECOMP. DELAY
6914 032616 000406 BR 10$ ;CLOCK IN ZERO
6915
6916 032620 005737 002324 5$: TST M2.BIT ;CHECK FOR PRECOMP. ADVANCE
6917 032624 001003 10$ ;CLOCK IN ZERO
6918 032626 052737 004000 002244 BIS #PCA,E.MR1 ;SET PRECOMP. ADVANCE
6919 032634 012762 000040 000026 10$: MOV #DMD,RKMR1(R2) ;CLOCK IN DATA BIT
6920 032642 016237 000026 002204 MOV RKMR1(R2),T.MR1 ;STORE MR1
6921 032650 023737 002204 002244 CMP T.MR1,E.MR1 ;CHECK IF MR1 CORRECT
6922 032656 001416 12$ ;YES, CONTINUE
6923 032660 012737 032700 001236 MOV #11$, $ESCAPE ;LOAD ESCAPE FOR LOOP ON ERROR
6924 032666 012737 045540 001512 MOV #EMW2,EMW+2 ;LOAD ERROR MESSAGE
6925 032674 011646 MOV (SP),-(SP) ;SAVE RETURN
6926 032676 000207 RTS PC ;MR1 INCORRECT
6927
6928 032700 032777 001000 146232 11$: BIT #SW9,$SWR ;CHECK IF LOOP ON ERROR
6929 032706 001402 12$ ;NO, CONTINUE
6930 032710 000137 033404 JMP 63$ ;YES, LOOP ON ERROR
6931
6932 032714 052737 002400 002244 12$: BIS #MCLK!MEWD,E.MR1 ;CREATE EXPECTED MAINT REG 1
6933 032722 012762 000440 000026 MOV #DMD!MCLK,RKMR1(R2) ;PROVIDE 2ND UPWARD TRANSITION
6934 032730 016237 000026 002204 MOV RKMR1(R2),T.MR1 ;STORE MAINT REG. 1
6935 032736 023737 002244 002204 CMP E.MR1,T.MR1 ;CHECK IF MAINT REG. 1 CORRECT
6936 032744 001416 15$ ;YES, CONTINUE
6937 032746 012737 032766 001236 MOV #13$, $ESCAPE ;LOAD ESCAPE FOR LOOP ON ERROR
6938 032754 012737 045630 001512 MOV #EMW3,EMW+2 ;LOAD ERROR MESSAGE
6939 032762 011646 MOV (SP),-(SP) ;SAVE RETURN
6940 032764 000207 RTS PC ;MR1 INCORRECT
6941
6942 032766 032777 001000 146144 13$: BIT #SW9,$SWR ;CHECK IF LOOP ON ERROR
6943 032774 001402 15$ ;NO, CONTINUE
6944 032776 000137 033404 JMP 63$ ;YES, LOOP ON ERROR
6945
6946 033002 052737 002000 002244 15$: BIS #MEWD,E.MR1 ;RESET TRANSITION INDICATION
6947 033010 042737 000400 002244 BIC #MCLK,E.MR1
6948 033016 012762 000040 000026 MOV #DMD,RKMR1(R2) ;SUPPLY LAST PART OF DATA
6949 033024 016237 000026 002204 MOV RKMR1(R2),T.MR1 ;STORE MR1
6950 033032 023737 002204 002244 CMP T.MR1,E.MR1 ;CHECK IF MR1 CORRECT
6951 033040 001414 18$ ;YES, RETURN
6952 033042 012737 033062 001236 MOV #17$, $ESCAPE ;LOAD ESCAPE FOR LOOP ON ERROR
6953 033050 012737 045716 001512 MOV #EMW4,EMW+2 ;LOAD ERROR MESSAGE
6954 033056 011646 MOV (SP),-(SP) ;SAVE RETURN
6955 033060 000207 RTS PC ;MR1 INCORRECT
6956

```

RK611 DISKLESS CONTROLLER DIAGNOSTIC: FS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 134
 DZR6EA.P11 28-SEP-76 15:31 SIMULATE ONE BIT OF WRITE DATA IN MAINTANANCE MODE

02R6EA.P11 28-SEP-76 15:31

SIMULATE ONE BIT OF WRITE DATA IN MAINTANANCE MODE

6957	033062	032777	001000	146050	17\$:	BIT	#SW9, \$SWR	;CHECK IF LOOP ON ERROR
6958	033070	001145				BNE	63\$	;YES, LOOP ON ERROR
6959	033072	005037	001236		18\$:	CLR	\$ESCAPE	;CLEAR ESCAPE
6960	033076	062716	000002			ADD	#2, (SP)	;ADJUST RETURN
6961	033102	000207				RTS	PC	;RETURN
6962								
6963	033104	005737	002316		20\$:	TST	P1.BIT	;CHECK IN NEXT BIT A ONE
6964	033110	001007				BNE	30\$	;YES, CHECK IF PRECOMP DELAY
6965	033112	005737	002322			TST	M1.BIT	;CHECK FOR PRECOMP ADVANCE
6966	033116	001415				BEQ	40\$	;NO, CHECK FO MRI
6967	033120	052737	004000	002244		BIS	#PCA, E.MRI	;SET PRECOMP. ADVZNCE
6968	033126	000411				BR	40\$	;CHECK MRI
6969								
6970	033130	042737	000400	002244	30\$:	BIC	#MCLK, E.MRI	;RESET MAINT CLOCK IN EXPECTED MRI
6971	033136	005737	002322			TST	M1.BIT	;CHECK FOR PRECOMP DELAY
6972	033142	001003				BNE	40\$	;NO, CHECK MRI
6973	033144	052737	010000	002244		BIS	#PCD, E.MRI	;SET SET PRECOMP DELAY
6974	033152	012762	000040	000026	40\$:	MOV	#DMD, RKMRI(R2)	;CLOCK IN DATA BIT
6975	033160	016237	000026	002204		MOV	RKMRI(R2), T.MRI	;STORE MRI
6976	033166	023737	002204	002244		CMP	T.MRI, E.MRI	;CHECK MRI CORRECT
6977	033174	001414				BEQ	42\$	;YES, CLOCK IN REST OF BIT
6978	033176	012737	033216	001236		MOV	#41\$, \$ESCAPE	;LOAD ESCAPE FOR LOOP ON ERROR
6979	033204	012737	045540	001512		MOV	#EMW2, EMW+2	;LOAD ERROR MESSAGE
6980	033212	011646				MOV	(SP), -(SP)	;SAVE RETURN
6981	033214	000207				RTS	PC	;MRI INCORRECT
6982								
6983	033216	032777	001000	145714	41\$:	BIT	#SW9, \$SWR	;CHECK IF LOOP ON ERROR
6984	033224	001067				BNE	63\$	;YES, LOOP ON ERROR
6985	033226	052737	000400	002244	42\$:	BIS	#MCLK, E.MRI	;CHREATE EXPECTED MAINT. REG. 1
6986	033234	012762	000440	000026		MOV	#DMD, MCLK, RKMRI(R2)	;PROVIDE 2ND UPWARD TRANSITION
6987	033242	016237	000026	002204		MOV	RKMRI(R2), T.MRI	;STORE MAINT REG 1
6988	033250	023737	002244	002204		CMP	E.MRI, T.MRI	;CHECK IF MAINT REG 1 CORRECT
6989	033256	001414				BEQ	45\$	;YES, CONTINUE
6990	033260	012737	033300	001236		MOV	#43\$, \$ESCAPE	;LOAD ESCAPE
6991	033266	012737	045630	001512		MOV	#EMW3, EMW+2	;LOAD ERROR MESSAGE
6992	033274	011646				MOV	(SP), -(SP)	;SAVE RETURN
6993	033276	000207				RTS	PC	;MRI INCORRECT
6994								
6995	033300	032777	001000	145632	43\$:	BIT	#SW9, \$SWR	;CHECK IF LOOP ON ERROR
6996	033306	001036				BNE	63\$	;YES, LOOP ON ERROR
6997	033310	042737	002400	002244	45\$:	BIC	#MEWD, MCLK, E.MRI	;SET TRANSISTION
6998	033316	012762	000040	000026		MOV	#DMD, RKMRI(R2)	;CLOCK TRANSISTION
6999	033324	016237	000026	002204		MOV	RKMRI(R2), T.MRI	;STORE MRI
7000	033332	023737	002204	002244		CMP	T.MRI, E.MRI	;CHECK IF MRI CORRECT
7001	033340	001414				BEQ	50\$	;YES, RETURN
7002	033342	012737	033362	001236		MOV	#47\$, \$ESCAPE	;LOAD ESCAPE FOR LOOP ON ERROR
7003	033350	012737	045716	001512		MOV	#EMW4, EMW+2	;LOAD ERROR MESSAGE
7004	033356	011646				MOV	(SP), -(SP)	;SAVE RETURN
7005	033360	000207				RTS	PC	;MRI INCORRECT
7006								
7007	033362	032777	001000	145550	47\$:	BIT	#SW9, \$SWR	;CHECK IF LCOP ON ERROR
7008	033370	001005			</			

F11

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 135
 DZR6EA.P11 28-SEP-76 15:31 SIMULATE ONE BIT OF WRITE DATA IN MAINTENANCE MODE

SEQ 0135

```

7013 033404 012706 001100      63$: MOV    #STACK,SP      ;FORCE STACK
7014 033410 000177 145474      JMP     @SLPERA      ;LOOP ON ERROR
7015
7016      .SBTTL  SIMULATE ONE BIT OR READ DATA IN MAINTENANCE MODE
7017      ;*      ONE BIT OF DATA IS READ SIMULATED. NO CHECKING IS
7018      ;*      DONE BY THIS ROUTINE.
7019      ;*
7020      ;*      CALL: JSR      PC,RDBIT
7021      ;*      RETURN: RTS     PC
7022 033414 005737 002320      RDBIT: TST     PR,BIT      ;CHECK IF ONE
7023 033420 001024          BNE     10$      ;YES, SIMULATE ONE
7024 033422 005737 002322      TST     M1,BIT      ;CHECK IF PREVIOUS ONE
7025 033426 001404          BEQ     4$      ;NO, INSERT TRANSITION
7026 033430 012762 000440 000026  MOV     #DMD:MCLK,RKMR1(R2) ;YES, DO NOT INSERT TRANSITION
7027 033436 000403          BR      5$      ;CLOCK IN ZERO
7028
7029 033440 012762 001440 000026  4$:  MOV     #DMD:MCLK:MERD,RKMR1(R2) ;INSERT TRANSITION
7030 033446 012762 000040 000026  5$:  MOV     #DMD,RKMR1(R2) ;CLOCK IN ZERO
7031 033454 012762 000440 000026  MOV     #DMD:MCLK,RKMR1(R2)
7032 033462 012762 000040 000026  MOV     #DMD,RKMR1(R2)
7033 033470 000207          RTS     PC      ;RETURN
7034
7035 033472 012762 000440 000026  10$: MOV     #DMD:MCLK,RKMR1(R2) ;CLOCK IN ONE
7036 033500 012762 000040 000026  MOV     #DMD,RKMR1(R2)
7037 033506 012762 001440 000026  MOV     #DMD:MCLK:MERD,RKMR1(R2)
7038 033514 012762 000040 000026  MOV     #DMD,RKMR1(R2)
7039 033522 000207          RTS     PC      ;RETURN
7040
7041      .SBTTL  MEMORY CHECK ENABLE TRAP
7042      ;*      IF THE PROCESSOR IN USE HAS MEMORY PARITY OPTION,
7043      ;*      IT WILL BE ENABLED. THIS ROUTINE WILL PROCESS TRAPS CAUSED
7044      ;*      BY MEMORY PARITY ERRORS.
7045 033524 012737 033540 001236  MEMERR: MOV     #10$, $ESCAPE ;LOAD ESCAPE
7046 033532 011637 002310          MOV     (SP),TRAPPC ;STORE PC
7047 033536 104001          ERROR  1      ;REPORT MEM PARITY ERROR
7048 033540 005037 001236          CLR     $ESCAPE ;CLEAR ESCAPE
7049 033544 032777 001000 145366  10$: BIT     #SW9,@SWR ;CHECK IF LOOP ON ERROR
7050 033552 001001          BNE     15$      ;YES, FORCE STACK AND TRY AGAIN
7051 033554 000002          RTI          ;NO, RETURN
7052
7053 033556 012706 001100      15$: MOV     #STACK,SP      ;INITIALIZE STACK
7054 033562 000177 145322      JMP     @SLPERA      ;LOOP ON ERROR
7055
7056      .SBTTL  CLEAR INPUT BUFFER
7057 033566 012700 050440      CLRIBF: MOV     #IBUFF,R0 ;GET BUFFER POINTER
7058 033572 010146          MOV     R1,-(SP) ;STORE R1
7059 033574 012701 000400      MOV     #400,R1 ;SET COUNT FOR CLEAR
7060 033600 005020          1$:  CLR     (R0)+ ;CLEAR BUFFER WORD
7061 033602 005301          DEC     R1 ;DEC COUNT
7062 033604 001375          BNE     1$      ;LOOP UNTIL COUNT IS ZERO
7063 033606 012601          MOV     (SP)+,R1 ;RESTORE R1
7064 033610 000204          RTS     R4 ;RETURN
7065
7066      .SBTTL  BUILD DATA BUFFER
7067      ;*      THE PATTERN SPECIFIED IN THE CALL IS LOADED INTO THE DATA
7068      ;*      BUFFER. THE ENTIRE BUFFER IS ALWAYS LOADED (400(8) WORDS).
7069 033612      BLDAT:

```

G11

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 136
 DZR6EA.P11 28-SEP-76 15:31 BUILD DATA BUFFER

SEQ 0136

7069	033612	010046		MOV	R0,-(SP)	::PLSH R0 ON STACK
7070	033614	010146		MOV	R1,-(SP)	::PUSH R1 ON STACK
7071	033616	010346		MOV	R3,-(SP)	::PUSH R3 ON STACK
7072	033620	010546		MOV	R5,-(SP)	::PUSH R5 ON STACK
7073	033622	012700	051444	MOV	#0BUFF,R0	::LOAD R0 TO POINT TO BUFFER
7074	033626	012701	000400	MOV	#400,R1	::SET DATA LENGTH
7075	033632	021427	000001	CMP	(R4),#1	::TEST IF PATTERN 1(ALL ZERO)
7076	033636	001004		BNE	2\$	::NO - SKIP
7077	033640	005020		1\$: CLR	(R0)+	::ELSE CLEAR OBUFF
7078	033642	005301		DEC	R1	
7079	033644	001375		BNE	1\$	
7080	033646	000461		BR	13\$	
7081	033650	021427	000007	2\$: CMP	(R4),#7	::TEST IF PAT 7 (ALL ONES)
7082	033654	001006		BNE	4\$	::NO SKIP
7083	033656	012703	177777	MOV	#177777,R3	::ELSE SET BUFFER TO ALL ONES
7084	033662	010320		3\$: MOV	R3,(R0)+	
7085	033664	005301		DEC	R1	
7086	033666	001375		BNE	3\$	
7087	033670	000450		BR	13\$	
7088	033672	021427	000006	4\$: CMP	(R4),#6	::TEST IF PAT 6 (COMPOSIT ROTATING)
7089	033676	001003		BNE	5\$	::NO - SKIP
7090	033700	012705	050400	MOV	#PAT6,R5	::ELSE GET ADDRESS OF PATTERN 6
7091	033704	000427		BR	10\$	::GO LOAD BUFFER
7092	033706	021427	000005	5\$: CMP	(R4),#5	::TEST IF PATTERN 5 (ALT 1 & 0)
7093	033712	001006		BNE	7\$	::NO SKIP
7094	033714	012703	125252	MOV	#125252,R3	::ELSE LOAD OBUFF WITH PAT 5
7095	033720	010320		6\$: MOV	R3,(R0)+	
7096	033722	005301		DEC	R1	
7097	033724	001375		BNE	6\$	
7098	033726	000431		BR	13\$	
7099	033730	021427	000004	7\$: CMP	(R4),#4	::TEST IF PATTERN 4 (HI-LO FREQ MIX)
7100	033734	001003		BNE	8\$	::NO - SKIP
7101	033736	012705	050340	MOV	#PAT4,R5	::LOAD POINTER WITH ADD OF PAT 4
7102	033742	000410		BR	10\$	
7103	033744	021427	000003	8\$: CMP	(R4),#3	::TEST IF PAT 3 (MAX PRECOMP PHASE MIX)
7104	033750	001003		BNE	9\$	::NO - SKIP
7105	033752	012705	050300	MOV	#PAT3,R5	::LOAD POINTER WITH ADD OF PAT 3
7106	033756	000402		BR	10\$	::GO LOAD BUFFER
7107	033760	012705	050240	9\$: MOV	#PAT2,R5	::GET ADDRESS OF PAT 2
7108	033764	012703	000020	10\$: MOV	#16,R3	::SET PATTERN LENGTH COUNT
7109	033770	012520		11\$: MOV	(R5)+,(R0)+	::MOV PATTERN INTO OBUFF
7110	033772	005301		DEC	R1	::DEC COUNTERS
7111	033774	005303		DEC	R3	
7112	033776	001374		BNE	11\$	::LOOP UNTIL PATTERN IS MOVED
7113	034000	012705	051444	MOV	#0BUFF,R5	::SET R5 TO START OF BUFFER
7114	034004	012520		12\$: MOV	(R5)+,(R0)+	::REPEAT PATTERN THROUGH BUFFER
7115	034006	005301		DEC	R1	
7116	034010	001375		BNE	12\$	
7117	034012	005724		13\$: TST	(R4)+	::BUMP PAST PARAMETER
7118	034014	012605		MOV	(SP)+,R5	::POP STACK INTO R5
7119	034016	012603		MOV	(SP)+,R3	::POP STACK INTO R3
7120	034020	012601		MOV	(SP)+,R1	::POP STACK INTO R1
7121	034022	012600		MOV	(SP)+,R0	::POP STACK INTO R0
7122	034024	000204		RTS	R4	

7123
7124

.SBTTL LOAD "L" REGISTERS

H11

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZRG6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 137
 DZRG6E.P11 28-SEP-76 15:31 LOAD "L" REGISTERS

SEQ 0137

```

7125      ;*      THE "L" REGISTERS ARE LOADED FORM THE PARAMETERS FOLLOWING
7126      ;*      THE SUBROUTINE CALL. DIAGNOSTIC MODE IS SET IN L.MR1
7127      ;*      AND L.CS1 IS LOADED WITH THE COMMAND.
7128      ;*
7129      ;*      CALL:
7130      ;*      JSR      R4,LOADRK
7131      ;*      .WORD      CYLINDER
7132      ;*      .BYTE      ;SECTOR
7133      ;*      .BYTE      ;TRACK
7134      ;*      .WORD      ;BUFFER ADDRESS
7135      ;*      .WORD      ;WORD COUNT
7136      ;*      .WORD      ;COMMAND,FORMAT,DRIVE TYPE,&UPPER BUS ADDRESS BITS
7137      ;*
7138      ;*      RETURN:
7139      ;*      RTS      R4
7140      ;*
7141      034026 012437 002274      LOADRK: MOV      (R4)+,L.DCYL      ;LOAD CYLINDER
7142      034032 012437 002256      MOV      (R4)+,L.DA      ;      TRACK AND SECTOR
7143      034036 012437 002264      MOV      (R4)+,L.BA      ;      BUS ADDRESS
7144      034042 012437 002262      MOV      (R4)+,L.WC      ;      WORD COUNT
7145      034046 012737 000040      MOV      #DMD,L.MR1      ;SET DIAGNOSTIC MODE
7146      034054 005037 002272      CLR      L.ASOF      ;CLEAR OFFSET
7147      034060 005037 002270      CLR      L.CS2      ;CLEAR CS2
7148      034064 012437 002260      MOV      (R4)+,L.CS1      ;LOAD CS1
7149      034070 000204      RTS      R4      ;RETURN
7150
7151      .SBTTL  START THE OPERATION
7152      ;*      THE INFORMATION IN THE "L" REGISTERS ARE LOADED INTO THE
7153      ;*      RK611 REGISTERS IN A STARAIGHT TRANSFER.
7154
7155      034072 013762 002262 000002  OPSTRT: MOV      L.WC,RKWC(R2)      ;MOVE THE "L"REGISTERS INTO
7156      034100 013762 002264 000004      MOV      L.BA,RKBA(R2)      ;CORRESPONDING RK611 REGISTERS.
7157      034106 013762 002274 000020      MOV      L.DCYL,RKDCYL(R2)
7158      034114 013762 002266 000006      MOV      L.DA,RKDA(R2)
7159      034122 013762 002270 000010      MOV      L.CS2,RKCS2(R2)
7160      034130 013762 002276 000026      MOV      L.MR1,RKMR1(R2)
7161      034136 013762 002272 000016      MOV      L.ASOF,RKASOF(R2)
7162      034144 013762 002260 000000      MOV      L.CS1,RKCS1(R2)
7163      034152 000204      RTS      R4
7164
7165      .SBTTL  GET THE RK611 REGISTERS
7166      ;*      ALL THE RK611 REGISTERS EXCEPT THE DATA BUFFER ARE
7167      ;*      STORED IN THE "T" REGISTERS.
7168
7169      034154      GETREG:
7170      034154 010046      MOV      R0,-(SP)      ;;PUSH R0 ON STACK
7171      034156 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
7172      034160 010346      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
7173      034162 010200      MOV      R2,R0      ;GET RK BASE
7174      034164 012701 002160      MOV      #T.CS1,R1      ;GET START OF REGS
7175      034170 012703 000011      MOV      #11,R3      ;SET COUNT
7176      034174 012021      1$:      MOV      (R0)+,(R1)+      ;GET CS1 THRU DCYL
7177      034176 005303      DEC      R3
7178      034200 001375      BNE      1$      ;LOOP UNTIL DONE
7179
7180      034202 062700 000004      ADD      #4,R0      ;SKIP OVER DB AND SPARE

```

```

7181 034206 005021          CLR      (R1)+      ;CLEAR T.DB
7182 034210 012021          MOV      (R0)+,(R1)+  ;STORE MR1
7183 034212 012037 002212   MOV      (R0)+,T.ECPS  ;ECC POSITION
7184 034216 012037 002214   MOV      (R0)+,T.ECPT  ;ECC PATTERN
7185 034222 012037 002206   MOV      (R0)+,T.MR2   ;MR2
7186 034226 012037 002210   MOV      (R0)+,T.MR3   ;MR3
7187 034232 012603          MOV      (SP)+,R3     ;POP STACK INTO R3
7188 034234 012601          MOV      (SP)+,R1     ;POP STACK INTO R1
7189 034236 012600          MOV      (SP)+,R0     ;POP STACK INTO R0
7190 034240 000204          RTS      R4
7191
7192          .SBTTL  "FIRST SHALL BE LAST, LAST SHALL BE FIRST" SUBROUTINE
7193          ;*      THE CONTENTS OF R3 IS SWAPPED END FOR END, I.E. BIT 15
7194          ;*      BECOMES BIT 0 AND VICE VERSA,BIT 14 BECOMES BIT
7195          ;*      1 AND VICE VERSA, ETC.
7196          ;*
7197          ;*      CALL:
7198          ;*      JSR      R4,FSBLVV
7199          ;*      WITH R3 LOADED WITH THE WORD TO BE SWAPPED
7200          ;*
7201          ;*      RETURN:
7202          ;*      RTS      R4
7203          ;*      WITH R3 SWAPPED.
7204
7205          FSBLVV:
7206          034242 010046   MOV      R0,-(SP)     ;PUSH R0 ON STACK
7207          034244 010146   MOV      R1,-(SP)     ;PUSH R1 ON STACK
7208          034246 010446   MOV      R4,-(SP)     ;PUSH R4 ON STACK
7209          034250 005004   CLR      R4           ;CLEAR R4 FOR SWAPPED WORD
7210          034252 012700 000001   MOV      #BIT0,R0 ;SET FOR BIT TEST
7211          034256 012701 100000   MOV      #BIT15,R1 ;SET FOR BIT SET
7212          034262 030003   1$:  BIT      R0,R3    ;TEST IF BIT SET
7213          034264 001401   BEQ      2$           ;NO - SKIP
7214          034266 050104   BIS      R1,R4        ;SET CORRESPONDING BIT
7215          034270 006300   2$:  ASL      R0       ;SHIFT FOR NEXT BIT TEST
7216          034272 001403   BEQ      3$           ;LAST BIT TESTED - YES - EXIT
7217          034274 000241   CLC                    ;CLEAR CARRY
7218          034276 006001   ROR      R1           ;SHIFT FOR NEXT BIT SET
7219          034300 000770   BR       1$           ;LOOP
7220          034302 010403   3$:  MOV      R4,R3    ;STORED SWAPPED WORD
7221          034304 012604   MOV      (SP)+,R4    ;POP STACK INTO R4
7222          034306 012601   MOV      (SP)+,R1    ;POP STACK INTO R1
7223          034310 012600   MOV      (SP)+,R0    ;POP STACK INTO R0
7224          034312 000204   RTS      R4
7225
7226          .SBTTL  CHECK FOR MEMORY CHECK ENABLE
7227
7228          PARCHK: MOV      #20$,ERRVEC ;SET VECTOR FOR MEMORY PARITY CHECK
7229          034314 012737 034400 000004   MOV      #PR7,ERRVEC+2
7230          034322 012737 000340 000006   MOV      #0,BADPAR ;LOAD GOOD PARITY
7231          034330 012737 000000 002334   CLR      MEMPAR ;CLEAR FLAG
7232          034336 005037 002332   MOV      #MEMBAS,R3 ;LOAD REGISTER TO DETERMINE IF
7233          034342 012703 172100          ;MEMORY CHECK ENABLE AVAILABLE
7234
7235          034346 012704 000001   MOV      #1,R4 ;INITIALIZE MASK
7236          034352 012713 000001   16$: MOV      #PAR.EN,(R3) ;ENABLE MEMORY CHECK

```

J11

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 139
 DZR6EA.P11 28-SEP-76 15:31 CHECK FOR MEMORY CHECK ENABLE

SEG 0139

```

7237 034356 005713          TST      (R3)
7238 034360 050437 002332    BIS      R4,MEMPAR      ;SET FLAG
7239 034364 062703 000002    ADD      #2,R3
7240 034370 000241          CLC
7241 034372 006104          ROL      R4              ;CHECK IF FINISHED
7242 034374 001366          BNE      16$            ;NO, SET UP NEXT MEMORY PARITY MODULE
7243 034376 000406          BR       22$            ;RESTORE TRAP VECTOR
7244
7245 034400 022626          20$:    CMP      (SP)+,(SP)+      ;ADJUST STACK
7246 034402 062703 000002    ADD      #2,R3
7247 034406 000241          CLC
7248 034410 006104          ROL      R4              ;CHECK IF FINISHED
7249 034412 001357          BNE      16$            ;NO, GET NEXT LOCATION
7250 034414 012737 000006 000004 22$:    MOV      #ERRVEC+2,ERRVEC ;RESTORE TRAP CATCHER
7251 034422 005037 000006    CLR      ERRVEC+2
7252 034426 005737 002332    TST      MEMPAR          ;CHECK IF MEMORY CHECK ENABLE AVAILIABLE
7253 034432 001005          BNE      25$            ;YES, RETURN
7254 034434 012737 000116 000114    MOV      #MEMVEC+2,MEMVEC ;RESTORE TRAP CATCHER
7255 034442 005037 000116    CLR      MEMVEC+2
7256 034446 000207          25$:    RTS      PC              ;RETURN
7257 .SBTTL ROUTINE TO SIZE MEMORY
7258
7259 *****
7260 *CALL:
7261 *      JSR      PC,$SIZE
7262 *      RETURN
7263 *$LSTAD WILL CONTAIN:
7264 *      WITH KT11 OPTION      -- LAST VIRTUAL ADDRESS OF THE LAST BANK
7265 *      WITHOUT KT11 OPTION   -- LAST ABSOLUTE ADDRESS OF AVAILABLE MEMORY
7266 *$LSTBK WILL CONTAIN THE LAST BANK AS A SAF
7267 *$KT11 IS THE MEMORY MANAGEMENT KEY
7268 *BIT07 = 0 DON'T USE MEMORY MANAGEMENT
7269 *      MUST BE SETUP BEFORE THE CALL
7270 *BIT15 = 0 DON'T HAVE MEMORY MANAGEMENT OPTION
7271 *      DETERMINED BY ROUTINE
7272
7273 $SIZE: MOV      R0,-(SP)      ;;SAVE R0 ON THE STACK
7274      MOV      R1,-(SP)      ;;SAVE R1 ON THE STACK
7275      MOV      R2,-(SP)      ;;SAVE R2 ON THE STACK
7276      MOV      R3,-(SP)      ;;SAVE R3 ON THE STACK
7277      MOV      @#ERRVEC,-(SP) ;;SAVE PRESENT ERROR VECTOR PS & PC
7278      MOV      @#ERRVEC+2,-(SP)
7279      MOV      SP,R0          ;;SAVE THE STACK POINTER
7280 ;;SET THE ERRVEC PS TO THE PRESENT PS
7281      TRAP
7282      MOV      (SP)+,@#ERRVEC+2 ;;PUSH OLD PSW AND PC ON STACK
7283      MOV      #3776,R1        ;;SAVE THE PSW IN @#ERRVEC+2
7284      TSTB     (PC)+           ;;SETUP ADDRESS
7285      .WORD    200             ;;USE MEMORY MANAGEMENT?
7286      BPL      $SCORE          ;;SET TO USE MEMORY MANAGEMENT
7287      MOV      #SKTNEX,@#ERRVEC ;;BR IF NO
7288      TST      @#SRO           ;;SET FOR TIMEOUT
7289      BIS      #100000,$KT11   ;;KT11 ARE YOU THERE?
7290      CLR      -(SP)           ;;YES--SET KT11 KEY
7291      MOV      #KIPARO,R2      ;;INITIALIZE FOR "PAR" LOADING
7292      MOV      #+D8,R3         ;;ADDRESS OF FIRST "PAR"
                                ;;LOAD EIGHT "PAR.'S" AND EIGHT "PDR.'S"

```

K11

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 140
 DZR6EA.P11 28-SEP-76 15:31 ROUTINE TO SIZE MEMORY

SEG 0140

```

7293 034544 012762 077406 177740 1$: MOV #77406,-40(R2) ;;PDR = 4K, UP, READ/WRITE
7294 034552 011622 MOV (SP),(R2)+ ;;LOAD "PAR"
7295 034554 062716 000200 ADD #200,(SP) ;;UPDATE FOR NEXT "PAR"
7296 034560 077307 SOB R3,1$ ;;LOOP UNTIL ALL EIGHT ARE LOADED
7297 034562 012742 177600 MOV #177600,-(R2) ;;SETUP KIPAR7 FOR I/O
7298 034566 005042 CLR -(R2) ;;SETUP KIPAR6 FOR TESTING
7299 034570 012737 034606 000004 MOV #2$,2$ERRVEC ;;CATCH TIMEOUT IF NO SR3
7300 034576 012737 000020 172516 MOV #20,2$SR3 ;;ENABLE 22 BIT MODE
7301 034604 000401 BR 3$ ;;THIS PDP-11 HAS A SR3 REGISTER
7302 034606 022626 2$: CMP (SP)+,(SP)+ ;;CLEAN OFF THE STACK--NO SR3
7303 034610 005237 177572 3$: INC 2$SR0 ;;TURN ON MEMORY MANAGEMENT
7304 034614 012737 034640 000004 MOV #SKTOUT,2$ERRVEC ;;SET FOR TIME OUT
7305 034622 005737 143776 4$: TST 2$143776 ;;TRAP ON NON-EX-MEM
7306 034626 062712 000040 ADD #40,(R2) ;;MAKE A 1K STEP
7307 034632 023712 172356 CMP 2$KIPAR7,(R2) ;;LAST ONE?
7308 034636 101371 BHI 4$ ;;NO--TRY IT
7309 034640 011202 SKTOUT: MOV (R2),R2 ;;GET LAST BANK+1
7310 034642 005037 177572 CLR 2$SR0 ;;TURN OFF MEMORY MANAGEMENT
7311 034646 000421 BR $SIZE
7312 034650 042737 100000 034506 SKTNEX: BIC #100000,SKT11 ;;KT11 NON-EXISTENT
7313 034656 012737 034706 000004 SCORE: MOV #SCROUT,2$ERRVEC ;;SET FOR TIMEOUT
7314 034664 005002 CLR R2 ;;SET UP BANK
7315 034666 062701 004000 1$: ADD #4000,R1 ;;INCREMENT BY 1K
7316 034672 062702 000040 ADD #40,R2 ;;1K STEP
7317 034676 005711 TST (R1) ;;TRAP ON TIME OUT
7318 034700 022701 177776 CMP #177776,R1 ;;LAST ONE
7319 034704 001370 BNE 1$ ;;NO--TRY AGAIN
7320 034706 162701 004000 SCROUT: SUB #4000,R1
7321 034712 162702 000040 $SIZE: SUB #40,R2 ;;DROP BACK
7322 034716 010006 MOV R0,SP ;;RESTORE THE STACK
7323 034720 012637 000006 MOV (SP)+,2$ERRVEC+2 ;;RESTORE ERROR VECTOR
7324 034724 012637 000004 MOV (SP)+,2$ERRVEC
7325 034730 010137 034752 MOV R1,$LSTAD ;;LAST ADDRESS
7326 034734 010237 034754 MOV R2,$LSTBK ;;LAST BANK
7327 034740 012603 MOV (SP)+,R3 ;;RESTORE R3
7328 034742 012602 MOV (SP)+,R2 ;;RESTORE R2
7329 034744 012601 MOV (SP)+,R1 ;;RESTORE R1
7330 034746 012600 MOV (SP)+,R0 ;;RESTORE R0
7331 034750 000207 RTS PC
7332 034752 000000 $LSTAD: .WORD 0 ;;CONTAINS THE LAST ADDRESS
7333 034754 000000 $LSTBK: .WORD 0 ;;CONTAINS THE LAST BANK
7334 .SBTTL SCOPE HANDLER ROUTINE
7335
7336 *****
7337 *THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
7338 *AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
7339 *AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
7340 *THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
7341 *SW14=1 LOOP ON TEST
7342 *SW11=1 INHIBIT ITERATIONS
7343 *SW09=1 LOOP ON ERROR
7344 *SW08=1 LOOP ON TEST IN SWR<7:0>
7345 *CALL
7346 * SCOPE ;;SCOPE=IOT
7347
7348 034756 $SCOPE:

```

```

7349 034756 104407          CKSWR          ;; TEST FOR CHANGE IN SOFT-SWR
7350 034760 032777 040000 144152 1$: BIT      #BIT14,2SWR  ;; LOOP ON PRESENT TEST?
7351 034766 001131          BNE      $OVER  ;; YES IF SW14=1
7352          ;; *****START OF CODE FOR THE XOR TESTER*****
7353 034770 000416          $XTSTR: BR      6$          ;; IF RUNNING ON THE "XOR" TESTER CHANGE
7354          ;; THIS INSTRUCTION TO A "NOP" (NOP=240)
7355 034772 013746 000004          MOV      2$ERRVEC, -(SP)  ;; SAVE THE CONTENTS OF THE ERROR VECTOR
7356 034776 012737 035016 000004          MOV      2$5, 2$ERRVEC  ;; SET FOR TIMEOUT
7357 035004 005737 177060          TST      2$177060  ;; TIME OUT ON XOR?
7358 035010 012637 000004          MOV      (SP)+, 2$ERRVEC  ;; RESTORE THE ERROR VECTOR
7359 035014 000500          BR      $SVLAD  ;; GO TO THE NEXT TEST
7360 035016 022626          5$: CMP      (SP)+, (SP)+  ;; CLEAR THE STACK AFTER A TIME OUT
7361 035020 012637 000004          MOV      (SP)+, 2$ERRVEC  ;; RESTORE THE ERROR VECTOR
7362 035024 000440          BR      7$          ;; LOOP ON THE PRESENT TEST
7363 035026          6$: ;; *****END OF CODE FOR THE XOR TESTER*****
7364 035026 032777 000400 144104          BIT      #BIT08,2SWR  ;; LOOP ON SPEC. TEST?
7365 035034 001421          BEQ      2$          ;; BR IF NO
7366 035036 005046          CLR      -(SP)  ;; CLEAR A TEMP. LOCATION
7367 035040 117716 144074          MOVB     2$SWR, (SP)  ;; PICKUP THE DESIRED TEST NUMBER
7368 035044 001414          BEQ      8$          ;; BRANCH IF BAD TEST NUMBER IN SWR
7369 035046 022716 000042          CMP      #42, (SP)  ;; CHECK THE NUMBER IN THE SWR
7370 035052 002411          BLT      8$          ;; BRANCH IF TEST NUMBER IS OUT OF RANGE
7371 035054 011637 001102          MOV      (SP), $TSTNM  ;; UPDATE THE TEST NUMBER
7372 035060 005316          DEC      (SP)  ;; BACKUP BY ONE
7373 035062 006316          ASL      (SP)  ;; SCALE THE TEST NUMBER AS AN INDEX
7374 035064 062716 035270          ADD      2$SW08TBL, (SP)  ;; FORM THE ADDRESS OF TEST POINTER
7375 035070 013637 001106          MOV      2(SP)+, $LPADR  ;; SET LOOP ADDRESS TO DESIRED TEST
7376 035074 000466          BR      $OVER  ;; GO LOOP ON THE TEST
7377 035076 005726          8$: TST      (SP)+  ;; CLEAN THE BAD TEST NUMBER OFF OF THE STACK
7378 035100 105737 001103          2$: TSTB     $ERFLG  ;; HAS AN ERROR OCCURRED?
7379 035104 001421          BEQ      3$          ;; BR IF NO
7380 035106 123737 001115 001103          CMPB     $ERMAX, $ERFLG  ;; MAX. ERRORS FOR THIS TEST OCCURRED?
7381 035114 101015          BHI      3$          ;; BR IF NO
7382 035116 032777 001000 144014          BIT      #BIT09,2SWR  ;; LOOP ON ERROR?
7383 035124 001404          BEQ      4$          ;; BR IF NO
7384 035126 013737 001110 001106          7$: MOV      $LPERR, $LPADR  ;; SET LOOP ADDRESS TO LAST SCOPE
7385 035134 000446          BR      $OVER
7386 035136 105037 001103          4$: CLRB     $ERFLG  ;; ZERO THE ERROR FLAG
7387 035142 005037 001234          CLR      $TIMES  ;; CLEAR THE NUMBER OF ITERATIONS TO MAKE
7388 035146 000415          BR      1$          ;; ESCAPE TO THE NEXT TEST
7389 035150 032777 004000 143762          3$: BIT      #BIT11,2SWR  ;; INHIBIT ITERATIONS?
7390 035156 001011          BNE      1$          ;; BR IF YES
7391 035160 005737 001256          TST      $PASS  ;; IF FIRST PASS OF PROGRAM
7392 035164 001406          BEQ      1$          ;; INHIBIT ITERATIONS
7393 035166 005237 001104          INC      $ICNT  ;; INCREMENT ITERATION COUNT
7394 035172 023737 001234 001104          CMP      $TIMES, $ICNT  ;; CHECK THE NUMBER OF ITERATIONS MADE
7395 035200 002024          BGE      $OVER  ;; BR IF MORE ITERATION REQUIRED
7396 035202 012737 000001 001104          1$: MOV      #1, $ICNT  ;; REINITIALIZE THE ITERATION COUNTER
7397 035210 013737 035266 001234          MOV      $MXCNT, $TIMES  ;; SET NUMBER OF ITERATIONS TO DO
7398 035216 105237 001102          $SVLAD: INCB     $TSTNM  ;; COUNT TEST NUMBERS
7399 035222 113737 001102 001254          MOVB     $TSTNM, $TESTN  ;; SET TEST NUMBER IN APT MAILBOX
7400 035230 011637 001106          MOV      (SP), $LPADR  ;; SAVE SCOPE LOOP ADDRESS
7401 035234 011637 001110          MOV      (SP), $LPERR  ;; SAVE ERROR LOOP ADDRESS
7402 035240 005037 001236          CLR      $ESCAPE  ;; CLEAR THE ESCAPE FROM ERROR ADDRESS
7403 035244 112737 000001 001115          MOVB     #1, $ERMAX  ;; ONLY ALLOW ONE(1) ERROR ON NEXT TEST
7404 035252 013777 001102 143662          $OVER: MOV      $TSTNM, 2$DISPLAY  ;; DISPLAY TEST NUMBER

```

M11

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 142
 DZR6EA.P11 28-SEP-76 15:31 SCOPE HANDLER ROUTINE

SEQ 0142

```

7405 035260 013716 001106      MOV      $LPADR, (SP)      ;; FLDGE RETURN ADDRESS
7406 035264 000002      RTI                          ;; FIXES PS
7407 035266 003720      $MXCNT: 2000.      ;; MAX. NUMBER OF ITERATIONS
7408 035270      $SWC8TBL:
7409 035270 003426      .WORD      TST1+2      ;; STARTING ADDRESS OF TEST 1
7410 035272 004040      .WORD      TST2+2      ;; STARTING ADDRESS OF TEST 2
7411 035274 004452      .WORD      TST3+2      ;; STARTING ADDRESS OF TEST 3
7412 035276 005064      .WORD      TST4+2      ;; STARTING ADDRESS OF TEST 4
7413 035300 005476      .WORD      TST5+2      ;; STARTING ADDRESS OF TEST 5
7414 035302 006110      .WORD      TST6+2      ;; STARTING ADDRESS OF TEST 6
7415 035304 006522      .WORD      TST7+2      ;; STARTING ADDRESS OF TEST 7
7416 035306 007144      .WORD      TST10+2     ;; STARTING ADDRESS OF TEST 10
7417 035310 007414      .WORD      TST11+2     ;; STARTING ADDRESS OF TEST 11
7418 035312 007656      .WORD      TST12+2     ;; STARTING ADDRESS OF TEST 12
7419 035314 010126      .WORD      TST13+2     ;; STARTING ADDRESS OF TEST 13
7420 035316 010370      .WORD      TST14+2     ;; STARTING ADDRESS OF TEST 14
7421 035320 010704      .WORD      TST15+2     ;; STARTING ADDRESS OF TEST 15
7422 035322 011176      .WORD      TST16+2     ;; STARTING ADDRESS OF TEST 16
7423 035324 011724      .WORD      TST17+2     ;; STARTING ADDRESS OF TEST 17
7424 035326 012452      .WORD      TST20+2     ;; STARTING ADDRESS OF TEST 20
7425 035330 013200      .WORD      TST21+2     ;; STARTING ADDRESS OF TEST 21
7426 035332 013726      .WORD      TST22+2     ;; STARTING ADDRESS OF TEST 22
7427 035334 014462      .WORD      TST23+2     ;; STARTING ADDRESS OF TEST 23
7428 035336 015216      .WORD      TST24+2     ;; STARTING ADDRESS OF TEST 24
7429 035340 015744      .WORD      TST25+2     ;; STARTING ADDRESS OF TEST 25
7430 035342 016260      .WORD      TST26+2     ;; STARTING ADDRESS OF TEST 26
7431 035344 016552      .WORD      TST27+2     ;; STARTING ADDRESS OF TEST 27
7432 035346 017300      .WORD      TST30+2     ;; STARTING ADDRESS OF TEST 30
7433 035350 020034      .WORD      TST31+2     ;; STARTING ADDRESS OF TEST 31
7434 035352 020350      .WORD      TST32+2     ;; STARTING ADDRESS OF TEST 32
7435 035354 021032      .WORD      TST33+2     ;; STARTING ADDRESS OF TEST 33
7436 035356 021346      .WORD      TST34+2     ;; STARTING ADDRESS OF TEST 34
7437 035360 022116      .WORD      TST35+2     ;; STARTING ADDRESS OF TEST 35
7438 035362 022426      .WORD      TST36+2     ;; STARTING ADDRESS OF TEST 36
7439 035364 022764      .WORD      TST37+2     ;; STARTING ADDRESS OF TEST 37
7440 035366 023314      .WORD      TST40+2     ;; STARTING ADDRESS OF TEST 40
7441 035370 023650      .WORD      TST41+2     ;; STARTING ADDRESS OF TEST 41
7442 035372 024224      .WORD      TST42+2     ;; STARTING ADDRESS OF TEST 42
7443      ;;*****
7444      .SBTTL  LOOP ON INTERNAL ERROR
7445
7446 035374 032777 001000 143536 SCOP1$: BIT      #SW9, 2SWR      ;; CHECK IF LOOP ON ERROR
7447 035402 001405      BEQ        5$              ;; NO RETURN
7448 035404 105737 001103      TSTB     $ERFLG      ;; CHECK IF ERROR OCCURED
7449 035410 001402      BEQ        5$              ;; NO, RETURN
7450 035412 013716 001110      MOV      $LPERR, (SP)      ;; GO BACK TO BEGINNING OF LOOP
7451 035416 000002      5$:      RTI                          ;; RETURN
7452      .SBTTL  APT COMMUNICATIONS ROUTINE
7453
7454      ;;*****
7455 035420 112737 000001 035664 $ATY1:  MOVB     #1, $FFLG      ;; TO REPORT FATAL ERROR
7456 035426 112737 000001 035662 $ATY3:  MOVB     #1, $MFLG      ;; TO TYPE A MESSAGE
7457 035434 000403      BR         $ATYC
7458 035436 112737 000001 035664 $ATY4:  MOVB     #1, $FFLG      ;; TO ONLY REPORT FATAL ERROR
7459 035444      $ATYC:
7460 035444 010046      MOV      R0, -(SP)      ;; PUSH R0 ON STACK

```

N11

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 143
 DZR6EA.P11 28-SEP-76 15:31 APT COMMUNICATIONS ROUTINE

SEQ 0143

```

7461 035446 010146      MOV      R1, -(SP)      ;; PUSH R1 ON STACK
7462 035450 105737 035662  TSTB     $MFLG      ;; SHOULD TYPE A MESSAGE?
7463 035454 001450      BEQ      5$          ;; IF NOT: BR
7464 035456 122737 000001 001270  CMPB     $APTENV, $ENV      ;; OPERATING UNDER APT?
7465 035464 001031      SNE      3$          ;; IF NOT: BR
7466 035466 132737 000100 001271  BITB     $APTPOOL, $ENVM      ;; SHOULD SPOOL MESSAGES?
7467 035474 001425      BEQ      3$          ;; IF NOT: BR
7468 035476 017600 000004      MOV      @4(SP), R0      ;; GET MESSAGE ADDR.
7469 035502 062766 000002 000004  ADD      #2, 4(SP)      ;; BUMP RETURN ADDR.
7470 035510 005737 001250      TST      $MSGTYPE      ;; SEE IF DONE W/ LAST XMISSION?
7471 035514 001375      BNE      1$          ;; IF NOT: WAIT
7472 035516 010037 001264      MOV      R0, $MSGAD      ;; PUT ADDR IN MAILBOX
7473 035522 105720      TSTB     (R0)+      ;; FIND END OF MESSAGE
7474 035524 001376      BNE      2$          ;;
7475 035526 163700 001264      SUB      $MSGAD, R0      ;; SUB START OF MESSAGE
7476 035532 006200      ASF      R0          ;; GET MESSAGE LNTH IN WORDS
7477 035534 010037 001266      MOV      R0, $MSGGLT      ;; PUT LENGTH IN MAILBOX
7478 035540 012737 000004 001250  MOV      #4, $MSGTYPE      ;; TELL APT TO TAKE MSG.
7479 035546 000413      BR       5$          ;;
7480 035550 017637 000004 035574 3$:      MOV      @4(SP), 4$      ;; PUT MSG ADDR IN JSR LINKAGE
7481 035556 062766 000002 000004  ADD      #2, 4(SP)      ;; BUMP RETURN ADDRESS
7482 035564 013746 177776      MOV      177776, -(SP)      ;; PUSH 177776 ON STACK
7483 035570 004737 036444      JSR      PC, $TYPE      ;; CALL TYPE MACRO
7484 035574 000000      .WORD     0
7485 035576      4$:
7486 035576 105737 035664      5$:
7487 035602 001416      10$:      TSTB     $FFLG      ;; SHOULD REPORT FATAL ERROR?
7488 035604 005737 001270      BEQ      12$      ;; IF NOT: BR
7489 035610 001413      TST      $ENV      ;; RUNNING UNDER APT?
7490 035612 005737 001250      BEQ      12$      ;; IF NOT: BR
7491 035616 001375      11$:      TST      $MSGTYPE      ;; FINISHED LAST MESSAGE?
7492 035620 017637 000004 001252  BNE      11$      ;; IF NOT: WAIT
7493 035626 062766 000002 000004  MOV      @4(SP), $FATAL      ;; GET ERROR #
7494 035634 005237 001250      ADD      #2, 4(SP)      ;; BUMP RETURN ADDR.
7495 035640 105037 035664      INC      $MSGTYPE      ;; TELL APT TO TAKE ERROR
7496 035644 105037 035663      12$:      CLRB     $FFLG      ;; CLEAR FATAL FLAG
7497 035650 105037 035662      CLRB     $LFLG      ;; CLEAR LOG FLAG
7498 035654 012601      CLRB     $MFLG      ;; CLEAR MESSAGE FLAG
7499 035656 012600      MOV      (SP)+, R1      ;; POP STACK INTO R1
7500 035660 000207      MOV      (SP)+, R0      ;; POP STACK INTO R0
7501 035662 000      RTS      PC      ;; RETURN
7502 035663 000      $MFLG: .BYTE 0      ;; MESSG. FLAG
7503 035664 000      $LFLG: .BYTE 0      ;; LOG FLAG
7504      000      $FFLG: .BYTE 0      ;; FATAL FLAG
7505      035666      .EVEN
7506 000200      APTSIZE=200
7507 000001      APTENV=001
7508 000100      APTPOOL=100
7509 000040      APTCSUP=040
7510      .SBTTL  ERROR HANDLER ROUTINE
7511      *****
7512      *THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
7513      *SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
7514      *AND GO TO TYPERR ON ERROR
7515      *THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
7516      *SW15=1      HALT ON ERROR

```

```

7517 035666 104407 001103 143236 143226
7518 035666 105237 001102 143236 143226
7519 035670 001775 001102 143236 143226
7520 035674 013777 002000 143236 143226
7521 035676 032777 001402 143236 143226
7522 035704 001402 001240 143236 143226
7523 035712 104401 001112 143236 143226
7524 035714 005237 001116 143236 143226
7525 035720 011637 000002 143236 143226
7526 035724 162737 001114 143236 143226
7527 035730 117737 020000 143236 143226
7528 035736 032777 001004 143236 143226
7529 035744 004737 036066 143236 143226
7530 035752 004737 001245 143236 143226
7531 035754 104401 000001 001270 143236 143226
7532 035756 122737 001007 001114 036006 143236 143226
7533 035772 001007 001114 036006 143236 143226
7534 035774 113737 004737 035436 143236 143226
7535 035776 004737 000 143236 143226
7536 035778 000 143236 143226
7537 035780 000777 143122 143236 143226
7538 035782 005777 143122 143236 143226
7539 035784 100002 143122 143236 143226
7540 035786 100002 143122 143236 143226
7541 035788 000000 143122 143236 143226
7542 035790 104407 143122 143236 143226
7543 035792 032777 001000 143106 143236 143226
7544 035794 001402 001110 143236 143226
7545 035796 013716 001236 143236 143226
7546 035798 005737 001236 143236 143226
7547 035800 001402 001236 143236 143226
7548 035802 013716 001236 143236 143226
7549 035804 036052 022737 025042 000042 143236 143226
7550 035806 036052 022737 025042 000042 143236 143226
7551 035808 036060 001001 000000 143236 143226
7552 035810 036062 000000 143236 143226
7553 035812 036064 000002 143236 143226
7554 035814 036064 000002 143236 143226
7555 035816 036064 000002 143236 143226
7556 035818 036064 000002 143236 143226
7557 035820 036064 000002 143236 143226
7558 035822 036064 000002 143236 143226
7559 035824 036064 000002 143236 143226
7560 035826 036064 000002 143236 143226
7561 035828 036064 000002 143236 143226
7562 035830 036064 000002 143236 143226
7563 035832 036064 000002 143236 143226
7564 035834 036064 000002 143236 143226
7565 035836 036064 000002 143236 143226
7566 035838 036064 000002 143236 143226
7567 035840 036064 000002 143236 143226
7568 035842 036064 000002 143236 143226
7569 035844 036064 000002 143236 143226
7570 035846 036064 000002 143236 143226
7571 035848 036064 000002 143236 143226
7572 035850 036064 000002 143236 143226

```

```

; *SW13=1      INHIBIT ERROR TYPEOUTS
; *SW10=1      BELL ON ERROR
; *SW09=1      LOOP ON ERROR
; *CALL
; *          ERROR  N      ;;ERROR=EMT AND N=ERROR ITEM NUMBER

ERROR:
75:  CKSWR          ;;TEST FOR CHANGE IN SOFT-SWR
    INCB          SERFLG  ;;SET THE ERROR FLAG
    BEQ 75        ;;DON'T LET THE FLAG GO TO ZERO
    MOV STSTNM, @DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
    BIT #BIT10, @SWR  ;;BELL ON ERROR?
    BEQ 15        ;;NO - SKIP
    TYPE $BELL     ;;RING BELL
    INC $ERRCTL    ;;COUNT THE NUMBER OF ERRORS
    MOV (SP), $ERRPC ;;GET ADDRESS OF ERROR INSTRUCTION
    SUB #2, $ERRPC
    MOVB @ERRPC, $ITEMB ;;STRIP AND SAVE THE ERROR ITEM CODE
    BIT #BIT13, @SWR  ;;SKIP TYPEOUT IF SET
    BNE 20$        ;;SKIP TYPEOUTS
    JSR PC, TYPERR  ;;GO TO USER ERROR ROUTINE
    TYPE $SCRLF

20$:  CMPB #APTENV, $ENV  ;;RUNNING IN APT MODE
    BNE 25$        ;;NO SKIP APT ERROR REPORT
    MOVB $ITEMB, 21$ ;;SET ITEM NUMBER AS ERROR NUMBER
    JSR PC, SATY4   ;;REPORT FATAL ERROR TO APT

21$:  .BYTE 0
    .BYTE 0

22$:  BR 22$        ;;APT ERROR LOOP
25$:  TST @SWR      ;;HALT ON ERROR
    BPL 35$        ;;SKIP IF CONTINUE
    HALT           ;;HALT ON ERROR!
    CKSWR          ;;TEST FOR CHANGE IN SOFT-SWR
    BIT #BIT09, @SWR ;;LOOP ON ERROR SWITCH SET?
    BEQ 45$        ;;BR IF NO
    MOV $LPERR, (SP) ;;FUDGE RETURN FOR LOOPING
    TST $ESCAPE     ;;CHECK FOR AN ESCAPE ADDRESS
    BEQ 55$        ;;BR IF NONE
    MOV $ESCAPE, (SP) ;;FUDGE RETURN ADDRESS FOR ESCAPE

55$:  CMP #SENDAD, @42  ;;ACT-11 AUTO-ACCEPT?
    BNE 65$        ;;BRANCH IF NO
    HALT           ;;YES

65$:  RTI          ;;RETURN

;*****
;SBTTL TYPE ERROR ROUTINE
;*ENTRY JSR PC, TYPERR
;*RETURN RTS PC
;*
;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
;*ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
;*THE ERROR.

```

```

*****
7573          036066 104413          001102 001254  TYPEERR: SAVREG
7574          036070 113737          001102 001254  MOV B $STNM,STESTN ;SAVE TEST NUMBER FOR REPORT
7575          036076 042737          177400 001254  BIC B $177400,STESTN ;CLEAR UNUSED BITS
7576          036104 113700          001114          MOV B $ITEMB,RO ;ENTER ERROR NUMBER
7577          036110 042700          177400          BIC B $177400,RO ;CLEAR UNUSED BITS
7578          036114 005300          DEC RO ;FORM INDEX FOR ERROR TABLE
7579          036116 006300          ASL RO
7580          036120 006300          ASL RO
7581          036122 006300          ASL RO
7582          036124 062700          001330 15: ADD B$ERRTB,RO ;FORM ADDRESS OF ERROR ENTRY
7583          036130 012037          036144          MOV (RO)+,2$ ;GET EM POINTER
7584          036134 001404          BEQ 3$ ;BRANCH IF THERE ISN'T ONE
7585          036136 104401          001245          TYPE ,SCRLF ;TYPE CARRIAGE RETURN LINE FEED
7586          036142 104401          TYPE ;TYPE ERROR MESSAGE (EM)
7587          036144 000000          25: .WORD 0 ;EM POINTER GOES HERE
7588          036146 012037          036162 35: MOV (RO)+,4$ ;GET DH POINTER
7589          036152 001404          BEQ 5$ ;BRANCH IF THERE ISN'T ONE
7590          036154 104401          001245          TYPE ,SCRLF ;TYPE CR-LF
7591          036160 104401          TYPE ;TYPE DATA HEADER
7592          036162 000000          45: .WORD 0 ;DH POINTER GOES HERE
7593          036164 012001          55: MOV (RO)+,R1 ;GET DT POINTER
7594          036166 001445          BEQ 20$ ;BRANCH IF THERE ARE NONE
7595          036170 005004          CLR R4 ;RESET INDENT SWITCH
7596          036172 012000          MOV (RO)+,RO ;GET DF POINTER
7597          036174 012002          MOV (RO)+,R2 ;STORE NUMBER OF DH'S
7598          036176 104401          001245          TYPE ,SCRLF
7599          036202 112003          105: MOV B (RO)+,R3 ;GET & STORE NUMBER OF DATA WORDS
7600          036204 105720          TSTB (RO)+ ;BUMP PAST FORMAT WORD
7601          036206 005703          TST R3 ;TEST IF ANY DATA FOR THIS HEADER
7602          036210 001416          BEQ 14$ ;NO - SKIP DATA PRINT
7603          036212 005704          TST R4 ;CHECK IF INDENT WORDS
7604          036214 001004          BNE 12$ ;YES, GO INDENT
7605          036216 013146          115: MOV B(R1)+,-(SP) ;PUT FIRST DATA WORD ON STACK
7606          036220 104402          TYPOC ;TYPE IT
7607          036222 005303          DEC R3 ;MORE DATA WORDS
7608          036224 001403          BEQ 13$ ;NO-BRANCH
7609          036226 104401          042724 125: TYPE ,SPACE2 ;TYPE SEPARATORS
7610          036232 000771          BR 11$ ;LOOP
7611          036234 104401          001245 135: TYPE ,SCRLF ;TYPE (CR)(LF)
7612          036240 005710          TST (RO) ;CHECK IF NEXT HEADER AVAILABLE
7613          036242 001401          BEQ 14$ ;NO, DO NOT CHANGE INDENT
7614          036244 005104          COM R4 ;CHANGE INDENT
7615          036246 005302          145: DEC R2 ;MORE DH'S?
7616          036250 003414          BLE 20$ ;NO-BRANCH
7617          036252 012037          036272 155: MOV (RO)+,18$ ;GET NEXT DH POINTER
7618          036256 001751          BEQ 10$ ;IF NO HEADER GET DATA
7619          036260 005704          TST R4 ;INDENT?
7620          036262 001402          BEQ 17$ ;NO-BRANCH
7621          036264 104401          042724          TYPE ,SPACE2 ;INDENT
7622          036270 104401          175: TYPE ;TYPE DH
7623          036272 000000          185: .WORD 0 ;DH POINTER GOES HERE
7624          036274 104401          001245          TYPE ,SCRLF
7625          036300 000740          BR 10$ ;LOOP
7626          036302 104414          205: RESREG
7627          036304 005237          002314          INC ERRCNT ;INCREMENT ERROR COUNT

```

7629	036310	032777	010000	142622	BIF	#SW12,2SWR	:CHECK IF ABORT AFTER 20 ERRORS
7630	036316	001421			BEQ	25\$	:NO RETURN
7631	036320	022737	000024	002314	CMP	#20.,ERRCNT	:CHECK IF EROR THRESHOLD EXCEEDED
7632	036326	103015			BHIS	25\$	:NO RETURN
7633	036330	104401	042727		TYPE	.ABORT	:TYPE "PROGRAM HAS BEEN ABORTED BECAUSE
7634							: ERROR THRESHOLD EXCEEDED"
7635	036334	005737	000042		TST	42	:CHECK IF IN CHAIN MODE
7636	036340	001407			BEQ	30\$	:NO HALT
7637	036342	012737	000001	024700	MOV	#1,SEOPCT	:FORCE END OF PASS COUNT TO ONE FOR ABCRT
7638	036350	012706	001100		MOV	#STACK,SP	:INITIALIZE STACK
7639	036354	000137	024652		JMP	SEOP	:BRING IN NEXT PROGRAM IN CHAIN
7640	036360	000000					
7641	036362	000207					
				30\$:	HALT		
				25\$:	RTS	PC	

```
.SBTTL  CONTROLLED PROGRAM HALT
```

Address	Hex	Dec	Label	Instruction	Comment
7645	036364	013702	001324	CTRHLT: MOV	\$BASE, R2 ; SET '611 BASE
7646	036370	012762	005000	MOV	\$CLR, RKCS2(R2) ; CLEAR SUBSYSTEM
7647	036376	005037	001236	CLR	\$ESCAPE ; CLEAR ESCAPE
7648	036402	105037	001103	CLRB	\$ERFLG ; CLEAR ERROR FLAG
7649	036406	012706	001100	MOV	\$STACK, SP ; CLEAR STACK
7650	036412	104401	042661	TYPE	\$OPROD ; TYPE HALT MESSAGE
7651	036416	005737	000042	TST	42 ; TEST IF MONITOR PRESENT
7652	036422	001405		BEQ	\$S ; NO SKIP
7653	036424	012737	000001	MOV	\$1, \$EOPCT ; FORCE END OF PROGRAM
7654	036432	000137	024652	JMP	\$EOP ; JUMP TO END OF PASS
7655					
7656	036436	000000		\$S: HALT	PROGRAM
7657	036440	000137	002404	JMP	START1 ; RESTART IF CONTINUE

```
.SBTTL  TYPE ROUTINE
```

```

7660 ;*****
7661 ;ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
7662 ;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
7663 ;NOTE1:          $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
7664 ;NOTE2:          $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
7665 ;NOTE3:          $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
7666

```

ADDRESS	DATA	DISPATCH	STATUS	OPERATION	REMARKS	
7676	036444	105737	001157	STYPE: TSTB	STPFLG	
7677	036450	100002		BPL	IS	
7678	036452	000000		HALT	IS	
7679	036454	000430		BR	3\$	
7680	036456	010046		1\$: MOV	RO, -(SP)	
7681	036460	017600	000002	MOV	02(SP), RO	
7682	036464	122737	000001	001270	CMPB	*APTENV, \$ENV
7683	036472	001011		BNE	62\$	
7684	036474	132737	000100	001271	BITB	*APTSPool, \$ENVm

E12

 RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZ6E-A
 DZ6EA.P11 28-SEP-76 15:31 TYPE ROUTINE

MACY11 27(1006) 05-OCT-76 09:12 PAGE 147

SEQ 0147

7685	036502	001405			BEQ	62\$	::NO GO CHECK FOR CONSOLE
7686	036504	010037	036514		MOV	RO,61\$	::SETUP MESSAGE ADDRESS FOR APT
7687	036510	004737	035426		JSR	PC,\$ATY3	::SPOOL MESSAGE TO APT
7688	036514	000000			.WORD	0	::MESSAGE ADDRESS
7689	036516	132737	000040	001271	61\$: BITB	#APTC\$UP,\$ENVN	::APT CONSOLE SUPPRESSED
7690	036524	001003			62\$: BNE	60\$	::YES, SKIP TYPE OUT
7691	036526	112046			2\$: MOVB	(RO)+,-(SP)	::PUSH CHARACTER TO BE TYPED ONTO STACK
7692	036530	001005			BNE	4\$	::BR IF IT ISN'T THE TERMINATOR
7693	036532	005726			TST	(SP)+	::IF TERMINATOR POP IT OFF THE STACK
7694	036534	012600			60\$: MOV	(SP)+,RO	::RESTORE RO
7695	036536	062716	000002		3\$: ADD	#2,(SP)	::ADJUST RETURN PC
7696	036542	000002			RTI		::RETURN
7697	036544	122716	000011		4\$: CMPB	#HT,(SP)	::BRANCH IF <HT>
7698	036550	001430			BEQ	8\$	
7699	036552	122716	000200		CMPB	#CRLF,(SP)	::BRANCH IF NOT <CRLF>
7700	036556	001006			BNE	5\$	
7701	036560	005726			TST	(SP)+	::POP <CR><LF> EQUIV
7702	036562	104401			TYPE		::TYPE A CR AND LF
7703	036564	001245			\$CRLF		
7704	036566	105037	036722		CLRB	\$CHARCNT	::CLEAR CHARACTER COUNT
7705	036572	000755			BR	2\$	::GET NEXT CHARACTER
7706	036574	004737	036656		5\$: JSR	PC,\$TYPEC	::GO TYPE THIS CHARACTER
7707	036600	123726	001156		6\$: CMPB	\$FILLC,(SP)+	::IS IT TIME FOR FILLER CHARS.?
7708	036604	001350			BNE	2\$	::IF NO GO GET NEXT CHAR.
7709	036606	013746	001154		MOV	\$NULL,-(SP)	::GET # OF FILLER CHARS. NEEDED
7710							::AND THE NULL CHAR.
7711	036612	105366	000001		7\$: DECB	1(SP)	::DOES A NULL NEED TO BE TYPED?
7712	036616	002770			BLT	6\$	::BR IF NO--GO POP THE NULL OFF OF STACK
7713	036620	004737	036656		JSR	PC,\$TYPEC	::GO TYPE A NULL
7714	036624	105337	036722		DECB	\$CHARCNT	::DO NOT COUNT AS A COUNT
7715	036630	000770			BR	7\$	::LOOP
7716							
7717							
7718							
7719	036632	112716	000040		8\$: MOVB	#' ,(SP)	::REPLACE TAB WITH SPACE
7720	036636	004737	036656		9\$: JSR	PC,\$TYPEC	::TYPE A SPACE
7721	036642	132737	000007	036722	BITB	#7,\$CHARCNT	::BRANCH IF NOT AT
7722	036650	001372			BNE	9\$	::TAB STOP
7723	036652	005726			TST	(SP)+	::POP SPACE OFF STACK
7724	036654	000724			BR	2\$	::GET NEXT CHARACTER
7725	036656	105777	142266		\$TYPEC: TSTB	\$STPS	::WAIT UNTIL PRINTER IS READY
7726	036662	100375			BPL	\$TYPEC	
7727	036664	116677	000002	142260	MOVB	2(SP),2\$TPB	::LOAD CHAR TO BE TYPED INTO DATA REG.
7728	036672	122766	000015	000002	CMPB	#CR,2(SP)	::IS CHARACTER A CARRIAGE RETURN?
7729	036700	001003			BNE	1\$	::BRANCH IF NO
7730	036702	105037	036722		CLRB	\$CHARCNT	::YES--CLEAR CHARACTER COUNT
7731	036706	000406			BR	\$TYPEX	::EXIT
7732	036710	122766	000012	000002	1\$: CMPB	#LF,2(SP)	::IS CHARACTER A LINE FEED?
7733	036716	001402			BEQ	\$TYPEX	::BRANCH IF YES
7734	036720	105227			INCB	(PC)+	::COUNT THE CHARACTER
7735	036722	000000			\$CHARCNT: .WORD	0	::CHARACTER COUNT STORAGE
7736	036724	000207			\$TYPEX: RTS	PC	

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

;*****

F12

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZ6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 148
 DZ6EA.P11 28-SEP-76 15:31 BINARY TO OCTAL (ASCII) AND TYPE

SEQ 0148

```

7741 ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
7742 ;*OCTAL (ASCII) NUMBER AND TYPE IT.
7743 ;*STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
7744 ;*CALL:
7745 ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
7746 ;*      TYPPOS      ;;CALL FOR TYPEOUT
7747 ;*      .BYTE      N      ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
7748 ;*      .BYTE      M      ;;M=1 OR 0
7749 ;*                               ;;1=TYPE LEADING ZEROS
7750 ;*                               ;;0=SUPPRESS LEADING ZEROS
7751 ;*
7752 ;*STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
7753 ;*STYPOS OR STYPOC
7754 ;*CALL:
7755 ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
7756 ;*      TYPON      ;;CALL FOR TYPEOUT
7757 ;*
7758 ;*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
7759 ;*CALL:
7760 ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
7761 ;*      TYPOC      ;;CALL FOR TYPEOUT
7762 ;*
7763 036726 017646 000000 037151 STYPOS: MOV      2(SP),-(SP)      ;;PICKUP THE MODE
7764 036732 116637 000001      MOVVB 1(SP),SOFILL      ;;LOAD ZERO FILL SWITCH
7765 036740 112637 037153      MOVVB (SP)+,SOMODE+1      ;;NUMBER OF DIGITS TO TYPE
7766 036744 062716 000002      ADD      #2,(SP)      ;;ADJUST RETURN ADDRESS
7767 036750 000406      BR      STYPON
7768 036752 112737 000001 037151 STYPOC: MOVVB #1,SOFILL      ;;SET THE ZERO FILL SWITCH
7769 036760 112737 000006 037153      MOVVB #6,SOMODE+1      ;;SET FOR SIX(6) DIGITS
7770 036766 112737 000005 037150 STYPON: MOVVB #5,SOCNT      ;;SET THE ITERATION COUNT
7771 036774 010346      MOV      R3,-(SP)      ;;SAVE R3
7772 036776 010446      MOV      R4,-(SP)      ;;SAVE R4
7773 037000 010546      MOV      R5,-(SP)      ;;SAVE R5
7774 037 02 113704 037153      MOVVB SOMODE+1,R4      ;;GET THE NUMBER OF DIGITS TO TYPE
7775 037006 005404      NEG      R4
7776 037010 062704 000006      ADD      #6,R4      ;;SUBTRACT IT FOR MAX. ALLOWED
7777 037014 110437 037152      MOVVB R4,SOMODE      ;;SAVE IT FOR USE
7778 037020 113704 037151      MOVVB SOFILL,R4      ;;GET THE ZERO FILL SWITCH
7779 037024 016605 000012      MOV      12(SP),R5      ;;PICKUP THE INPUT NUMBER
7780 037030 005003      CLR      R3      ;;CLEAR THE OUTPUT WORD
7781 037032 006105      1$: ROL      R5      ;;ROTATE MSB INTO "C"
7782 037034 000404      BR      3$      ;;GO DO MSB
7783 037036 006105      2$: ROL      R5      ;;FORM THIS DIGIT
7784 037040 006105      ROL      R5
7785 037042 006105      ROL      R5
7786 037044 010503      MOV      R5,R3
7787 037046 006103      3$: ROL      R3      ;;GET LSB OF THIS DIGIT
7788 037050 105337 037152      DECB  SOMODE      ;;TYPE THIS DIGIT?
7789 037054 100016      BPL      7$      ;;BR IF NO
7790 037056 042703 177770      BIC      #177770,R3      ;;GET RID OF JUNK
7791 037062 001002      BNE      4$      ;;TEST FOR 0
7792 037064 005704      TST      R4      ;;SUPPRESS THIS 0?
7793 037066 001403      BEQ      5$      ;;BR IF YES
7794 037070 005204      4$: INC      R4      ;;DON'T SUPPRESS ANYMORE 0'S
7795 037072 052703 000060      BIS      #'0,R3      ;;MAKE THIS DIGIT ASCII
7796 037076 052703 000040      5$: BIS      #' ,R3      ;;MAKE ASCII IF NOT ALREADY

```

G12

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MC-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 149
 DZR6EA.P11 28-SEP-76 15:31 BINARY TO OCTAL (ASCII) AND TYPE

SEQ 0149

```

7797 037102 110337 037146      MOVB    R3,B5      ;;SAVE FOR TYPING
7798 037106 104401 037146      TYPE     B5      ;;GO TYPE THIS DIGIT
7799 037112 105337 037150 7$:  DECB     $OCNT      ;;COUNT BY 1
7800 037116 003347          BGT      2$      ;;BR IF MORE TO DO
7801 037120 002402          BLT      6$      ;;BR IF DONE
7802 037122 005204          INC      R4      ;;INSURE LAST DIGIT ISN'T A BLANK
7803 037124 000744          BR       2$      ;;GO DO THE LAST DIGIT
7804 037126 012605 6$:      MOV      (SP)+,R5      ;;RESTORE R5
7805 037130 012604          MOV      (SP)+,R4      ;;RESTORE R4
7806 037132 012603          MOV      (SP)+,R3      ;;RESTORE R3
7807 037134 016666 000002 000004 MOV      2(SP),4(SP)  ;;SET THE STACK FOR RETURNING
7808 037142 012616          MOV      (SP)+,(SP)
7809 037144 000002          RTI              ;;RETURN
7810 037146          000      8$:      .BYTE    0      ;;STORAGE FOR ASCII DIGIT
7811 037147          000      .BYTE    0      ;;TERMINATOR FOR TYPE ROUTINE
7812 037150          000      $OCNT:  .BYTE    0      ;;OCTAL DIGIT COUNTER
7813 037151          000      $OFILL: .BYTE    0      ;;ZERO FILL SWITCH
7814 037152 000000          $OMODE:  .WORD    0      ;;NUMBER OF DIGITS TO TYPE
7815          .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
7816
7817      ;*****
7818      ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
7819      ;SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
7820      ;NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
7821      ;BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
7822      ;REPLACED WITH SPACES.
7823      ;CALL:
7824      ;*      MOV      NUM,-(SP)      ;;PUT THE BINARY NUMBER ON THE STACK
7825      ;*      TYPDS          ;;GO TO THE ROUTINE
7826
7827      $TYPDS:
7828      037154 010046      MOV      R0,-(SP)      ;;PUSH R0 ON STACK
7829      037156 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
7830      037160 010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
7831      037162 010346      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
7832      037164 010546      MOV      R5,-(SP)      ;;PUSH R5 ON STACK
7833      037166 012746 020200 MOV      #20200,-(SP)  ;;SET BLANK SWITCH AND SIGN
7834      037172 016605 000020 MOV      20(SP),R5      ;;GET THE INPUT NUMBER
7835      037176 100004      BPL      1$      ;;BR IF INPUT IS POS.
7836      037200 005405      NEG      R5      ;;MAKE THE BINARY NUMBER POS.
7837      037202 112766 000055 000001 MOVB     #'-,1(SP)  ;;MAKE THE ASCII NUMBER NEG.
7838      037210 005000 1$:      CLR      R0      ;;ZERO THE CONSTANTS INDEX
7839      037212 012703 037370      MOV      #SDBLK,R3      ;;SETUP THE OUTPUT POINTER
7840      037216 112723 000040      MOVB     #'',(R3)+      ;;SET THE FIRST CHARACTER TO A BLANK
7841      037222 005002 2$:      CLR      R2      ;;CLEAR THE BCD NUMBER
7842      037224 016001 037360      MOV      $DTBL(R0),R1      ;;GET THE CONSTANT
7843      037230 160105 3$:      SUB      R1,R5      ;;FORM THIS BCD DIGIT
7844      037232 002402      BLT      4$      ;;BR IF DONE
7845      037234 005202      INC      R2      ;;INCREASE THE BCD DIGIT BY 1
7846      037236 000774      BR       3$
7847      037240 060105 4$:      ADD      R1,R5      ;;ADD BACK THE CONSTANT
7848      037242 005702      TST      R2      ;;CHECK IF BCD DIGIT=0
7849      037244 001002      BNE      5$      ;;FALL THROUGH IF 0
7850      037246 105716      TSTB     (SP)      ;;STILL DOING LEADING 0'S?
7851      037250 100407      BMI      7$      ;;BR IF YES
7852      037252 106316 5$:      ASLB      (SP)      ;;MSD?

```

H12

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZ6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 150
 DZ6EA.P11 28-SEP-76 15:31 CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

SEQ 3150

```

7853 037254 103003          BCC      6$          ;;BR IF NO
7854 037256 116663 000001 177777  MOVB    1(SP),-1(R3)  ;;YES--SET THE SIGN
7855 037264 052702 000060      BIS      #'0,R2      ;;MAKE THE BCD DIGIT ASCII
7856 037270 052702 000040 6$:      BIS      #' ,R2      ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
7857 037274 110223      MOVB    R2,(R3)+      ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
7858 037276 005720      TST      (R0)+      ;;JUST INCREMENTING
7859 037300 020027 000010      CMP      R0,#10      ;;CHECK THE TABLE INDEX
7860 037304 002746      BLT      2$          ;;GO DO THE NEXT DIGIT
7861 037306 003002      BGT      8$          ;;GO TO EXIT
7862 037310 010502      MOV      R5,R2      ;;GET THE LSD
7863 037312 000764      BR       6$          ;;GO CHANGE TO ASCII
7864 037314 105726 8$:      TSTB    (SP)+      ;;WAS THE LSD THE FIRST NON-ZERO?
7865 037316 100003      BPL      9$          ;;BR IF NO
7866 037320 116663 177777 177776  MOVB    -1(SP),-2(R3)  ;;YES--SET THE SIGN FOR TYPING
7867 037326 105013 9$:      CLRB    (R3)      ;;SET THE TERMINATOR
7868 037330 012605      MOV      (SP)+,R5      ;;POP STACK INTO R5
7869 037332 012603      MOV      (SP)+,R3      ;;POP STACK INTO R3
7870 037334 012602      MOV      (SP)+,R2      ;;POP STACK INTO R2
7871 037336 012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
7872 037340 012600      MOV      (SP)+,R0      ;;POP STACK INTO R0
7873 037342 104401 037370      TYPE    ,SDBLK      ;;NOW TYPE THE NUMBER
7874 037346 016666 000002 000004  MOV      2(SP),4(SP)  ;;ADJUST THE STACK
7875 037354 012616      MOV      (SP)+,(SP)
7876 037356 000002      RTI
7877 037360 023420      SOTBL: 10000.      ;;RETURN TO USER
7878 037362 001750      1000.
7879 037364 000144      100.
7880 037366 000012      10.
7881 037370 000004      SDBLK: .BLKW 4
7882      .SBTTL  TTY INPUT ROUTINE
7883
7884      ;*****
7885      .ENABL  LSB
7886 037400 000000      $TKCNT: .WORD 0      ;;NUMBER OF ITEMS IN QUEUE
7887 037402 000000      $TKQIN: .WORD 0      ;;INPUT POINTER
7888 037404 000000      $TKQOUT: .WORD 0      ;;OUTPUT POINTER
7889 037406 000001      $TKQSRT: .BLKB 1      ;;TTY KEYBOARD QUEUE
7890      $TKQEND=.
7891      .EVEN
7892
7893      ;*TK INITIALIZE ROUTINE
7894      ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
7895      ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
7896
7897      ;*CALL:
7898      ;*      JSR      PC,$TKINT
7899      ;*      RETURN
7900
7901 037410 005037 037400      $TKINT: CLR      $TKCNT      ;;CLEAR COUNT OF ITEMS IN QUEUE
7902 037414 012737 037406 037402  MOV      $TKQSRT,$TKQIN  ;;MOVE THE STARTING ADDRESS OF THE
7903 037422 013737 037402 037404  MOV      $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
7904 037430 012737 037460 000060  MOV      $TKSRV,$TKVEC  ;;INITIALIZE THE KEYBOARD VECTOR
7905 037436 012737 000200 000062  MOV      #200,$TKVEC+2  ;;BR LEVEL 4
7906 037444 005777 141476      TST      $TKB      ;;CLEAR DONE FLAG
7907 037450 012777 000100 141466  MOV      #100,$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
7908 037456 000207      RTS      PC      ;;RETURN TO CALLER

```

```

7909
7910      ;*TK SERVICE ROUTINE
7911      ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
7912      ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
7913      ;*IT IN THE QUEUE.
7914      ;*IF THE CHARACTER IS A "CONTROL-C" (1C) $TKINT IS CALLED AND
7915      ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (CTRHLT)
7916
7917 037460 117746 141462 $TKSRV: MOVB 2$TKB,-(SP)      ;: PICKUP THE CHARACTER
7918 037464 042716 177600      BIC  #1C177,(SP)      ;: STRIP THE JUNK
7919 037470 021627 000003      CMP  (SP),#3      ;: IS IT A CONTROL C?
7920 037474 001007      BNE  1$      ;: BRANCH IF NO
7921 037476 104401 040574      TYPE $CNTLC      ;: TYPE A CONTROL-C (1C)
7922 037502 004737 037410      JSR  PC,$TKINT      ;: INIT THE KEYBOARD
7923 037506 005726      TST  (SP)+      ;: CLEAN UP STACK
7924 037510 000137 036364      JMP  CTRHLT      ;: CONTROL C RESTART
7925 037514 021627 000007 1$: CMP  (SP),#7      ;: IS IT A CONTROL G?
7926 037520 001004      BNE  2$      ;: BRANCH IF NO
7927 037522 022737 000176 001140      CMP  #SWREG,SWR      ;: IS SOFT-SWR SELECTED?
7928 037530 001500      BLO  6$      ;: GO TO SWR CHANGE
7929
7930 037532      2$:      CMP  #1,$TKCNT      ;: IS THE QUEUE FULL?
7931 037532 022737 000001 037400      BNE  3$      ;: BRANCH IF NO
7932 037540 001004      TYPE  $BELL      ;: RING THE TTY BELL
7933 037542 104401 001240      TST  (SP)+      ;: CLEAN CHARACTER OFF OF STACK
7934 037546 005726      BR  5$      ;: EXIT
7935 037550 000451      3$: CMP  (SP),#23      ;: IS IT A CONTROL-S?
7936 037552 021627 000023      BNE  32$      ;: BRANCH IF NO
7937 037556 001021      CLR  2$TKS      ;: DISABLE TTY KEYBOARD INTERRUPTS
7938 037560 005077 141360      TST  (SP)+      ;: CLEAN CHAR OFF STACK
7939 037564 005726      31$: TSTB 2$TKS      ;: WAIT FOR A CHAR
7940 037566 105777 141352      BPL  31$      ;: LOOP UNTIL ITS THERE
7941 037572 100375      MOVB 2$TKB,-(SP)      ;: GET THE CHARACTER
7942 037574 117746 141346      BIC  #1C177,(SP)      ;: MAKE IT 7-BIT ASCII
7943 037600 042716 177600      CMP  (SP)+,#21      ;: IS IT A CONTROL-Q?
7944 037604 022627 000021      BNE  31$      ;: BRANCH IF NO
7945 037610 001366      MOV  #100,2$TKS      ;: REENABLE TTY KEYBOARD INTERRUPTS
7946 037612 012777 000100 141324      RTI      ;: RETURN
7947 037620 000002      32$: INC  $TKCNT      ;: COUNT THIS CHARACTER
7948 037622 005237 037400      CMP  (SP),#140      ;: IS IT UPPER CASE?
7949 037626 021627 000140      BLT  4$      ;: BRANCH IF YES
7950 037632 002405      CMP  (SP),#175      ;: IS IT A SPECIAL CHAR?
7951 037634 021627 000175      BGT  4$      ;: BRANCH IF YES
7952 037640 003002      BIC  #40,(SP)      ;: MAKE IT UPPER CASE
7953 037642 042716 000040      MOVB (SP)+,2$TKQIN      ;: AND PUT IT IN QUEUE
7954 037646 112677 177530      INC  $TKQIN      ;: UPDATE THE POINTER
7955 037652 005237 037402      CMP  $TKQIN,#$TKQEND      ;: GO OFF THE END?
7956 037656 023727 037402 037407      BNE  5$      ;: BRANCH IF NO
7957 037664 001003      MOV  #2$TKQRT,$TKQIN      ;: RESET THE POINTER
7958 037666 012737 037406 037402      RTI      ;: RETURN
7959 037674 000002      5$:
7960
7961      ;*****
7962      ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
7963      ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
7964      ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP

```

J12

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZ6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 152
 DZ6EA.P11 28-SEP-76 15:31 TTY INPUT ROUTINE

SEG 0152

```

7965      : *CALL WHEN OPERATING IN TTY INTERRUPT MODE.
7966 037676 022737 000176 001140 $CKSWR: CMP      #SWREG,SWR      ;; IS THE SOFT-SWR SELECTED
7967 037704 001124      BNE      15$      ;; EXIT IF NOT
7968 037706 105777 141232      TSTB     @STKS      ;; IS A CHAR WAITING?
7969 037712 100121      BPL      15$      ;; IF NOT, EXIT
7970 037714 117746 141226      MOVB     @STKB,-(SP)      ;; YES
7971 037720 042716 177600      BIC      #C177,(SP)      ;; MAKE IT 7-BIT ASCII
7972 037724 021627 000007      CMP      (SP),#7      ;; IS IT A CONTROL-G?
7973 037730 001300      BNE      25$      ;; IF NOT, PUT IT IN THE TTY QUEUE
7974      ;; AND EXIT
7975
7976      ;*****
7977      ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
7978      ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
7979      ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
7980 037732 123727 001134 000001 6$: CMPB     $AUTOB,#1      ;; ARE WE RUNNING IN AUTO-MODE?
7981 037740 001674      BEQ      25$      ;; BRANCH IF YES
7982 037742 005726      TST      (SP)+      ;; CLEAR CONTROL-G OFF STACK
7983 037744 004737 037410      JSR      PC,STKINT      ;; FLUSH THE TTY INPUT QUEUE
7984 037750 005077 141170      CLR      @STKS      ;; DISABLE TTY KEYBOARD INTERRUPTS
7985 037754 112737 000001 001135      MOVB     #1,$INTAG      ;; SET INTERRUPT MODE INDICATOR
7986
7987 037762 104401 040606      TYPE      , $CNTLG      ;; ECHO THE CONTROL-G (↑G)
7988 037766 104401 040613      $GTSWR: TYPE      , $MSWR      ;; TYPE CURRENT CONTENTS
7989 037772 013746 000176      MOV      $WREG,-(SP)      ;; SAVE SWREG FOR TYPEOUT
7990 037776 104402      TYPOC      ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
7991 040000 104401 040624      TYPE      , $MNEW      ;; PROMPT FOR NEW SWR
7992 040004 005046      19$: CLR      -(SP)      ;; CLEAR COUNTER
7993 040006 005046      CLR      -(SP)      ;; THE NEW SWR
7994 040010 105777 141130      7$: TSTB     @STKS      ;; CHAR THERE?
7995 040014 100375      BPL      7$      ;; IF NOT TRY AGAIN
7996
7997 040016 117746 141124      MOVB     @STKB,-(SP)      ;; PICK UP CHAR
7998 040022 042716 177600      BIC      #C177,(SP)      ;; MAKE IT 7-BIT ASCII
7999
8000 040026 021627 000003      CMP      (SP),#3      ;; IS IT A CONTROL-C?
8001 040032 001015      BNE      9$      ;; BRANCH IF NOT
8002 040034 104401 040574      TYPE      , $CNTLC      ;; YES, ECHO CONTROL-C (↑C)
8003 040040 062706 000006      ADD      #6,SP      ;; CLEAN UP STACK
8004 040044 123727 001135 000001      CMPB     $INTAG,#1      ;; REENABLE TTY KEYBOARD INTERRUPTS?
8005 040052 001003      BNE      8$      ;; BRANCH IF NO
8006 040054 012777 000100 141062      MOV      #100,@STKS      ;; ALLOW TTY KEYBOARD INTERRUPTS
8007 040062 000137 036364      8$: JMP      CTRHLT      ;; CONTROL-C RESTART
8008
8009
8010 040066 021627 000025      9$: CMP      (SP),#25      ;; IS IT A CONTROL-U?
8011 040072 001005      BNE      10$      ;; BRANCH IF NOT
8012 040074 104401 040601      TYPE      , $CNTLU      ;; YES, ECHO CONTROL-U (↑U)
8013 040100 062706 000006      20$: ADD      #6,SP      ;; IGNORE PREVIOUS INPUT
8014 040104 000737      BR      19$      ;; LET'S TRY IT AGAIN
8015
8016
8017 040106 021627 000015      10$: CMP      (SP),#15      ;; IS IT A <CR>?
8018 040112 001022      BNE      16$      ;; BRANCH IF NO
8019 040114 005766 000004      TST      4(SP)      ;; YES, IS IT THE FIRST CHAR?
8020 040120 001403      BEQ      11$      ;; BRANCH IF YES

```

K12

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 153
 DZR6EA.P11 28-SEP-76 15:31 TTY INPUT ROUTINE

SEQ 0153

```

8021 040122 016677 000002 141010      MOV      2(SP),2SWR      ;;SAVE NEW SWR
8022 040130 062706 000006      11$: ADD      #6,SP      ;;CLEAR UP STACK
8023 040134 104401 001245      14$: TYPE     $CRLF      ;;ECHO <CR> AND <LF>
8024 040140 123727 001135 000001      CMPB     $INTAG,#1      ;;RE-ENABLE TTY KBD INTERRUPTS?
8025 040146 001003      BNE      15$      ;;BRANCH IF NOT
8026 040150 012777 000100 140766      MOV      #100,2$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
8027 040156 000002      15$: RTI      ;;RETURN
8028 040160 004737 036656      16$: JSR      PC,$TYPEC      ;;ECHO CHAR
8029 040164 021627 000060      CMP      (SP),#60      ;;CHAR < 0?
8030 040170 002420      BLT      18$      ;;BRANCH IF YES
8031 040172 021627 000067      CMP      (SP),#67      ;;CHAR > ??
8032 040176 003015      BGT      18$      ;;BRANCH IF YES
8033 040200 042726 000060      BIC      #60,(SP)+      ;;STRIP-OFF ASCII
8034 040204 005766 000002      TST      2(SP)      ;;IS THIS THE FIRST CHAR
8035 040210 001403      BEQ      17$      ;;BRANCH IF YES
8036 040212 006316      ASL      (SP)      ;;NO, SHIFT PRESENT
8037 040214 006316      ASL      (SP)      ;;CHAR OVER TO MAKE
8038 040216 006316      ASL      (SP)      ;;ROOM FOR NEW ONE.
8039 040220 005266 000002      17$: INC      2(SP)      ;;KEEP COUNT OF CHAR
8040 040224 056616 177776      BIS      -2(SP),(SP)      ;;SET IN NEW CHAR
8041 040230 000667      BR      7$      ;;GET THE NEXT ONE
8042 040232 104401 001244      18$: TYPE     $QUES      ;;TYPE ?<CR><LF>
8043 040236 000720      BR      20$      ;;SIMULATE CONTROL-U
8044      .DSABL   LSB

```

```

8045
8046
8047      ;;*****
8048      ;;THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
8049      ;;CALL:
8050      ;;      RDCHR      ;;GET A CHARACTER FROM THE QUEUE
8051      ;;      RETURN HERE      ;;CHARACTER IS ON THE STACK
8052      ;;      ;;WITH PARITY BIT STRIPPED OFF
8053
8054
8055 040240 011646 000004 000002 $RDCHR: MOV      (SP),-(SP)      ;;PUSH DOWN THE PC AND
8056 040242 016666 000004      MOV      4(SP),2(SP)      ;;THE PS
8057 040250 005066 000004      CLR      4(SP)      ;;GET READY FOR A CHARACTER
8058 040254 005046      CLR      -(SP)      ;;PUT NEW PS ON STACK
8059 040256 012746 040264      MOV      #64$,-(SP)      ;;PUT NEW PC ON STACK
8060 040262 000002      RTI      ;;POP NEW PC AND PS
8061 040264      64$:
8062 040264 005737 037400      1$: TST      $TKCNT      ;;WAIT ON A CHARACTER
8063 040270 001775      BEQ      1$
8064 040272 005337 037400      DEC      $TKCNT      ;;DECREMENT THE COUNTER
8065 040276 117766 177102 000004      MOVB     2$TKQOUT,4(SP)      ;;GET ONE CHARACTER
8066 040304 005237 037404      INC      $TKQOUT      ;;UPDATE THE POINTER
8067 040310 023727 037404 037407      CMP      $TKQOUT,$$TKQEND      ;;DID IT GO OFF OF THE END?
8068 040316 001003      BNE      2$      ;;BRANCH IF NO
8069 040320 012737 037406 037404      MOV      $$TKQSRT,$TKQOUT      ;;RESET THE POINTER
8070 040326 000002      2$: RTI      ;;RETURN
8071      ;;*****
8072      ;;THIS ROUTINE WILL INPUT A STRING FROM THE TTY
8073      ;;CALL:
8074      ;;      RDLIN      ;;INPUT A STRING FROM THE TTY
8075      ;;      RETURN HERE      ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
8076      ;;      ;;TERMINATOR WILL BE A BYTE OF ALL 0'S

```

8077					\$RDLIN:	MOV R3,-(SP)	::SAVE R3
8078	040330	010346				CLR -(SP)	::CLEAR THE RUBOUT KEY
8079	040332	005046			1\$:	MOV \$TTYIN,R3	::GET ADDRESS
8080	040334	012703	040564		2\$:	CMP \$TTYIN+8.,R3	::BUFFER FULL?
8081	040340	022703	040574			BLOS 4\$	::BR IF YES
8082	040344	101456				RDCHR	::GO READ ONE CHARACTER FROM THE TTY
8083	040346	104410				MOVB (SP)+,(R3)	::GET CHARACTER
8084	040350	112613			10\$:	CMPB #177,(R3)	::IS IT A RUBOUT
8085	040352	122713	000177			BNE 5\$	::BR IF NO
8086	040356	001022				TST (SP)	::IS THIS THE FIRST RUBOUT?
8087	040360	005716				BNE 6\$	::BR IF NO
8088	040362	001007				MOVB #'\\,9\$	::TYPE A BACK SLASH
8089	040364	112737	000134	040562		TYPE 9\$	
8090	040372	104401	040562			MOV #-1,(SP)	::SET THE RUBOUT KEY
8091	040376	012716	177777		6\$:	DEC R3	::BACKUP BY ONE
8092	040402	005303				CMP R3,\$TTYIN	::STACK EMPTY?
8093	040404	020327	040564			BLO 4\$	::BR IF YES
8094	040410	103434				MOVB (R3),9\$	::SETUP TO TYPEOUT THE DELETED CHAR.
8095	040412	111337	040562			TYPE 9\$	::GO TYPE
8096	040416	104401	040562			BR 2\$	::GO READ ANOTHER CHAR.
8097	040422	000746			5\$:	TST (SP)	::RUBOUT KEY SET?
8098	040424	005716				BEQ 7\$	::BR IF NO
8099	040426	001406				MOVB #'\\,9\$	::TYPE A BACK SLASH
8100	040430	112737	000134	040562		TYPE 9\$	
8101	040436	104401	040562			CLR (SP)	::CLEAR THE RUBOUT KEY
8102	040442	005016			7\$:	CMPB #25,(R3)	::IS CHARACTER A CTRL U?
8103	040444	122713	000025			BNE 9\$	::BR IF NO
8104	040450	001003				TYPE \$CNTLU	::TYPE A CONTROL "U"
8105	040452	104401	040601			BR 1\$	::GO START OVER
8106	040456	000726			8\$:	CMPB #22,(R3)	::IS CHARACTER A "↑R"?
8107	040460	122713	000022			BNE 3\$	::BRANCH IF NO
8108	040464	001011				CLRB (R3)	::CLEAR THE CHARACTER
8109	040466	105013				TYPE \$CRLF	::TYPE A "CR" & "LF"
8110	040470	104401	001245			TYPE \$TTYIN	::TYPE THE INPUT STRING
8111	040474	104401	040564			BR 2\$	::GO PICKUP ANOTHER CHACTER
8112	040500	000717			4\$:	TYPE \$QUES	::TYPE A "?"
8113	040502	104401	001244			BR 1\$	::CLEAR THE BUFFER AND LOOP
8114	040506	000712			3\$:	MOVB (R3),9\$	::ECHO THE CHARACTER
8115	040510	111337	040562			TYPE 9\$	
8116	040514	104401	040562			CMPB #15,(R3)+	::CHECK FOR RETURN
8117	040520	122723	000015			BNE 2\$	::LOOP IF NOT RETURN
8118	040524	001305				CLRB -1(R3)	::CLEAR RETURN (THE 15)
8119	040526	105063	177777			TYPE \$LF	::TYPE A LINE FEED
8120	040532	104401	001246			TST (SP)+	::CLEAN RUBOUT KEY FROM THE STACK
8121	040536	005726				MOV (SP)+,R3	::RESTORE R3
8122	040540	012603				MOV (SP)-,(SP)	::ADJUST THE STACK AND PUT ADDRESS OF THE
8123	040542	011646				MOV 4(SP),2(SP)	::FIRST ASCII CHARACTER ON IT
8124	040544	016666	000004	000002		MOV \$TTYIN,4(SP)	
8125	040552	012766	040564	000004		RTI	::RETURN
8126	040560	000002				.BYTE 0	::STORAGE FOR ASCII CHAR. TO TYPE
8127	040562	000			9\$:	.BYTE 0	::TERMINATOR
8128	040563	000				.BLKB 8.	::RESERVE 8 BYTES FOR TTY INPUT
8129	040564	000010			\$TTYIN:	.ASCIZ /↑C/<15><12>	::CONTROL "C"
8130	040574	041536	005015	000	\$CNTLC:	.ASCIZ /↑U/<15><12>	::CONTROL "U"
8131	040601	136	006525	000012	\$CNTLU:	.ASCIZ /↑G/<15><12>	::CONTROL "G"
8132	040606	043536	005015	000	\$CNTLG:	.ASCIZ	

M12

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 155
 DZR6EA.P11 28-SEP-76 15:31 TTY INPUT ROUTINE

SEQ 0155

```

8133 040613 015 051412 051127 $MSWR: .ASCIZ <15><12>/SWR = /
8134 040620 036440 000040
8135 040624 020040 042516 020127 $MNEW: .ASCIZ / NEW = /
8136 040632 020075 000
8137 040636
8138 .EVEN
8139 .SBTTL READ AN OCTAL NUMBER FROM THE TTY
8140
8141 ;*****
8142 ;THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
8143 ;CHANGE IT TO BINARY.
8144 ;THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
8145 ;OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
8146 ;FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
8147 ;THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
8148 ;CALL:
8149 ;* RDOCT ;READ AN OCTAL NUMBER
8150 ;* RETURN HERE ;LOW ORDER BITS ARE ON TOP OF THE STACK
8151 ;* ;HIGH ORDER BITS ARE IN $HIOCT
8152 040636 011646 $RDOCT: MOV (SP),-(SP) ;PROVIDE SPACE FOR THE
8153 040640 016666 000004 000002 MOV 4(SP),2(SP) ;INPUT NUMBER
8154 040646 010046 MOV RO,-(SP) ;PUSH RO ON STACK
8155 040650 010146 MOV R1,-(SP) ;PUSH R1 ON STACK
8156 040652 010246 MOV R2,-(SP) ;PUSH R2 ON STACK
8157 040654 104411 1$: RDLIN ;READ AN ASCII LINE
8158 040656 012600 MOV (SP)+,RO ;GET ADDRESS OF 1ST CHARACTER
8159 040660 010037 040764 MOV RO,$$ ;AND SAVE IT
8160 040664 005001 CLR R1 ;CLEAR DATA WORD
8161 040666 005002 CLR R2
8162 040670 112046 2$: MOVB (RO)+,-(SP) ;PICKUP THIS CHARACTER
8163 040672 001420 BEQ 3$ ;IF ZERO GET OUT
8164 040674 122716 000060 CMPB #'0,(SP) ;MAKE SURE THIS CHARACTER
8165 040700 003026 BGT 4$ ;IS AN OCTAL DIGIT
8166 040702 122716 000067 CMPB #'7,(SP)
8167 040706 002423 BLT 4$
8168 040710 006301 ASL R1 ;;*2
8169 040712 006102 ROL R2 ;;*4
8170 040714 006301 ASL R1 ;;*8
8171 040716 006102 ROL R2
8172 040720 006301 ASL R1
8173 040722 006102 ROL R2
8174 040724 042716 177770 BIC #'C7,(SP) ;STRIP THE ASCII JUNK
8175 040730 062601 ADD (SP)+,R1 ;ADD IN THIS DIGIT
8176 040732 000756 BR 2$ ;LOOP
8177 040734 005726 3$: TST (SP)+ ;CLEAN TERMINATOR FROM STACK
8178 040736 010166 000012 MOV R1,12(SP) ;SAVE THE RESULT
8179 040742 010237 040774 MOV R2,$HIOCT
8180 040746 012602 MOV (SP)+,R2 ;POP STACK INTO R2
8181 040750 012601 MOV (SP)+,R1 ;POP STACK INTO R1
8182 040752 012600 MOV (SP)+,RO ;POP STACK INTO RO
8183 040754 000002 RTI ;RETURN
8184 040756 005726 4$: TST (SP)+ ;CLEAN PARTIAL FROM STACK
8185 040760 105010 CLRB (RO) ;SET A TERMINATOR
8186 040762 104401 TYPE ;TYPE UP THRU THE BAD CHAR.
8187 040764 000000 5$: .WORD 0
8188 040766 104401 001244 TYPE ,SQUES ;;"?" "CR" & "LF"

```

N12

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 156
 DZR6EA.P11 28-SEP-76 15:31 READ AN OCTAL NUMBER FROM THE TTY

SEQ 0156

```

8189 040772 000730          BR      15          ;; TRY AGAIN
8190 040774 000000          $HI OCT: .WORD 0          ;; HIGH ORDER BITS GO HERE
8191                          .SBTTL  SAVE AND RESTORE RO-R5 ROUTINES
8192
8193          ;;*****
8194          ;;SAVE RO-R5
8195          ;;CALL:
8196          ;;      SAVREG
8197          ;;UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
8198          ;;
8199          ;;TOP---(+16)
8200          ;;+2---(+18)
8201          ;;+4---R5
8202          ;;+6---R4
8203          ;;+8---R3
8204          ;;+10---R2
8205          ;;+12---R1
8206          ;;+14---R0
8207
8208 040776          $SAVREG:
8209 040776 010046          MOV      R0,-(SP)          ;; PUSH R0 ON STACK
8210 041000 010146          MOV      R1,-(SP)          ;; PUSH R1 ON STACK
8211 041002 010246          MOV      R2,-(SP)          ;; PUSH R2 ON STACK
8212 041004 010346          MOV      R3,-(SP)          ;; PUSH R3 ON STACK
8213 041006 010446          MOV      R4,-(SP)          ;; PUSH R4 ON STACK
8214 041010 010546          MOV      R5,-(SP)          ;; PUSH R5 ON STACK
8215 041012 016646 000022          MOV      22(SP),-(SP)          ;; SAVE PS OF MAIN FLOW
8216 041016 016646 000022          MOV      22(SP),-(SP)          ;; SAVE PC OF MAIN FLOW
8217 041022 016646 000022          MOV      22(SP),-(SP)          ;; SAVE PS OF CALL
8218 041026 016646 000022          MOV      22(SP),-(SP)          ;; SAVE PC OF CALL
8219 041032 000002          RTI
8220
8221          ;;RESTORE RO-R5
8222          ;;CALL:
8223          ;;      RESREG
8224 041034          $RESREG:
8225 041034 012666 000022          MOV      (SP)+,22(SP)          ;; RESTORE PC OF CALL
8226 041040 012666 000022          MOV      (SP)+,22(SP)          ;; RESTORE PS OF CALL
8227 041044 012666 000022          MOV      (SP)+,22(SP)          ;; RESTORE PC OF MAIN FLOW
8228 041050 012666 000022          MOV      (SP)+,22(SP)          ;; RESTORE PS OF MAIN FLOW
8229 041054 012605          MOV      (SP)+,R5          ;; POP STACK INTO R5
8230 041056 012604          MOV      (SP)+,R4          ;; POP STACK INTO R4
8231 041060 012603          MOV      (SP)+,R3          ;; POP STACK INTO R3
8232 041062 012602          MOV      (SP)+,R2          ;; POP STACK INTO R2
8233 041064 012601          MOV      (SP)+,R1          ;; POP STACK INTO R1
8234 041066 012600          MOV      (SP)+,R0          ;; POP STACK INTO R0
8235 041070 000002          RTI
8236
8237          .SBTTL  POWER DOWN AND UP ROUTINE
8238
8239          ;;*****
8240
8241          :POWER DOWN ROUTINE
8242 041072 017737 140042 002356 $PWRDN: MOV      2SWR,SAVSWR          ;; SAVE SWITCH REGISTER
8243 041100 012737 041120 000024          MOV      #PWRUP,PWRVEC          ;; SET UP VECTOR
8244 041106 012737 000340 0C0026          MOV      #PR7,PWRVEC+2

```

B13

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZ6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 157
 DZ6EA.P11 28-SEP-76 15:31 POWER DOWN AND UP ROUTINE

SEQ 0157

```

8245 041114 000000          HALT
8246 041116 000776          BR      -2      ;HANG UP
8247
8248
8249
8250
8251 041120 005037 041214    ;*****
8252 041124 012737 000144 041216 ;POWER UP ROUTINE
8253 041132 005237 041214    $PWUP: CLR      $PWCT      ;LOAD WAIT COUNT
8254 041136 001375          IS:  MOV      $100,$PWCT+2
8255 041140 005337 041216    INC      $PWCT      ;WAIT FOR TELETYPE
8256 041144 001372          BNE      IS
8257 041146 012737 041072 000024 DEC      $PWCT+2
8258 041154 012737 000340 000026 BNE      IS
8259 041162 012706 001100      MOV      $SPWRDN,$PWVEC ;SET UP FOR POWER DOWN VECTOR
8260 041166 104401 041220      MOV      $PR7,$PWVEC+2
8261 041172 004737 074314      MOV      $STACK,$P      ;FORCE STACK
8262 041176 013777 002356 137734 TYPE      $POWER      ;TYPE POWER
8263 041204 013702 001324      JSR      PC,$PARCHK ;REINITIALIZE MEMORY CHECK ENABLE
8264 041210 000177 137672      MOV      $AVSWR,$SWR ;RESTORE SWITCH REGISTER
8265          JMP      $SLPADR ;GO BACK TO LAST TEST
8266 041214 000000 000000    $PWCT: .WORD 0,0      ;TELETYPE TIME OUT
8267 041220 047520 042527 000122 $POWER: .ASCIZ 'POWER'
8268          EVEN
8269          .SBTTL TRAP DECODER
8270
8271
8272
8273
8274
8275
8276
8277 041226 010046          ;*****
8278 041230 016600 000002    ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
8279 041234 005740          ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
8280 041236 111000          ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
8281 041240 006300          ;*GO TO THAT ROUTINE.
8282 041242 016000 041262    STRAP: MOV      R0,-(SP)      ;:SAVE R0
8283 041246 000200          MOV      2(SP),R0      ;:GET TRAP ADDRESS
8284          TST      -(R0)      ;:BACKUP BY 2
8285          MOVB     (R0),R0      ;:GET RIGHT BYTE OF TRAP
8286          ASL      R0      ;:POSITION FOR INDEXING
8287          MOV      $TRPAD(R0),R0 ;:INDEX TO TABLE
8288          RTS      R0      ;:GO TO ROUTINE
8289
8290
8291
8292
8293
8294
8295
8296
8297
8298
8299 041250 011646          ;:THIS IS USE TO HANDLE THE "GETPRI" MACRO
8300 041252 016666 000004 000002 STRAP2: MOV      (SP),-(SP) ;:MOVE THE PC DOWN
8301 041260 000002          MOV      4(SP),2(SP) ;:MOVE THE PSW DOWN
8302          RTI      ;:RESTORE THE PSW
8303
8304
8305
8306
8307
8308
8309
8310
8311
8312
8313
8314
8315
8316
8317
8318
8319
8320
8321
8322
8323
8324
8325
8326
8327
8328
8329
8330
8331
8332
8333
8334
8335
8336
8337
8338
8339
8340
8341
8342
8343
8344
8345
8346
8347
8348
8349
8350
8351
8352
8353
8354
8355
8356
8357
8358
8359
8360
8361
8362
8363
8364
8365
8366
8367
8368
8369
8370
8371
8372
8373
8374
8375
8376
8377
8378
8379
8380
8381
8382
8383
8384
8385
8386
8387
8388
8389
8390
8391
8392
8393
8394
8395
8396
8397
8398
8399
8400
8401
8402
8403
8404
8405
8406
8407
8408
8409
8410
8411
8412
8413
8414
8415
8416
8417
8418
8419
8420
8421
8422
8423
8424
8425
8426
8427
8428
8429
8430
8431
8432
8433
8434
8435
8436
8437
8438
8439
8440
8441
8442
8443
8444
8445
8446
8447
8448
8449
8450
8451
8452
8453
8454
8455
8456
8457
8458
8459
8460
8461
8462
8463
8464
8465
8466
8467
8468
8469
8470
8471
8472
8473
8474
8475
8476
8477
8478
8479
8480
8481
8482
8483
8484
8485
8486
8487
8488
8489
8490
8491
8492
8493
8494
8495
8496
8497
8498
8499
8500
8501
8502
8503
8504
8505
8506
8507
8508
8509
8510
8511
8512
8513
8514
8515
8516
8517
8518
8519
8520
8521
8522
8523
8524
8525
8526
8527
8528
8529
8530
8531
8532
8533
8534
8535
8536
8537
8538
8539
8540
8541
8542
8543
8544
8545
8546
8547
8548
8549
8550
8551
8552
8553
8554
8555
8556
8557
8558
8559
8560
8561
8562
8563
8564
8565
8566
8567
8568
8569
8570
8571
8572
8573
8574
8575
8576
8577
8578
8579
8580
8581
8582
8583
8584
8585
8586
8587
8588
8589
8590
8591
8592
8593
8594
8595
8596
8597
8598
8599
8600
8601
8602
8603
8604
8605
8606
8607
8608
8609
8610
8611
8612
8613
8614
8615
8616
8617
8618
8619
8620
8621
8622
8623
8624
8625
8626
8627
8628
8629
8630
8631
8632
8633
8634
8635
8636
8637
8638
8639
8640
8641
8642
8643
8644
8645
8646
8647
8648
8649
8650
8651
8652
8653
8654
8655
8656
8657
8658
8659
8660
8661
8662
8663
8664
8665
8666
8667
8668
8669
8670
8671
8672
8673
8674
8675
8676
8677
8678
8679
8680
8681
8682
8683
8684
8685
8686
8687
8688
8689
8690
8691
8692
8693
8694
8695
8696
8697
8698
8699
8700
8701
8702
8703
8704
8705
8706
8707
8708
8709
8710
8711
8712
8713
8714
8715
8716
8717
8718
8719
8720
8721
8722
8723
8724
8725
8726
8727
8728
8729
8730
8731
8732
8733
8734
8735
8736
8737
8738
8739
8740
8741
8742
8743
8744
8745
8746
8747
8748
8749
8750
8751
8752
8753
8754
8755
8756
8757
8758
8759
8760
8761
8762
8763
8764
8765
8766
8767
8768
8769
8770
8771
8772
8773
8774
8775
8776
8777
8778
8779
8780
8781
8782
8783
8784
8785
8786
8787
8788
8789
8790
8791
8792
8793
8794
8795
8796
8797
8798
8799
8800
8801
8802
8803
8804
8805
8806
8807
8808
8809
8810
8811
8812
8813
8814
8815
8816
8817
8818
8819
8820
8821
8822
8823
8824
8825
8826
8827
8828
8829
8830
8831
8832
8833
8834
8835
8836
8837
8838
8839
8840
8841
8842
8843
8844
8845
8846
8847
8848
8849
8850
8851
8852
8853
8854
8855
8856
8857
8858
8859
8860
8861
8862
8863
8864
8865
8866
8867
8868
8869
8870
8871
8872
8873
8874
8875
8876
8877
8878
8879
8880
8881
8882
8883
8884
8885
8886
8887
8888
8889
8890
8891
8892
8893
8894
8895
8896
8897
8898
8899
8900
8901
8902
8903
8904
8905
8906
8907
8908
8909
8910
8911
8912
8913
8914
8915
8916
8917
8918
8919
8920
8921
8922
8923
8924
8925
8926
8927
8928
8929
8930
8931
8932
8933
8934
8935
8936
8937
8938
8939
8940
8941
8942
8943
8944
8945
8946
8947
8948
8949
8950
8951
8952
8953
8954
8955
8956
8957
8958
8959
8960
8961
8962
8963
8964
8965
8966
8967
8968
8969
8970
8971
8972
8973
8974
8975
8976
8977
8978
8979
8980
8981
8982
8983
8984
8985
8986
8987
8988
8989
8990
8991
8992
8993
8994
8995
8996
8997
8998
8999
9000
9001
9002
9003
9004
9005
9006
9007
9008
9009
9010
9011
9012
9013
9014
9015
9016
9017
9018
9019
9020
9021
9022
9023
9024
9025
9026
9027
9028
9029
9030
9031
9032
9033
9034
9035
9036
9037
9038
9039
9040
9041
9042
9043
9044
9045
9046
9047
9048
9049
9050
9051
9052
9053
9054
9055
9056
9057
9058
9059
9060
9061
9062
9063
9064
9065
9066
9067
9068
9069
9070
9071
9072
9073
9074
9075
9076
9077
9078
9079
9080
9081
9082
9083
9084
9085
9086
9087
9088
9089
9090
9091
9092
9093
9094
9095
9096
9097
9098
9099
9100
9101
9102
9103
9104
9105
9106
9107
9108
9109
9110
9111
9112
9113
9114
9115
9116
9117
9118
9119
9120
9121
9122
9123
9124
9125
9126
9127
9128
9129
9130
9131
9132
9133
9134
9135
9136
9137
9138
9139
9140
9141
9142
9143
9144
9145
9146
9147
9148
9149
9150
9151
9152
9153
9154
9155
9156
9157
9158
9159
9160
9161
9162
9163
9164
9165
9166
9167
9168
9169
9170
9171
9172
9173
9174
9175
9176
9177
9178
9179
9180
9181
9182
9183
9184
9185
9186
9187
9188
9189
9190
9191
9192
9193
9194
9195
9196
9197
9198
9199
9200
9201
9202
9203
9204
9205
9206
9207
9208
9209
9210
9211
9212
9213
9214
9215
9216
9217
9218
9219
9220
9221
9222
9223
9224
9225
9226
9227
9228
9229
9230
9231
9232
9233
9234
9235
9236
9237
9238
9239
9240
9241
9242
9243
9244
9245
9246
9247
9248
9249
9250
9251
9252
9253
9254
9255
9256
9257
9258
9259
9260
9261
9262
9263
9264
9265
9266
9267
9268
9269
9270
9271
9272
9273
9274
9275
9276
9277
9278
9279
9280
9281
9282
9283
9284
9285
9286
9287
9288
9289
9290
9291
9292
9293
9294
9295
9296
9297
9298
9299
9300
9301
9302
9303
9304
9305
9306
9307
9308
9309
9310
9311
9312
9313
9314
9315
9316
9317
9318
9319
9320
9321
9322
9323
9324
9325
9326
9327
9328
9329
9330
9331
9332
9333
9334
9335
9336
9337
9338
9339
9340
9341
9342
9343
9344
9345
9346
9347
9348
9349
9350
9351
9352
9353
9354
9355
9356
9357
9358
9359
9360
9361
9362
9363
9364
9365
9366
9367
9368
9369
9370
9371
9372
9373
9374
9375
9376
9377
9378
9379
9380
9381
9382
9383
9384
9385
9386
9387
9388
9389
9390
9391
9392
9393
9394
9395
9396
9397
9398
9399
9400
9401
9402
9403
9404
9405
9406
9407
9408
9409
9410
9411
9412
9413
9414
9415
9416
9417
9418
9419
9420
9421
9422
9423
9424
9425
9426
9427
9428
9429
9430
9431
9432
9433
9434
9435
9436
9437
9438
9439
9440
9441
9442
9443
9444
9445
9446
9447
9448
9449
9450
9451
9452
9453
9454
9455
9456
9457
9458
9459
9460
9461
9462
9463
9464
9465
9466
9467
9468
9469
9470
9471
9472
9473
9474
9475
9476
9477
9478
9479
9480
9481
9482
9483
9484
9485
9486
9487
9488
9489
9490
9491
9492
9493
9494
9495
9496
9497
9498
9499
9500
9501
9502
9503
9504
9505
9506
9507
9508
9509
9510
9511
9512
9513
9514
9515
9516
9517
9518
9519
9520
9521
9522
9523
9524
9525
9526
9527
9528
9529
9530
9531
9532
9533
9534
9535
9536
9537
9538
9539
9540
9541
9542
9543
9544
9545
9546
9547
9548
9549
9550
9551
9552
9553
9554
9555
9556
9557
9558
9559
9560
9561
9562
9563
9564
9565
9566
9567
9568
9569
9570
9571
9572
9573
9574
9575
9576
9577
9578
9579
9580
9581
9582
9583
9584
9585
9586
9587
9588
9589
9590
9591
9592
9593
9594
9595
9596
9597
9598
9599
9600
9601
9602
9603
9604
9605
9606
9607
9608
9609
9610
9611
9612
9613
9614
9615
9616
9617
9618
9619
9620
9621
9622
9623
9624
9625
9626
9627
9628
9629
9630
9631
9632
9633
9634
9635
9636
9637
9638
9639
9640
9641
9642
9643
9644
9645
9646
9647
9648
9649
9650
9651
9652
9653
9654
9655
9656
9657
9658
9659
9660
9661
9662
9663
9664
9665
9666
9667
9668
9669
9670
9671
9672
9673
9674
9675
9676
9677
9678
9679
9680
9681
9682
9683
9684
9685
9686
9687
9688
9689
9690
9691
9692
9693
9694
9695
9696
9697
9698
9699
9700
9701
9702
9703
9704
9705
9706
9707
9708
9709
9710
9711
9712
9713
9714
9715
9716
9717
9718
9719
9720
9721
9722
9723
9724
9725
9726
9727
9728
9729
9730
9731
9732
9733
9734
9735
9736
9737
9738
9739
9740
9741
9742
9743
9744
9745
9746
9747
9748
9749
9750
9751
9752
9753
9754
9755
9756
9757
9758
9759
9760
9761
9762
9763
9764
9765
9766
9767
9768
9769
9770
9771
9772
9773
9774
9775
9776
9777
9778
9779
9780
9781
9782
9783
9784
9785
9786
9787
9788
9789
9790
9791
9792
9793
9794
9795
9796
9797
9798
9799
9800
9801
9802
9803
9804
9805
9806
9807
9808
9809
9810
9811
9812
9813
9814
9815
9816
9817
9818
9819
9820
9821
9822
9823
9824
9825
9826
9827
9828
9829
9830
9831
9832
9833
9834
9835
9836
9837
9838
9839
9840
9841
9842
9843
9844
9845
9846
9847
9848
9849
9850
9851
9852
9853
9854
9855
9856
9857
9858
9859
9860
9861
9862
9863
9864
9865
9866
9867
9868
9869
9870
9871
9872
9873
9874
9875
9876
9877
9878
9879
9880
9881
9882
9883
9884
9885
9886
9887
9888
9889
9890
9891
9892
9893
9894
9895
9896
9897
9898
9899
9900
9901
9902
9903
9904
9905
9906
9907
9908
9909
9910
9911
9912
9913
9914
9915
9916
9917
9918
9919
9920
9921
9922
9923
9924
9925
9926
9927
9928
9929
9930
9931
9932
9933
9934
9935
9936
9937
9938
9939
9940
9941
9942
9943
9944
9945
9946
9947
9948
9949
9950
9951
9952
9953
9954
9955
9956
9957
9958
9959
9960
9961
9962
9963
9964
9965
9966
9967
9968
9969
9970
9971
9972
9973
9974
9975
9976
9977
9978
9979
9980
9981
9982
9983
9984
9985
9986
9987
9988
9989
9990
9991
9992
9993
9994
9995
9996
9997
9998
9999

```

C13

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 158
 DZR6EA.P11 28-SEP-76 15:31 TRAP TABLE

SEQ 0158

8301	041266	036752	STYPOC	::CALL=TYPOC	TRAP+2(104402)	TYPE OCTAL NUMBER (WITH LEADING ZEROS)
8302	041270	036726	STYPOS	::CALL=TYPOS	TRAP+3(104403)	TYPE OCTAL NUMBER (NO LEADING ZEROS)
8303	041272	036766	STYPON	::CALL=TYPON	TRAP+4(104404)	TYPE OCTAL NUMBER (AS PER LAST CALL)
8304	041274	037154	STYPOS	::CALL=TYPOS	TRAP+5(104405)	TYPE DECIMAL NUMBER (WITH SIGN)
8305						
8306	041276	037766	SGTSWR	::CALL=GTSMR	TRAP+6(104406)	GET SOFT-SWR SETTING
8307						
8308	041300	037676	SKSWR	::CALL=CKSWR	TRAP+7(104407)	TEST FOR CHANGE IN SOFT-SWR
8309	041302	040240	SRDCHR	::CALL=RDCHR	TRAP+10(104410)	TTY TYPEIN CHARACTER ROUTINE
8310	041304	040330	SRDLIN	::CALL=RDLIN	TRAP+11(104411)	TTY TYPEIN STRING ROUTINE
8311	041306	040636	SRDOCT	::CALL=RDOCT	TRAP+12(104412)	READ AN OCTAL NUMBER FROM TTY
8312	041310	040776	SSAVREG	::CALL=SAVREG	TRAP+13(104413)	SAVE R0-R5 ROUTINE
8313	041312	041034	SRRESREG	::CALL=RESREG	TRAP+14(104414)	RESTORE R0-R5 ROUTINE
8314	041314	035374	SCOPI\$	::CALL=SCOPI	TRAP+15(104415)	INTERNAL LOOP ON ERROR

```

8315 .SBTTL DATA TABLE FOR PRINT OUT
8316
8317 041316 001254 002310 DT000: .WORD $TESTN,TRAPPC
8318 041322 001254 001116 002220 DT0004: .WORD $TESTN,$ERRPC,E.CS1,T.CS1,T.CS2,T.ER,T.DS
8319 041330 002160 002170 002174
8320 041336 002172
8321 041340 001254 001116 002244 DT0005: .WORD $TESTN,$ERRPC,E.MR1,T.MR1
8322 041346 002204
8323 041350 001254 001116 002246 DT0006: .WORD $TESTN,$ERRPC,E.MR2,T.MR2
8324 041356 002206
8325 041360 001254 001116 002250 DT0007: .WORD $TESTN,$ERRPC,E.MR3,T.MR3
8326 041366 002210
8327 041370 001254 001116 002244 DT0010: .WORD $TESTN,$ERRPC,E.MR1,T.MR1,PR.BIT,M1.BIT,BITCNT
8328 041376 002204 002320 002322
8329 041404 002326
8330 041406 001254 001116 002220 DT0011: .WORD $TESTN,$ERRPC,E.CS1,T.CS1,PR.BIT,M1.BIT,BITCNT,T.CS2,T.ER,T.DS
8331 041414 002160 002320 002322
8332 041422 002326 002170 002174
8333 041430 002172
8334 041432 001254 001116 002240 DT0014: .WORD $TESTN,$ERRPC,E.DCYL,T.DCYL,E.DA,T.DA
8335 041440 002200 002226 002166
8336 041446 001254 001116 002244 DT0017: .WORD $TESTN,$ERRPC,E.MR1,T.MR1,P1.BIT,PR.BIT,M1.BIT,M2.BIT,BITCNT
8337 041454 002204 002316 002320
8338 041462 002322 002324 002326
8339 041470 001254 001116 002254 DT0021: .WORD $TESTN,$ERRPC,E.ECPT,T.ECPT,BITCNT
8340 041476 002214 002326
8341 041502 001254 001116 002160 DT0026: .WORD $TESTN,$ERRPC,T.CS1,T.CS2,T.ER,T.DS,T.WC,T.DCYL,T.DA
8342 041510 002170 002174 002172
8343 041516 002162 002200 002166
8344 041524 001254 001116
8345 041530 001162 001164 001166 DT0037: .WORD $TESTN,$ERRPC
8346 041536 001254 001116 002254 DT0040: .WORD $REG0,$REG1,$REG2
8347 041544 002214 002252 002212 DT0041: .WORD $TESTN,$ERRPC,E.ECPT,T.ECPT,E.ECPS,T.ECPS,BITCNT
8348 041552 002326
8349 041554 001254 001116 002252 DT0043: .WORD $TESTN,$ERRPC,E.ECPS,T.ECPS,BITCNT
8350 041562 002212 002326
8351 041566 001254 001116 002234 DT0045: .WORD $TESTN,$ERRPC,E.ER,T.ER
8352 041574 002174
8353 041576 001254 001116 002224 DT0046: .WORD $TESTN,$ERRPC,E.BA,T.BA
8354 041604 002164
8355 041606 001254 001116 002222 DT0062: .WORD $TESTN,$ERRPC,E.WC,T.WC
8356 041614 002162
8357 041616 001254 001116 001162 DT0063: .WORD $TESTN,$ERRPC,$REG0,$REG3,$REG1,$REG4,$REG2
8358 041624 001170 001164 001172
8359 041632 001166

```

E13

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 160
 DZR6EA.P11 28-SEP-76 15:31 DATA FORMAT FOR PRINT OUT

SEQ 0150

				.SBTTL DATA FORMAT FOR PRINT OUT
8360				
8361				
8362	041634	000001		DF000: .WORD 1
8363	041636	002	000	.BYTE 2,0
8364	041640	000006		DF0004: .WORD 6,0
8365	041642	000	000	.BYTE 0,0
8366	041644	043040		.WORD 04000A
8367	041646	000	000	.BYTE 0,0
8368	041650	043056		.WORD 04000B
8369	041652	002	000	.BYTE 2,0
8370	041654	043122		.WORD 04004
8371	041656	000	000	.BYTE 0,0
8372	041660	043141		.WORD 04004A
8373	041662	002	000	.BYTE 2,0
8374	041664	043231		.WORD 04004E
8375	041666	003	000	.BYTE 3,0
8376	041670	000005		DF0005: .WORD 5,0
8377	041672	000	000	.BYTE 0,0
8378	041674	043040		.WORD 04000A
8379	041676	000	000	.BYTE 0,0
8380	041700	043056		.WORD 04000B
8381	041702	002	000	.BYTE 2,0
8382	041704	043122		.WORD 04004
8383	041706	000	000	.BYTE 0,0
8384	041710	043157		.WORD 04004B
8385	041712	002	000	.BYTE 2,0
8386	041714	000005		DF0006: .WORD 5,0
8387	041716	000	000	.BYTE 0,0
8388	041720	043040		.WORD 04000A
8389	041722	000	000	.BYTE 0,0
8390	041724	043056		.WORD 04000B
8391	041726	002	000	.BYTE 2,0
8392	041730	043122		.WORD 04004
8393	041732	000	000	.BYTE 0,0
8394	041734	043175		.WORD 04004C
8395	041736	002	000	.BYTE 2,0
8396	041740	000005		DF0007: .WORD 5,0
8397	041742	000	000	.BYTE 0,0
8398	041744	043040		.WORD 04000A
8399	041746	000	000	.BYTE 0,0
8400	041750	043056		.WORD 04000B
8401	041752	002	000	.BYTE 2,0
8402	041754	043122		.WORD 04004
8403	041756	000	000	.BYTE 0,0
8404	041760	043213		.WORD 04004D
8405	041762	002	000	.BYTE 2,0
8406	041764	000005		DF0010: .WORD 5,0
8407	041766	000	000	.BYTE 0,0
8408	041770	043040		.WORD 04000A
8409	041772	000	000	.BYTE 0,0
8410	041774	043056		.WORD 04000B
8411	041776	002	000	.BYTE 2,0
8412	042000	043325		.WORD 04010
8413	042002	000	000	.BYTE 0,0
8414	042004	043371		.WORD 04010A
8415	042006	005	000	.BYTE 5,0

F13

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 161
 DZR6EA.P11 28-SEP-76 15:31 DATA FORMAT FOR PRINT OUT

SEQ 0161

8416	042010	000006		DF0011: .WORD	6
8417	042012	000	000	.BYTE	0,0
8418	042014	043040		.WORD	DH000A
8419	042016	000	000	.BYTE	0,0
8420	042020	043056		.WORD	DH0008
8421	042022	002	000	.BYTE	2,0
8422	042024	043325		.WORD	DH010
8423	042026	000	000	.BYTE	0,0
8424	042030	043437		.WORD	DH010B
8425	042032	005	000	.BYTE	5,0
8426	042034	043231		.WORD	DH004E
8427	042036	003	000	.BYTE	3,0
8428	042040	000005		DF0014: .WORD	5
8429	042042	000	000	.BYTE	0,0
8430	042044	043040		.WORD	DH000A
8431	042046	000	000	.BYTE	0,0
8432	042050	043056		.WORD	DH0008
8433	042052	002	000	.BYTE	2,0
8434	042054	043505		.WORD	DH014
8435	042056	000	000	.BYTE	0,0
8436	042060	043544		.WORD	DH014A
8437	042062	004	000	.BYTE	4,0
8438	042064	000005		DF0017: .WORD	5
8439	042066	000	000	.BYTE	0,0
8440	042070	043040		.WORD	DH000A
8441	042072	000	000	.BYTE	0,0
8442	042074	043056		.WORD	DH0008
8443	042076	002	000	.BYTE	2,0
8444	042100	043601		.WORD	DH0017
8445	042102	000	000	.BYTE	0,0
8446	042104	043665		.WORD	DH017A
8447	042106	007	000	.BYTE	7,0
8448	042110	000005		DF0021: .WORD	5
8449	042112	000	000	.BYTE	0,0
8450	042114	043040		.WORD	DH000A
8451	042116	000	000	.BYTE	0,0
8452	042120	043056		.WORD	DH0008
8453	042122	002	000	.BYTE	2,0
8454	042124	043753		.WORD	DH0021
8455	042126	000	000	.BYTE	0,0
8456	042130	043777		.WORD	DH021A
8457	042132	003	000	.BYTE	3,0
8458	042134	000004		DF0026: .WORD	4
8459	042136	000	000	.BYTE	0,0
8460	042140	043040		.WORD	DH000A
8461	042142	000	000	.BYTE	0,0
8462	042144	043056		.WORD	DH0008
8463	042146	002	000	.BYTE	2,0
8464	042150	044053		.WORD	DH0026
8465	042152	007	000	.BYTE	7,0
8466	042154	000003		DF0027: .WORD	3
8467	042156	000	000	.BYTE	0,0
8468	042160	043056		.WORD	DH0008
8469	042162	002	000	.BYTE	2,0
8470	042164	044053		.WORD	DH0026
8471	042166	007	000	.BYTE	7,0

G13

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZRG6-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 162
 DZRG6A.P11 28-SEP-76 15:31 DATA FORMAT FOR PRINT OUT

SEQ 0152

8472	042170	000004		DF0037: .WORD	4
8473	042172	000	000	.BYTE	0,0
8474	042174	043056		.WORD	DH0008
8475	042176	002	000	.BYTE	2,0
8476	042200	044140		.WORD	DH037A
8477	042202	000	000	.BYTE	0,0
8478	042204	044165		.WORD	DH037B
8479	042206	003	000	.BYTE	3,0
8480	042210	000000		DF0040: .WORD	0
8481	042212	003	000	.BYTE	3,0
8482	042214	000004		DF0041: .WORD	4
8483	042216	000	000	.BYTE	0,0
8484	042220	043056		.WORD	DH0008
8485	042222	002	000	.BYTE	2,0
8486	042224	044214		.WORD	DH0041
8487	042226	000	000	.BYTE	0,0
8488	042230	044260		.WORD	DH041A
8489	042232	005	000	.BYTE	5,0
8490	042234	000004		DF0042: .WORD	4
8491	042236	000	000	.BYTE	0,0
8492	042240	043056		.WORD	DH0008
8493	042242	002	000	.BYTE	2,0
8494	042244	043753		.WORD	DH0021
8495	042246	000	000	.BYTE	0,0
8496	042250	043777		.WORD	DH021A
8497	042252	003	000	.BYTE	3,0
8498	042254	000004		DF0043: .WORD	4
8499	042256	000	000	.BYTE	0,0
8500	042260	043056		.WORD	DH0008
8501	042262	002	000	.BYTE	2,0
8502	042264	043753		.WORD	DH0021
8503	042266	000	000	.BYTE	0,0
8504	042270	044025		.WORD	DH021B
8505	042272	003	000	.BYTE	3,0
8506	042274	000005		DF0045: .WORD	5
8507	042276	000	000	.BYTE	0,0
8508	042300	043040		.WORD	DH000A
8509	042302	000	000	.BYTE	0,0
8510	042304	043056		.WORD	DH0008
8511	042306	002	000	.BYTE	2,0
8512	042310	043122		.WORD	DH004
8513	042312	000	000	.BYTE	0,0
8514	042314	043256		.WORD	DH004F
8515	042316	002	000	.BYTE	2,0
8516	042320	000004		DF0046: .WORD	4
8517	042322	000	000	.BYTE	0,0
8518	042324	043056		.WORD	DH0008
8519	042326	002	000	.BYTE	2,0
8520	042330	043122		.WORD	DH004
8521	042332	000	000	.BYTE	0,0
8522	042334	043273		.WORD	DH004G
8523	042336	002	000	.BYTE	2,0
8524	042340	000005		DF0050: .WORD	5
8525	042342	000	000	.BYTE	0,0
8526	042344	043040		.WORD	DH000A
8527	042346	000	000	.BYTE	0,0

H13

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 163
 DZR6EA.P11 28-SEP-76 15:31 DATA FORMAT FOR PRINT OUT

SEQ 0153

8528	042350	043056		.WORD	DH0008
8529	042352	002	000	.BYTE	2,0
8530	042354	043122		.WORD	DH004
8531	042356	000	000	.BYTE	0,0
8532	042360	043141		.WORD	DH004A
8533	042362	002	000	.BYTE	2,0
8534	042364	000004		DF0060: .WORD	4
8535	042366	000	000	.BYTE	0,0
8536	042370	043056		.WORD	DH0008
8537	042372	002	000	.BYTE	2,0
8538	042374	044326		.WORD	DH060A
8539	042376	000	000	.BYTE	0,0
8540	042400	044353		.WORD	DH0608
8541	042402	003	000	.BYTE	3,0
8542	042404	000005		DF0061: .WORD	5
8543	042406	000	000	.BYTE	0,0
8544	042410	043040		.WORD	DH000A
8545	042412	000	000	.BYTE	0,0
8546	042414	043056		.WORD	DH0008
8547	042416	002	000	.BYTE	2,0
8548	042420	043122		.WORD	DH004
8549	042422	000	000	.BYTE	0,0
8550	042424	043273		.WORD	DH004G
8551	042426	002	000	.BYTE	2,0
8552	042430	000005		DF0062: .WORD	5
8553	042432	000	000	.BYTE	0,0
8554	042434	043040		.WORD	DH000A
8555	042436	000	000	.BYTE	0,0
8556	042440	043056		.WORD	DH0008
8557	042442	002	000	.BYTE	2,0
8558	042444	043122		.WORD	DH004
8559	042446	000	000	.BYTE	0,0
8560	042450	043310		.WORD	DH004H
8561	042452	002	000	.BYTE	2,0
8562	042454	000005		DF0063: .WORD	5
8563	042456	000	000	.BYTE	0,0
8564	042460	043040		.WORD	DH000A
8565	042462	000	000	.BYTE	0,0
8566	042464	043056		.WORD	DH0008
8567	042466	002	000	.BYTE	2,0
8568	042470	044400		.WORD	DH063A
8569	042472	000	000	.BYTE	0,0
8570	042474	044445		.WORD	DH063B
8571	042476	005	000	.BYTE	5,0

.SBTTL ASCII MESSAGES

8572					
8573					
8574	042500	005015	045522	030466	OPR001: .ASCIIZ <15><12>/RK611 VECTOR ADDRESS (/
8575	042506	020061	042526	052103	
8576	042514	051117	040440	042104	
8577	042522	042522	051523	024040	
8578	042530	000040			
8579	042532	024440	036440	000040	OPR002: .ASCIIZ /) = /
8580	042540	045522	030466	020061	OPR003: .ASCIIZ /RK611 VECTOR ADDRESS (/
8581	042546	042526	052103	051117	
8582	042554	040440	042104	042522	
8583	042562	051523	024040	000040	
8584	042570	045522	030466	020061	OPR004: .ASCIIZ /RK611 PRIORITY (/
8585	042576	051120	047511	044522	
8586	042604	054524	024040	000040	
8587	042612	005015	047062	020104	OPR006: .ASCIIZ <15><12>/2ND PASS RUN TIME APPROX 4 MINUTES/<15><12>
8588	042620	040520	051523	051040	
8589	042626	047125	052040	046511	
8590	042634	020105	050101	051120	
8591	042642	054117	032040	046440	
8592	042650	047111	052125	051505	
8593	042656	005015	000		
8594	042661	015	025012	025052	OPR007: .ASCIIZ <15><12>/***** PROGRAM HALTED *****/<15><12>
8595	042666	025052	020040	050040	
8596	042674	047522	051107	046501	
8597	042702	044040	046101	042524	
8598	042710	020104	020040	025052	
8599	042716	025052	006452	000012	
8600	042724	020040	000		
8601	042727	015	050012	047522	SPACE2: .ASCIIZ / /
8602	042734	051107	046501	040440	ABORT: .ASCIIZ <15><12>/PROGRAM ABORTED BECAUSE ERROR THRESHOLD EXCEEDED/<15><12>
8603	042742	047502	052122	042105	
8604	042750	041040	041505	052501	
8605	042756	042523	042440	051122	
8606	042764	051117	052040	051110	
8607	042772	051505	047510	042114	
8608	043000	042440	041530	042505	
8609	043006	042504	006504	000012	
8610	043014	005015	042524	052123	TSTBY1: .ASCIIZ <15><12>/TEST /
8611	043022	000040			
8612	043024	041040	050131	051501	TSTBY2: .ASCIIZ / BYPASSED/<15><12>
8613	043032	042523	006504	000012	

J13

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 165
 DZR6EA.P11 28-SEP-76 15:31 DATA HEADERS

SEQ 0165

.SBTTL DATA HEADERS				
8614				
8615				
8616	043040	042524	052123	020040
8617	043046	020040	051105	047522
8618	043054	000122		
8619	043056	052516	020115	020040
8620	043064	020040	041520	000
8621	043071	124	051505	020124
8622	043076	020040	052040	040522
8623	043104	006520	012	
8624	043107	116	046525	020040
8625	043114	020040	050040	000103
8626	043122	054105	042520	052103
8627	043130	020040	041501	052524
8628	043136	046101	000	
8629	043141	122	041513	030523
8630	043146	020040	051040	041513
8631	043154	030523	000	
8632	043157	122	046513	030522
8633	043164	020040	051040	046513
8634	043172	030522	000	
8635	043175	122	046513	031122
8636	043202	020040	051040	046513
8637	043210	031122	000	
8638	043213	122	046513	031522
8639	043220	020040	051040	046513
8640	043226	031522	000	
8641	043231	122	041513	031123
8642	043236	020040	051040	042513
8643	043244	020122	020040	051040
8644	043252	042113	000123	
8645	043256	045522	051105	020040
8646	043264	020040	045522	051105
8647	043272	000		
8648	043273	122	041113	020101
8649	043300	020040	051040	041113
8650	043306	000101		
8651	043310	045522	041527	020040
8652	043316	020040	045522	041527
8653	043324	000		
8654	043325	105	050130	041505
8655	043332	020124	040440	052103
8656	043340	040525	020114	050040
8657	043346	042522	042523	052116
8658	043354	050040	042522	047526
8659	043362	051525	041040	052111
8660	043370	000		
8661	043371	122	046513	030522
8662	043376	020040	051040	046513
8663	043404	030522	020040	041040
8664	043412	052111	020040	020040
8665	043420	041040	052111	020040
8666	043426	020040	041440	052517
8667	043434	052116	000	
8668	043437	122	041513	030523
8669	043444	020040	051040	041513

.SBTTL DATA HEADERS				
DH000A:	.ASCIZ	/TEST	ERRCR/	
DH000B:	.ASCIZ	/NUM	PC/	
DH000C:	.ASCII	/TEST	TRAP/<15><12>	
DH004:	.ASCIZ	/EXPECT	ACTUAL/	
DH004A:	.ASCIZ	/RKCS1	RKCS1/	
DH004B:	.ASCIZ	/RKMR1	RKMR1/	
DH004C:	.ASCIZ	/RKMR2	RKMR2/	
DH004D:	.ASCIZ	/RKMR3	RKMR3/	
DH004E:	.ASCIZ	/RKCS2	RKER	RKDS/
DH004F:	.ASCIZ	/RKER	RKER/	
DH004G:	.ASCIZ	/RKBA	RKBA/	
DH004H:	.ASCIZ	/RKWC	RKWC/	
DH010:	.ASCIZ	/EXPECT	ACTUAL	PRESENT PREVIOUS BIT/
DH010A:	.ASCIZ	/RKMR1	RKMR1	BIT BIT COUNT/
DH010B:	.ASCIZ	/RKCS1	RKCS1	BIT BIT COUNT/

L13

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11: 27(1006) 05-007-76 09:12 PAGE 167
 DZR6EA.P11 28-SEP-76 15:31 DATA HEADERS

SEG 0157

8726	044140	047507	042117	020040	DH037A: .ASCIZ	GOOD	BAD	WORD/
8727	044146	020040	040502	020104				
8728	044154	020040	020040	047527				
8729	044162	042122	000					
8730	044165	127	051117	020104	DH037B: .ASCIZ	WORD	WORD	NUMBER/
8731	044172	020040	053440	051117				
8732	044200	020104	020040	047040				
8733	044206	046525	042502	000122				
8734	044214	054105	042520	052103	DH0041: .ASCIZ	/EXPECT	ACTUAL	EXPECT ACTUAL BIT/
8735	044222	020040	041501	052524				
8736	044230	046101	020040	054105				
8737	044236	042520	052103	020040				
8738	044244	041501	052524	046101				
8739	044252	020040	044502	000124				
8740	044260	045522	041505	052120	DH041A: .ASCIZ	/RKECPT	RKECPT	RKECPS RKECPS COUNT/
8741	044266	020040	045522	041505				
8742	044274	052120	020040	045522				
8743	044302	041505	051520	020040				
8744	044310	045522	041505	051520				
8745	044316	020040	047503	047125				
8746	044324	000124						
8747	044326	045522	051503	020062	DH060A: .ASCIZ	/RKCS2	BUFFER	WORD/
8748	044334	020040	052502	043106				
8749	044342	051105	020040	047527				
8750	044350	042122	000					
8751	044353	040	020040	020040	DH060B: .ASCIZ	/	WORD	READ/
8752	044360	020040	053440	051117				
8753	044366	020104	020040	051040				
8754	044374	040505	000104					
8755	044400	045522	051503	020062	DH063A: .ASCIZ	/RKCS2	BA17-16 BUFFER	BA17-16 WORD/
8756	044406	020040	040502	033461				
8757	044414	030455	020066	052502				
8758	044422	043106	051105	020040				
8759	044430	040502	033461	030455				
8760	044436	020066	047527	042122				
8761	044444	000						
8762	044445	040	020040	020040	DH063B: .ASCIZ	/	BUF WD	WORD WD READ READ/
8763	044452	020040	041040	043125				
8764	044460	053440	020104	053440				
8765	044466	051117	020104	020040				
8766	044474	053440	020104	042522				
8767	044502	042101	051040	040505				
8768	044510	000104						

M13

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 168
 DZR6EA.P11 28-SEP-76 15:31 ERROR MESSAGES

SEQ 0158

8769					.SBTTL ERROR MESSAGES
8770					
8771	044512	047125	054105	042520	EM000: .ASCIZ /UNEXPECTED MEMORY PARITY ENABLE TRAP/
8772	044520	052103	042105	046440	
8773	044526	046505	051117	020131	
8774	044534	040520	044522	054524	
8775	044542	042440	040516	046102	
8776	044550	020105	051124	050101	
8777	044556	000			
8778	044557	105	051122	051117	EM5004: .ASCIZ /ERROR AFTER ATTEMPTING IMPLIED SEEK/
8779	044564	040440	052106	051105	
8780	044572	040440	052124	046505	
8781	044600	052120	047111	020107	
8782	044606	046511	046120	042511	
8783	044614	020104	042523	045505	
8784	044622	000			
8785	044623	122	041513	030523	E5004A: .ASCIZ /RKCS1 IN ERROR/
8786	044630	044440	020116	051105	
8787	044636	047522	000122		
8788	044642	045522	051115	020061	E5004B: .ASCIZ /RKMR1 IN ERROR/
8789	044650	047111	042440	051122	
8790	044656	051117	000		
8791	044661	122	046513	031122	E5004C: .ASCIZ /RKMR2 IN ERROR/
8792	044666	044440	020116	051105	
8793	044674	047522	000122		
8794	044700	045522	051115	020063	E5004D: .ASCIZ /RKMR3 IN ERROR/
8795	044706	047111	042440	051122	
8796	044714	051117	000		
8797	044717	122	042513	020122	E5004E: .ASCIZ /RKER IN ERROR/
8798	044724	047111	042440	051122	
8799	044732	051117	000		
8800	044735	122	041113	020101	E5004F: .ASCIZ /RKBA IN ERROR/
8801	044742	047111	042440	051122	
8802	044750	051117	000		
8803	044753	122	053513	020103	E5004G: .ASCIZ /RKWC IN ERROR/
8804	044760	047111	042440	051122	
8805	044766	051117	000		
8806	044771	105	051122	051117	EM5010: .ASCIZ /ERROR ATTEMPTING TO READ HEADER SYNC (PREAMBLE)/
8807	044776	040440	052124	046505	
8808	045004	052120	047111	020107	
8809	045012	047524	051040	040505	
8810	045020	020104	042510	042101	
8811	045026	051105	051440	047131	
8812	045034	020103	050050	042522	
8813	045042	046501	046102	024505	
8814	045050	000			
8815	045051	105	051122	051117	EM5011: .ASCIZ &ERROR ATTEMPTING TO DO HEADER READ/COMPARE&
8816	045056	040440	052124	046505	
8817	045064	052120	047111	020107	
8818	045072	047524	042040	020117	
8819	045100	042510	042101	051105	
8820	045106	051040	040505	027504	
8821	045114	047503	050115	051101	
8822	045122	000105			
8823	045124	045522	041504	046131	E5011A: .ASCIZ /RKDCYL OR RKDA IN ERROR/
8824	045132	047440	020122	045522	

N13

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A
DZR6EA.P11 28-SEP-76 15:31

MACY11 27(1006) 05-OCT-76 09:12 PAGE 169

SEQ 0169

ERROR MESSAGES

8825	045140	040504	044440	020116	
8826	045146	051105	047522	000122	
8827	045154	051105	047522	020122	EM5015: .ASCIZ /ERROR ATTEMPTING TO READ GAP/
8828	045162	052101	042524	050115	
8829	045170	044524	043516	052040	
8830	045176	020117	042522	042101	
8831	045204	043440	050101	000	
8832	045211	105	051122	051117	EM5020: .ASCIZ /ERROR ATTEMPTING TO WRITE DATA SYNC (PREAMBLE)/
8833	045216	040440	052124	046505	
8834	045224	052120	047111	020107	
8835	045232	047524	053440	044522	
8836	045240	042524	042040	052101	
8837	045246	020101	054523	041516	
8838	045254	024040	051120	040505	
8839	045262	041115	042514	000051	
8840	045270	051105	047522	020122	EM5021: .ASCIZ /ERROR ATTEMPTING TO WRITE DATA OR ECC/
8841	045276	052101	042524	050115	
8842	045304	044524	043516	052040	
8843	045312	020117	051127	052111	
8844	045320	020105	040504	040524	
8845	045326	047440	020122	041505	
8846	045334	000103			
8847	045336	051105	047522	020122	ES021A: .ASCIZ /ERROR IN ECC GENERATION/
8848	045344	047111	042440	041503	
8849	045352	043440	047105	051105	
8850	045360	052101	047511	000116	
8851	045366	051105	047522	020122	EM5023: .ASCIZ /ERROR BETWEEN SECTORS ATTEMPTING MULTI-SECTOR WRITE/
8852	045374	042502	053524	042505	
8853	045402	020116	042523	052103	
8854	045410	051117	020123	052101	
8855	045416	042524	050115	044524	
8856	045424	043516	046440	046125	
8857	045432	044524	051455	041505	
8858	045440	047524	020122	051127	
8859	045446	052111	000105		
8860	045452	051115	020061	047111	EMW1: .ASCIZ /MR1 INCORRECT ON 1ST UPWARD TRANSITION OF MAINT CLOCK/
8861	045460	047503	051122	041505	
8862	045466	020124	047117	030440	
8863	045474	052123	052440	053520	
8864	045502	051101	020104	051124	
8865	045510	047101	044523	044524	
8866	045516	047117	047440	020106	
8867	045524	040515	047111	020124	
8868	045532	046103	041517	000113	
8869	045540	051115	020061	047111	EMW2: .ASCIZ /MR1 INCORRECT ON 1ST DOWNWARD TRANSITION OF MAINT CLOCK/
8870	045546	047503	051122	041505	
8871	045554	020124	047117	030440	
8872	045562	052123	042040	053517	
8873	045570	053516	051101	020104	
8874	045576	051124	047101	044523	
8875	045604	044524	047117	047440	
8876	045612	020106	040515	047111	
8877	045620	020124	046103	041517	
8878	045626	000113			
8879	045630	051115	020061	047111	EMW3: .ASCIZ /MR1 INCORRECT ON 2ND UPWARD TRANSITION OF MAINT CLOCK/
8880	045636	047503	051122	041505	

8881	045644	020124	047117	031040
8882	045652	042116	052440	053520
8883	045660	051101	020104	051124
8884	045666	047101	044523	044524
8885	045674	047117	047440	020106
8886	045702	040515	047111	020124
8887	045710	046103	041517	000113
8888	045716	051115	020061	047111
8889	045724	047503	051122	041505
8890	045732	020124	047117	031040
8891	045740	042116	042040	053517
8892	045746	053516	051101	020104
8893	045754	051124	047101	044523
8894	045762	044524	047117	047440
8895	045770	020106	040515	047111
8896	045776	020124	046103	041517
8897	046004	000113		
8898	046006	054105	042520	052103
8899	046014	042105	041440	046131
8900	046022	047111	042504	020122
8901	046030	053117	051105	046106
8902	046036	053517	042440	051122
8903	046044	042040	042111	047040
8904	046052	052117	051440	052105
8905	046060	000		
8906	046061	127	044522	042524
8907	046066	052040	020117	054503
8908	046074	020114	031466	026062
8909	046102	052040	040522	045503
8910	046110	031040	020054	042523
8911	046116	052103	051117	031040
8912	046124	026065	053440	051117
8913	046132	020104	047503	047125
8914	046140	020124	030064	000061
8915	046146	047125	054105	042520
8916	046154	052103	042105	042440
8917	046162	051122	051117	053440
8918	046170	044510	042514	052040
8919	046176	054522	047111	020107
8920	046204	047524	043040	051117
8921	046212	042503	041440	046131
8922	046220	047440	042526	043122
8923	046226	047514	000127	
8924	046232	051127	052111	020105
8925	046240	047524	041440	046131
8926	046246	030440	032464	026062
8927	046254	052040	040522	045503
8928	046262	031040	020054	042523
8929	046270	052103	051117	031040
8930	046276	026065	053440	051117
8931	046304	020104	047503	047125
8932	046312	020124	030064	000061
8933	046320	047125	054105	042520
8934	046326	052103	042105	041440
8935	046334	046131	047111	042504
8936	046342	020122	053117	051105

EMW4: .ASCIZ /MFI INCORRECT ON 2ND DOWNWARD TRANSITION OF MAINT CLOCK/

EM5026: .ASCIZ EXPECTED CYLINDER OVERFLOW ERR DID NOT SET/

E5026A: .ASCIZ /WRITE TO CYL 632, TRACK 2, SECTOR 25, WORD COUNT 401/

EM5027: .ASCIZ /UNEXPECTED ERROR WHILE TRYING TO FORCE CYL OVERFLOW/

E5026B: .ASCIZ /WRITE TO CYL 1452, TRACK 2, SECTOR 25, WORD COUNT 401/

EM5030: .ASCIZ /UNEXPECTED CYLINDER OVERFLOW WRITING LAST PHYSICAL ADDRESS/

C14

 RX611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZ6E-A
 DZ6EA.P11 28-SEP-76 15:31 ERROR MESSAGES

MACY11 27(1006) 05-OCT-76 09:12 PAGE 171

SEG 317:

8937	046350	046106	053517	053440
8938	046356	044522	044524	043516
8939	046364	046040	051501	020124
8940	046372	044120	051531	041511
8941	046400	046101	040440	042104
8942	046406	042522	051523	000
8943	046413	125	042516	050130
8944	046420	041505	042524	020104
8945	046426	051105	047522	020122
8946	046434	044127	046111	020105
8947	046442	051127	052111	047111
8948	046450	020107	040514	052123
8949	046456	050040	054510	044523
8950	046464	040503	020114	042101
8951	046472	051104	051505	000123
8952	046500	051105	047522	020122
8953	046506	052101	042524	050115
8954	046514	044524	043516	052040
8955	046522	020117	042522	042101
8956	046530	042040	052101	020101
8957	046536	054523	041516	024040
8958	046544	051120	040505	041115
8959	046552	042514	000051	
8960	046556	051105	047522	020122
8961	046564	052101	042524	050115
8962	046572	044524	043516	052040
8963	046600	020117	042522	042101
8964	046606	042040	052101	020101
8965	046614	051117	042440	041503
8966	046622	000		
8967	046623	105	041503	042440
8968	046630	051122	051117	044440
8969	046636	020116	042522	042101
8970	046644	042040	052101	020101
8971	046652	050117	051105	052101
8972	046660	047511	000116	
8973	046664	040504	040524	041440
8974	046672	046517	040520	042522
8975	046700	042440	051122	051117
8976	046706	040440	052106	051105
8977	046714	044440	050116	052125
8978	046722	047040	051120	052040
8979	046730	040522	051516	042506
8980	046736	000122		
8981	046740	041505	020103	042522
8982	046746	044507	052123	051105
8983	046754	047040	052117	041440
8984	046762	051117	042522	052103
8985	046770	040440	052106	051105
8986	046776	042440	041503	051040
8987	047004	040505	020104	047111
8988	047012	051040	040505	020104
8989	047020	040504	040524	000
8990	047025	105	051122	051117
8991	047032	044440	020116	041505
8992	047040	020103	040520	052124

EM5031: .ASCIZ /UNEXPECTED ERROR WHILE WRITING LAST PHYSICAL ADDRESS/

EM5032: .ASCIZ /ERROR ATTEMPTING TO READ DATA SYNC (PREAMBLE)/

EM5034: .ASCIZ /ERROR ATTEMPTING TO READ DATA OR ECC/

EM5035: .ASCIZ /ECC ERROR IN READ DATA OPERATION/

EM5037: .ASCIZ /DATA COMPARE ERROR AFTER INPUT NPR TRANSFER/

EM5041: .ASCIZ /ECC REGISTER NOT CORRECT AFTER ECC READ IN READ DATA/

EM5042: .ASCIZ /ERROR IN ECC PATTERN CALCULATION AFTER ECC ERROR/

D14

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A
 DZR6EA.P11 28-SEP-76 15:31 ERROR MESSAGES

MACY11 27(1006) 05-OCT-76 09:12 PAGE 172

SEG 0172

8993	047046	051105	020116	040503
8994	047054	041514	046125	052101
8995	047062	047511	020116	043101
8996	047070	042524	020122	041505
8997	047076	020103	051105	047522
8998	047104	000122		
8999	047106	051105	047522	020122
9000	047114	047111	042440	041503
9001	047122	050040	051517	052111
9002	047130	047511	020116	047503
9003	047136	047125	044524	043516
9004	047144	040440	052106	051105
9005	047152	042440	041503	042440
9006	047160	051122	051117	000
9007	047165	105	051122	051117
9008	047172	040440	052106	051105
9009	047200	050040	047522	042503
9010	047206	051523	047111	020107
9011	047214	040504	040524	041440
9012	047222	042510	045503	042440
9013	047230	051122	051117	000
9014	047235	124	040522	051516
9015	047242	042506	020122	042514
9016	047250	043516	044124	042440
9017	047256	051122	051117	040440
9018	047264	052124	046505	052120
9019	047272	047111	020107	040520
9020	047300	052122	040511	020114
9021	047306	042523	052103	051117
9022	047314	047440	042520	040522
9023	047322	044524	047117	000
9024	047327	102	042101	051440
9025	047334	041505	047524	020122
9026	047342	051105	047522	020122
9027	047350	044504	020104	047516
9028	047356	020124	051120	053105
9029	047364	047105	020124	042504
9030	047372	044523	042522	020104
9031	047400	042101	051104	051505
9032	047406	020123	047111	051103
9033	047414	046505	047105	000124
9034	047422	040502	020104	042523
9035	047430	052103	051117	042440
9036	047436	051122	051117	042040
9037	047444	042111	047040	052117
9038	047452	041440	052501	042523
9039	047460	041440	051105	020122
9040	047466	047524	051440	052105
9041	047474	000		
9042	047475	102	042101	051440
9043	047502	041505	047524	020122
9044	047510	051105	047522	020122
9045	047516	040506	046111	042105
9046	047524	052040	020117	042523
9047	047532	020124	044127	047105
9048	047540	042440	050130	041505

EM5043: .ASCIZ /ERROR IN ECC POSITION COUNTING AFTER ECC ERROR/

EM5044: .ASCIZ /ERROR AFTER PROCESSING DATA CHECK ERROR/

EM5046: .ASCIZ /TRANSFER LENGTH ERROR ATTEMPTING PARTIAL SECTOR OPERATION/

EM5047: .ASCIZ /BAD SECTOR ERROR DID NOT PREVENT DESIRED ADDRESS INCREMENT/

EM5050: .ASCIZ /BAD SECTOR ERROR DID NOT CAUSE CERR TO SET/

EM5051: .ASCIZ /BAD SECTOR ERROR FAILED TO SET WHEN EXPECTED/

E14

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 173
 DZR6EA.P11 28-SEP-76 15:31 ERROR MESSAGES

SEQ 0173

9049	047546	042524	000104		
9050	047552	040504	040524	052040	EM5052: .ASCII /DATA TRANSFER DID NOT CONTINUE TO END OF SECTOR/
9051	047560	040522	051516	042506	
9052	047566	020122	044504	020104	
9053	047574	047516	020124	047503	
9054	047602	052116	047111	042525	
9055	047610	052040	020117	047105	
9056	047616	020104	043117	051440	
9057	047624	041505	047524	122	
9058	047631	015	053412	042510	.ASCIIZ <15><12>/WHEN BAD SECTOR ERROR OCCURRED/
9059	047636	020116	040502	020104	
9060	047644	042523	052103	051117	
9061	047652	042440	051122	051117	
9062	047660	047440	041503	051125	
9063	047666	042522	000104		
9064	047672	051105	047522	020122	EM5053: .ASCIIZ /ERROR ATTEMPTING WRITE CHECK OR ECC READ AFTER WRT CHK/
9065	047700	052101	042524	050115	
9066	047706	044524	043516	053440	
9067	047714	044522	042524	041440	
9068	047722	042510	045503	047440	
9069	047730	020122	041505	020103	
9070	047736	042522	042101	040440	
9071	047744	052106	051105	053440	
9072	047752	052122	041440	045510	
9073	047760	000			
9074	047761	105	041503	042440	EM5054: .ASCIIZ /ECC ERROR IN WRITE CHECK OPERATION/
9075	047766	051122	051117	044440	
9076	047774	020116	051127	052111	
9077	050002	020105	044103	041505	
9078	050010	020113	050117	051105	
9079	050016	052101	047511	000116	
9080	050024	051127	052111	020105	EM5060: .ASCIIZ /WRITE CHECK ERROR FAILED TO SET IN 16 BIT MODE/
9081	050032	044103	041505	020113	
9082	050040	051105	047522	020122	
9083	050046	040506	046111	042105	
9084	050054	052040	020117	042523	
9085	050062	020124	047111	030440	
9086	050070	020066	044502	020124	
9087	050076	047515	042504	000	
9088	050103	124	040522	051516	EM5061: .ASCIIZ /TRANSFER LENGTH ERROR WITH WRITE CHECK ERROR/
9089	050110	042506	020122	042514	
9090	050116	043516	044124	042440	
9091	050124	051122	051117	053440	
9092	050132	052111	020110	051127	
9093	050140	052111	020105	044103	
9094	050146	041505	020113	051105	
9095	050154	047522	000122		
9096	050160	051127	052111	020105	EM5063: .ASCIIZ /WRITE CHECK ERROR FAILED TO SET IN 18 BIT MODE/
9097	050166	044103	041505	020113	
9098	050174	051105	047522	020122	
9099	050202	040506	046111	042105	
9100	050210	052040	020117	042523	
9101	050216	020124	047111	030440	
9102	050224	020070	044502	020124	
9103	050232	047515	042504	000	
9104					

F14

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 174
 DZR6EA.P11 28-SEP-76 15:31 ERROR MESSAGES

SEG 3174

9105	050240	.EVEN
9106		.SBTTL DATA PATTERNS AND BUFFERS
9107		:* PATTERN 1 - ALL ZEROS
9108		:* PATTERN 2 - ROTATING CELL PULSE PRECOMP
9109		
9110	050240	PAT2:
9111	050240	121105
9112	050242	150442
9113	050244	064221
9114	050246	132110
9115	050250	055044
9116	050252	026422
9117	050254	013211
9118	050256	105504
9119	050260	042642
9120	050262	021321
9121	050264	110550
9122	050266	044264
9123	050270	022132
9124	050272	011055
9125	050274	104426
9126	050276	042213
9127		
9128		:* PATTERN 3 - MAX PRECOMP PHASE MIX
9129		
9130	050300	PAT3:
9131	050300	133333
9132	050302	066666
9133	050304	155555
9134	050306	155555
9135	050310	133333
9136	050312	066666
9137	050314	066666
9138	050316	155555
9139	050320	155555
9140	050322	133333
9141	050324	133333
9142	050326	133333
9143	050330	133333
9144	050332	133333
9145	050334	133333
9146	050336	133333
9147		
9148		:* PATTERN 4 - HI-LO FREQ. MIX
9149		
9150	050340	PAT4:
9151	050340	177777
9152	050342	177777
9153	050344	177777
9154	050346	052525
9155	050350	052525
9156	050352	052525
9157	050354	177777
9158	050356	177777
9159	050360	052525
9160	050362	052525

G14

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 175
 DZR6EA.P11 28-SEP-76 15:31 DATA PATTERNS AND BUFFERS

SEQ 0175

9161 050364 177777
 9162 050366 052525
 9163 050370 177252
 9164 050372 177252
 9165 050374 172765
 9166 050376 172765

177777
 052525
 177252
 177252
 172765
 172765

;* PATTERN 5 - ALTERNATING 1'S AND 0'S

;* PATTERN 6 - COMPOSIT ROTATING

FAT6:

9172 050400
 9173 050400 072307
 9174 050402 135143
 9175 050404 156461
 9176 050406 167230
 9177 050410 073514
 9178 050412 035646
 9179 050414 016723
 9180 050416 107351
 9181 050420 143564
 9182 050422 061672
 9183 050424 030735
 9184 050426 114356
 9185 050430 046167
 9186 050432 123073
 9187 050434 151453
 9188 050436 164616

072307
 135143
 156461
 167230
 073514
 035646
 016723
 107351
 143564
 061672
 030735
 114356
 046167
 123073
 151453
 164616

;* PATTERN 7 - ALL ONES

9191
 9192 051440 000402
 9193 051444 000402
 9194 000001

IBUFF: .BLKW 402
 OBUFF: .BLKW 402
 .END

H14

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 177
 DZR6EA.P11 28-SEP-76 15:31 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0176

ABASE = 177440	1098#	1351	1392
ABORT = 042727	7633	8601#	
ACDW1 = 000000	1351		
ACDW2 = 000000	1351		
ACLO = 000010	1193#		
ACPUOP = 000000	1351	1366	
ADDW0 = 000000	1351		
ADDW1 = 000000	1351		
ADDW10 = 000000	1351		
ADDW11 = 000000	1351		
ADDW12 = 000000	1351		
ADDW13 = 000000	1351		
ADDW14 = 000000	1351		
ADDW15 = 000000	1351		
ADDW2 = 000000	1351		
ADDW3 = 000000	1351		
ADDW4 = 000000	1351		
ADDW5 = 000000	1351		
ADDW6 = 000000	1351		
ADDW7 = 000000	1351		
ADDW8 = 000000	1351		
ADDW9 = 000000	1351		
ADEVCT = 000000	1351	1357	
ADEVN = 000000	1351	1393	
AENV = 000000	1351	1362	
AENVN = 000000	1351	1363	
AFATAL = 000000	1351	1354	
AMADR1 = 000000	1351	1378	
AMADR2 = 000000	1351	1383	
AMADR3 = 000000	1351	1386	
AMADR4 = 000000	1351	1389	
AMAMS1 = 000000	1351	1373	
AMAMS2 = 000000	1351	1381	
AMAMS3 = 000000	1351	1384	
AMAMS4 = 000000	1351	1387	
AMSGAD = 000000	1351	1359	
AMSGLG = 000000	1351	1360	
AMSGTY = 000000	1351	1353	
AMTYP1 = 000000	1351	1374	
AMTYP2 = 000000	1351	1382	
AMTYP3 = 000000	1351	1385	
AMTYP4 = 000000	1351	1388	
APASS = 000000	1351	1356	
APRIOR = 000005	1097#	1351	
APTCU = 000040	7508#	7689	
APTEV = 000001	7464	7506#	7540 7682
APTSIZ = 000200	1850	7505#	
APTSP0 = 000100	7466	7507#	7684
ASWREG = 000000	1351	1364	
ATESTN = 000000	1351	1355	
AUNIT = 000000	1351	1358	
AUSWR = 000000	1351	1365	
AVECT1 = 120210	1096#	1351	1390
AVECT2 = 000000	1351	1391	
BADPAR = 002334	1784#	7231*	
BAI = 000020	1156#		

BA16 = 000400	1141*													
BA17 = 001000	1142*													
BITCNT 002326	1781*	3372*	3389*	3502*	3519*	3635*	3652*	3766*	3783*	3897*	3914*	4028*	4045*	
	4160*	4177*	4483*	4500*	4614*	4631*	5076*	5093*	5900*	5925*	5938*	5953*	5996*	
	6055*	6188*	6193*	6195*	6208*	6232*	6242*	6253*	6290*	6322*	6365*	6391*	6466*	
	6471*	6473*	6486*	6510*	6521*	6533*	6545*	6632*	6661	6669*	6688	6714	6722*	
	6723	6726	6739	6744	8327	8330	8336	8339	8346	8349				
BIT0 = 000001	1042*	1138	1171	1190	3728	3858	3989	4575	5351	5448	6027	6032	6304	
	6353	6379	6646	6702	6772	7210								
BIT00 = 000001	1032*	1042												
BIT01 = 000002	1031*	1041												
BIT02 = 000004	1030*	1040												
BIT03 = 000010	1029*	1039												
BIT04 = 000020	1028*	1038												
BIT05 = 000040	1027*	1037												
BIT06 = 000100	1026*	1036												
BIT07 = 000200	1025*	1035												
BIT08 = 000400	1024*	1034	7364											
BIT09 = 001000	1023*	1033	7382	7551										
BIT1 = 000002	1041*	1172	3859	4576	5802	6843								
BIT10 = 002000	1022*	1143	1162	1181	1214	1229	6095	6867	6869	7528				
BIT11 = 004000	1021*	1144	1163	1182	1199	1215	1230	3725	3855	3986	4572	7389		
BIT12 = 010000	1020*	1146	1164	1183	1216	3726	3856	3987	3990	4573	6106			
BIT13 = 020000	1019*	1147	1165	1184	1200	1217	6863	6865	7535					
BIT14 = 040000	1018*	1148	1166	1185	1201	1218	3726	3856	3987	4573	6006	6343	6679	
	7350													
BIT15 = 100000	1017*	1149	1150	1167	1186	1202	1219	4837	6006	6076	7211			
BIT2 = 000004	1040*	1173	1192											
BIT3 = 000010	1039*	1155	1174	1193										
BIT4 = 000020	1038*	1156	1175	1194	1208	1223	6863	6865						
BIT5 = 000040	1037*	1157	1176	1195	1209	1224								
BIT6 = 000100	1036*	1139	1158	1177	1196	1210	1225	3329	3331	3333	3459	3461	3463	
	3592	3594	3596	3726	3856	3987	4117	4119	4121	4440	4442	4444	4573	
	5032	5034	5036	5652	5654	5656	6100							
BIT7 = 000200	1035*	1140	1159	1178	1197	1211	1226							
BIT8 = 000400	1034*	1141	1160	1179	1198	1212	1227	6104	6867	6869				
BIT9 = 001000	1033*	1142	1161	1180	1213	1228	2023	2141	2258	2375	2492	2609	2729	
	3725	3855	3986	4572	5830	5842	5851	5859	5867	5872	5915	5962	6010	
	6041	6097	6152	6214	6263	6417	6427	6492	6555	6804	6821			
BLDDAT 033612	1969	2087	2204	2321	2438	2555	2675	2790	2873	2957	3040	3127	3234	
	3321	3451	3584	3717	3847	3978	4109	4240	4345	4432	4564	4696	4791	
	4925	5018	5154	5245	5638	7068*								
BPTVEC= 000014	1049*													
BSE = 000200	1178*	4859	4861											
BSEDES 002352	1791*	1959*	2077*	2194*	2311*	2428*	2545*	2662*	2779*	2862*	2947*	3029*	3112*	
	3219*	3307*	3437*	3570*	3702*	3833*	3964*	4095*	4225*	4330*	4418*	4550*	4681*	
	4778*	4837*	4853*	4911*	5006*	5141*	5232*	5337*	5434*	5527*	5627*	6016	6018	
	6019													
CCLR = 100000	1150*	1957	2075	2192	2309	2426	2543	2660	2780	2863	2945	3030	3113	
	3220	3308	3438	3571	3704	3834	3965	4096	4226	4331	4419	4551	4682	
	4776	4935	5039	5164	5255	5356	5453	5541	5659					
COT = 002000	1143*	2436	2553	2655	3038	6093	6131							
CERR = 100000	1149*	2833	2919	3000	3086	3409	3539	3672	3803	3934	4065	4197	4520	
	4651	4881	5113											
CFMT = 010000	1146*	2021	2139	2256	2373	2490	2607	2673	2727	4236	4341	4428	4560	
	5150	5535	5808	5828	6008	6090	6123	6328	6339	6636	6675			

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 179
DZR6EA.P11 28-SEP-76 15:31 CROSS REFERENCE TABLE -- USER SYMBOLS

CKSWR =	104407	7349	7524	755	8308#									
CLEAR =	000005	1123#												
CLRIBF	033566	3125	3232	3319	3449	3582	3715	3845	3976	4107	4238	4343	4430	4562
		4694	4789	4923	5016	5152	5243	5636	7057#					
COE =	001000	1180#	2836	2916	3002	3083								
CR =	000015	957#	7728	7738										
CRLF =	000200	958#	1875	7699	7738									
CTO =	004000	1144#												
CTRHLT	036364	7645#	7924	8007										
DCK =	100900	1186#	3417	3547	3680	3811	3942	4073	4205	4528	4659	5121		
DDISP =	177570	964#	1309	1838										
DDT =	000400	1198#												
DF000	041634	1415	8362#											
DF0004	041640	1430	1532	1622	1682	1688	8364#							
DF0005	041670	1436	8376#											
DF0006	041714	1442	1520	8386#										
DF0007	041740	1448	1526	8396#										
DF0010	041764	1454	1466	1484	1562	1574	1664	8406#						
DF0011	042010	1460	1472	1490	1502	1514	1568	1586	1676	8416#				
DF0014	042040	1478	1640	8428#										
DF0017	042064	1496	8438#											
DF0021	042110	1508	1580	1670	8448#									
DF0026	042134	1538	1544	1550	1556	8458#								
DF0027	042154	8466#												
DF0037	042170	1592	8472#											
DF0040	042210	1598	8480#											
DF0041	042214	1604	8482#											
DF0042	042234	1610	8490#											
DF0043	042254	1616	8498#											
DF0045	042274	1628	1652	8506#										
DF0046	042320	1634	1658	8516#										
DF0050	042340	1646	8524#											
DF0060	042364	1694	8534#											
DF0061	042404	1700	8542#											
DF0062	042430	1706	8552#											
DF0063	042454	1712	8562#											
DHDCMP	026042	1986	2052	2104	2170	2221	2287	2338	2404	2455	2521	2572	2638	2692
		2758	2807	2890	2974	3057	3144	3251	3349	3479	3612	3743	3874	4005
		4137	4257	4362	4460	4591	4713	4808	4845	4949	5053	5178	5269	5370
		5467	5560	5673	5991#									
DH000A	043040	1542	1590	1602	1608	1614	1632	1656	1692	8366	8378	8388	8398	8408
		8418	8430	8440	8450	8460								

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 180
 DZR6EA.P11 28-SEP-76 15:31 CROSS REFERENCE TABLE -- USER SYMBOLS

DH004H	043310	8560	8651#											
DH004I	044214	8486	8734#											
DH010	043325	8412	8422	8654#										
DH010A	043371	8414	8661#											
DH010B	043437	8424	8668#											
DH014	043505	8434	8675#											
DH014A	043544	8436	8681#											
DH017A	043665	8446	8695#											
DH021A	043777	8456	8496	8709#										
DH021B	044025	8504	8713#											
DH037A	044140	8476	8726#											
DH037B	044165	8478	8730#											
DH041A	044260	8488	8740#											
DH060A	044326	8538	8747#											
DH060B	044353	8540	8751#											
DH063A	044400	8568	8755#											
DH063B	044445	8570	8762#											
DI =	040000	1148#												
DI SEEK	025100	1974	2092	2209	2326	2443	2560	2680	2795	2878	2962	3045	3132	3239
		3337	3467	3600	3731	3862	3993	4125	4245	4350	4448	4579	4701	4796
		4937	5041	5166	5257	5358	5455	5548	5661	5799#				
DISPLA	001142	1309#	1838*	1846*	7404*	7527*								
DISPRE	000174	1238#	1846											
DLT =	100000	1167#												
DMD =	000040	1209#	2005	2006	2123	2124	2240	2241	2357	2358	2474	2475	2591	2592
		2711	2712	2827	2828	2910	2911	2994	2995	3077	3078	3276	3277	4387
		4388	4744	4745	4832	4833	4974	4975	5203	5204	5300	5301	5393	5394
		5490	5491	5583	5584	5697	5698	5777	5778	5813	5814	5835	5905	5930
		6025	6199	6237	6477	6511	6889	6919	6933	6948	6974	6986	6998	7026
		7029	7030	7031	7032	7035	7036	7037	7038	7145				
DRA =	000001	1190#												
DRDY =	000200	1197#												
DREAD	030774	3157	3264	3362	3492	3625	3756	3887	4018	4150	4270	4375	4473	4604
		4726	4821	6614#										
DRGPSN	030276	3149	3256	3354	3484	3617	3748	3879	4010	4142	4262	4367	4465	4596
		4718	4813	4954	5058	5183	5274	5375	5472	5565	5678	6458#		
DROT =	000040	1195#												
DRPAR =	000010	1174#												
DRSYNC	025504	1982	2048	2100	2166	2217	2283	2334	2400	2451	2517	2568	2634	2688
		2754	2803	2886	2970	3053	3140	3247	3345	3475	3608	3739	3870	4001
		4133	4253	43										

DT0010	041370	1453	1465	1483	1561	1573	1663	8327*						
DT0011	041406	1459	1471	1489	1501	1513	1567	1585	1675	8330*				
DT0014	041432	1477	1639	8334*										
DT0017	041446	1495	8336*											
DT0021	041470	1507	1579	1609	1669	8339*								
DT0026	041502	1537	1543	1549	1555	8341*								
DT0037	041524	1591	1693	8344*										
DT0040	041530	1597	8345*											
DT0041	041536	1603	8346*											
DT0043	041554	1615	8349*											
DT0045	041566	1627	1651	8351*										
DT0046	041576	1633	1657	1699	8353*									
DT0062	041606	1705	8355*											
DT0063	041616	1711	8357*											
DWGPSN	027002	1992	2110	2227	2344	2461	2578	2678	2813	2896	2980	3053	6180*	
DWRITE	027402	1999	2117	2234	2351	2468	2585	2705	2820	2903	2987	3070	6285*	
DWRTCK	030774	4962	5066	5191	5282	5383	5480	5573	5686	6613*				
ECCGEN	032260	3378	3508	3641	3772	3903	4034	4166	4489	4620	5082	6316	6663	6694
		6716	6841*											
ECCHI	002340	1786*	6292*	6352	6633*	6698	6849	6855	6860*	6868	6869*	6870*	6871	
ECCLO	002342	1787*	6291*	6380	6634*	6699	6859*	6864	6865*	6866*				
ECCSRC	002344	1788*	3115*	3221*	3222*	3328*	3458*	3591*	3724*	3854*	3985*	4116*	4228*	4332*
		4333*	4439*	4571*	4684*	4779*	4913*	4914*	5031*	5143*	5234*	5339*	5436*	5528*
		5651*	6696											
ECCW =	020000	1217*	5835	5905	5930	6025	6199	6237	6335	6373	6477	6511	6725	6728
ECCXOR	002336	1785*	6851*	6857*	6861									
ECH =	000100	1177*	3415	3545	3678	3809	3940	3942	4071	4073	4203	4526	4657	4659
		5119												
ECPATX	002346	1789*	3155*	3262*	3360*	3490*	3623*	3754*	3885*	4016*	4148*	4268*	4373*	4471*
		4602*	4724*	4819*	4960*	5064*	5189*	5280*	5381*	5478*	5571*	5684*	6741	
ECPOSX	002350	1790*	3156*	3263*	3361*	3381*	3382	3394*	3402*	3491*	3511*	3512	3524*	3532*
		3624*	3644*	3645	3657*	3665*	3755*	3775*	3776	3788*	3796*	3886*	3906*	3907
		3919*	3927*	4017*	4037*	4038	4050*	4058*	4149*	4169*	4170	4182*	4190*	4269*
		4374*	4472*	4492*	4493	4505*	4513*	4603*	4623*	4624	4636*	4644*	4725*	4820*
		4961*	5065*	5085*	5086	5098*	5106*	5190*	5281*	5332*	5479*	5572*	5685*	6733
		6746*	6749											
EH026	001602	1536*	2924*	3091*										
EH030	001622	1548*	2838*	3005*										
EH031	001632	1554*	2842*	3009*										
EMTVEC=	000030	1052*	1821*	1822*										
EMW	001510	1493*	1991*	1998*	2109*	2116*	2226*	2233*	2343*	2350*	2460*	2467*	2577*	2584*
		2697*	2704*	2812*	2819*	2895*	2902*	2979*	2986*	3062*	3069*	6894*	6924*	6938*
		6953*	6979*	6991*	7003*									
		6894	8860*											
EMW1	045452	6924	6979	8869*										
EMW2	045540	6938	6991	8879*										
EMW3	045630	6953	7003	8888*										
EMW4	045716	1412	8771*											
EM000	044512	1427	1433	1439	1445	1631	8778*							
EM5004	044557	1451	1457	1487	8806*									
EM5010	044771	1463	1469	1475	8815*									
EM5011	045051	1481	8827*											
EM5015	045154	1499	1991	2109	2226	2343	2460	2577	2697	2812	2895	2979	3062	8832*
EM5020	045211	1505	1511	1998	2116	2233	2350	2467	2584	2704	2819	2902	2986	3069
EM5021	045270	8840*												
		1517	1523	1529	8851*									
EM5023	045366													

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 182
 DZR6EA.P11 28-SEP-76 15:31 CROSS REFERENCE TABLE -- USER SYMBOLS

EMSO26	046006	1535	8898#											
EMSO27	046146	1541	8915#											
EMSO30	046320	1547	8933#											
EMSO31	046413	1553	8943#											
EMSO32	046500	1559	1565	8952#										
EMSO34	046556	1571	1583	1679	8960#									
EMSO35	046623	1577	8967#											
EMSO37	046664	1589	8973#											
EMSO41	046740	1601	8981#											
EMSO42	047025	1607	8990#											
EMSO43	047106	1613	8999#											
EMSO44	047165	1619	1625	9007#										
EMSO46	047235	9014#												
EMSO47	047327	1637	9024#											
EMSO50	047422	1643	9034#											
EMSO51	047475	1649	9042#											
EMSO52	047552	1655	9050#											
EMSO53	047672	1661	1673	1685	9064#									
EMSO54	047761	1667	9074#											
EMSO60	050024	1691	9080#											
EMSO61	050103	1697	1703	9088#										
EMSO63	050160	1709	9096#											
ERRCNT	002314	1776#	1854*	7628*	7631									
ERRVEC=	000004	1045#	1835	1836*	1847*	7229*	7230*	7250*	7251*	7277	7278	7282*	7287*	7299*
		7304*	7313*	7323*	7324*	7355	7356*	7358*	7361*					
E.ASOF	002236	1741#												
E.BA	002224	1736#	4740*	4887*	4888	5296*	5711*	5712*	5713	8353				
E.CS1	002220	1734#	2030*	2031	2148*	2149	2265*	2266	2382*	2383	2499*	2500	2616*	2617
		2736*	2737	3283*	3284*	3285	3408*	3409*	3410*	3411	3538*	3539*	3540*	3541
		3671*	3672*	3673*	3674	3802*	3803*	3804*	3805	3933*	3934*	3935*	3936	4064*
		4065*	4066*	4067	4196*	4197*	4198*	4199	4393*	4394*	4395*	4396	4519*	4520*
		4521*	4522	4650*	4651*	4652*	4653	4750*	4751*	4752*	4753	4866*	4867	4880*
		4881*	4882	4980*	4981*	4982*	4983	5112*	5113*	5114*	5115	5209*	5210*	5211*
		5212	5306*	5307*	5308*	5309	5834*	5838	5955*	5956	6114*	6115	6226*	6227
		6257	6398*	6400	6504*	6505	6549	6777*	6778	8318	8330			
E.CS2	002230	1738#												
E.DA	002226	1737#	4855*	6120*	6121*	6126	6128*	6129*	6134	6137*	6142	6147	8334	
E.DB	002242	1743#												
E.DCYL	002240	1742#	4854*	6119*	6138*	6140	6146	8334						
E.OS	002232	1739#												
E.ECPS	002252	1747#	3382*	3386										

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 183
DZR6EA.P11 28-SEP-76 15:31 CROSS REFERENCE TABLE -- USER SYMBOLS

E.WC	002222	1735#	5718*	5719	8355									
ES004A	044623	1428	1458	1470	1488	1500	1512	1530	1566	1584	1620	1644	1674	1680
		1686	8785#											
ES004B	044642	1434	1452	1464	1482	1560	1572	1662	8788#					
ES004C	044661	1440	1518	8791#										
ES004D	044700	1446	1524	8794#										
ES004E	044717	1626	1650	8797#										
ES004F	044735	1698	8800#											
ES004G	044753	1704	1710	8803#										
ES011A	045124	1476	1638	8823#										
ES021A	045336	1506	1578	1668	8847#									
ES026A	046061	2838	2842	2924	8906#									
ES026B	046232	3005	3009	3091	8924#									
FMTE =	000020	1175#												
FSBLVV	034242	2011	2025	2129	2143	2246	2260	2363	2377	2480	2494	2597	2611	2717
		2731	5819	5831	7205#									
GETREG	034154	2832	2915	2999	3082	3282	3407	3537	3670	3801	3932	4063	4195	4392
		4518	4649	4749	4865	4879	4979	5111	5208	5305	5398	5495	5588	5702
		5836	5952	6112	6224	6255	6395	6502	6547	6776	7169#			
GNS =	***** L	1237	1874	5748	5755	8300	8301	8302	8303	8304	8306	8308	8309	8310
		8311	8312	8313	8314									
GC =	000001	1138#	3283	3410	3540	3673	3804	3935	4066	4198	4394	4521	4652	4751
		4880	4981	5114	5210	5307								
GTSWR =	104406	1869	8306#											
HIBITS	002354	1792#	3116#	3223*	3310*	3440*	3573*	3706*	3836*	3967*	4098*	4229*	4334*	4421*
		4553*	4685*	4780*	4910*	5007*	5142*	5233*	5338*	5435*	5537*	5595	5604	5606*
		5626*	6680											
HT =	000011	955#	7697	7738										
HVRC =	000400	1179#												
IBUFF	050440	3121	3168	3228	3315	3445	3578	3711	3841	3972	4103	4234	4281	4339
		4426	4558	4690	4737	4740	4785	4887	4919	4929	5012	5022	5148	5158
		5239	5249	5293	5296	5344	5350*	5351	5402	5408*	5441	5447*	5448	5499
		5505*	5506*	5533	5538*	5592	5632	5642	5706	5711	7057	9192#		
IDAE =	002000	1181#												
IE =	000100	1139#												
ILF =	000001	1171#												
INTR =	000300	1134#												
IOTVEC=	000020	1050#	1819*	1820*										
IR =	000100	1158#												
KIPAR0=	172340	1083#	7263	7291										

AK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZ96E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 184
DZ96EA.P11 28-SEP-76 15:31 CROSS REFERENCE TABLE -- USER SYMBOLS

LOADRK	C34026	1961	2079	2196	2313	2430	2547	2667	2782	2865	2949	3032	3117	3224
		3311	3441	3574	3707	3837	3968	4099	4230	4335	4422	4554	4686	4781
		4915	5008	5144	5235	5340	5437	5529	5628	5711				
L.ASOF	002272	1759#	7146*	7161										
L.BA	002264	1754#	7143*	7156										
L.CS1	002260	1752#	2021	2030	2139	2148	2256	2265	2373	2382	2490	2499	2607	2616
		2727	2736	3408	3538	3671	3802	3933	4064	4196	4393	4519	4650	4750
		4866	4980	5112	5209	5306	5802	5808	5828	5834	5955	6008	6090	6093
		6114	6123	6131	6226	6328	6339	6398	6504	6636	6675	6777	6843	7148*
		7162												
L.CS2	002270	1758#	7147*	7159										
L.DA	002266	1755#	4855	6120	6147*	6185	6463	7142*	7158					
L.DCXL	002274	1760#	2010	2128	2245	2362	2479	2596	2716	4854	5818	6007	6119	6146*
		7141*	7157											
L.DS	002266	1756#	2019	2137	2254	2371	2488	2605	2725	5827	6005	6099	6102	6190
		6468												
L.DT	002267	1757#	2013	2131	2248	2365	2482	2599	2719	5821	5998			
L.MR1	002276	1761#	7145*	7160										
L.MR2	002300	1762#												
L.MR3	002302	1763#												
L.WC	002262	1753#	7144*	7155										
MLK =	000400	1212#	2005	2123	2240	2357	2474	2591	2711	2827	2910	2994	3077	3276
		4387	4744	4832	4974	5203	5300	5393	5490	5583	5697	5813	6888	6889
		6902	6932	6933	6947	6970	6985	6986	6997	7026	7029	7031	7035	7037
MDS =	001000	1161#												
MEMBAS=	172100	1093#	7233											
MEMERR	033524	1256	7045#											
MEMPAR	002332	1783#	7232*	7238*	7252									
MEMVEC=	000114	1092#	1255	7254*	7255*									
MERO =	001000	1213#	7029	7037										
MESMSK=	000017	1206#												
MEMD =	002000	1214#	5835	5905	5930	6025	6199	6237	6477	6511	6898	6903	6908	6932
		6946	6997											
MINC =	000200	1211#												
MEMVEC =	000250	1061#												
MSP =	000100	1210#	5777											
M1.BIT	002322	1779#	4874*	5904*	5942*	6028*	6201*	6233*	6307	6308*	6355	6356*	6385	6396*
		6479*	6649*	6693*	6704*	6845	6906	6965	6971	7024	8327	8330	8336	
M2.BIT														

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZRG6-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 185
DZRG6A.P11 28-SEP-76 15:31 CROSS REFERENCE TABLE -- USER SYMBOLS

OPR003	042540	1891	8580*											
OPR004	042570	1902	8584*											
OPR006	042612	1878	8587*											
OPR007	042661	7650	8594*											
JPSTRT	034072	1972	2090	2207	2324	2441	2558	2678	2793	2876	2960	3043	3131	3238
		3336	3466	3599	3730	3861	3992	4124	4244	4349	4447	4579	4700	4795
		4936	5040	5165	5256	5357	5454	5547	5660	7155*				
JR =	000200	1159#												
PACK =	000003	1122#												
PARCHA	034314	1933	7229*	8261										
FARM	002360	1244	1796#											
PAP.EN=	000001	1095#	7236											
PAT =	000020	1208#												
PAT2	050240	7107	9110#											
PAT3	050300	7105	9130#											
PAT4	050340	7101	9150#											
PAT6	050400	7090	9172#											
PCA =	004000	1215#	6902	6918	6967									
PCD =	010000	1216#	6902	6913	6973									
PGE =	002000	1162#												
PIP =	020000	1200#												
PIRG =	177772	962#												
PIRGVE=	0002+0	1056#												
PR.BIT	002320	1778#	4873*	5903*	5942	5943*	6028	6029*	6032*	6200*	6236*	6308	6309*	6356
		6357*	6396	6387*	6478*	6537*	6649	6650*	6653*	6693	6704	6705*	6708*	6904
		7022	8327	8330	8336									
		979#	1934											
PRO =	000000	980#												
PR1 =	000040	981#												
PR2 =	000100	982#												
PR3 =	000140	983#												
PR4 =	000200	984#	1770											
PR5 =	000240	985#												
PR6 =	000300	986#	1257	1806	7230	8244	8258							
PR7 =	000340	959#	960											
PS =	177776	960#												
PSW =	177776	1051#	1825*	1826*	8243*	8244*	8257*	8258*						
PWRVEC=	030024	1777#	6235*	6250*	6309	6310*	6313*	6357	6358*	6361*	6387	6388*	6842	6909
PL.BIT	002316	6963	8336											
		3377	3406	3507	3536	3640	3669	3771	3800	3902	3931	4033	4062	4165
		4194	4488	4517	4619	4648	4875	5081	5110	5907	5932	5944	6033	6202
		6480	6514	6526	6538	6655	6709	7022#						
		8083	8309#											
ROCHR =	104410	1129#	3123	3230	3317	3447	3580	3713	3843	3974	4105	4236	4341	4428
RODATA=	000021	4560	4692	4787										

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZ6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 186
 DZ6EA.P11 28-SEP-76 15:31 CROSS REFERENCE TABLE -- USER SYMBOLS

RKBA = 000004
RKCS1 = 000000

[illegible]

```

RKCS2 = 000010
RKDA  = 000006
RKDB  = 000024
RKDCYL= 000020
RKDS  = 000012
RKECPS= 000030
RKECPT= 000032

```

[illegible]

REF = 000014
RAIR = 000026

1108*	4858											
1112*	2005*	2006*	2123*	2124*	2240*	2241*	2357*	2358*	2474*	2475*	2591*	2592*
2711*	2712*	2827*	2828*	2910*	2911*	2994*	2995*	3077*	3078*	3276*	3277*	4387*
4388*	4744*	4745*	4832*	4833*	4974*	4975*	5203*	5204*	5300*	5301*	5393*	5394*
5490*	5491*	5583*	5584*	5697*	5698*	5777*	5778*	5813*	5814*	5908	5933	5945
6034	6203	6481	6515	6527	6539	6656	6710	6889*	6890	6919*	6920	6933*
6934	6948*	6949	6974*	6975	6986*	6987	6998*	6999	7026*	7029*	7030*	7031*

RKMR2 = 000034
RKMR3 = 000036

1113#	2027	2145	2262	2379	2496	2613	2733
1114#	2028	2146	2263	2380	2497	2614	2734

RR PRI 002306
RR SPAR= 000022

1770
1117

RKVEC 002304
RKWC = 000002

1769*	1926*	1927*
1103*	7155*	

```

RLS      = 000010
SAVREG= 104413

```

1155#
7574 8312#
1222# 2242# 2262#

002356
000040
104415

1793#	8242*	8262
1157#		
5406	5502	5602

[illegible]

5406 5503 5602 8314#
7446# 8314
1138#

```
SEER      = 000017
SELDREV= 000001
SKI       = 000002
```

1128#
1121#
1172#

```
SK1      - 00000E
SPACE2   042724
SPAR     = 020000
```

7610 7622 8600#

SPDLSS= 000020
SRTFLG 002312

1194* 1772* 1796* 1799* 1802* 1879

SPTSPL = 000011
SRQ = 177572

1125*	7288	7303*	7310*
-------	------	-------	-------

SR1 = 177574
SR2 = 177576
SR3 = 177577

1066#
1067#

```
SR3      = 172516
STACK    = 001100
START     002400
```

1068*	7300*								
950*	1805	1817	5881	7013	7053	7638	7649	8259	
1241	1822*								

START 002400
START1 002404
ETK1 MT- 133324

1241 1802#
1797 1800 1803# 7657
861#

```
SYSLINK= 1777774
SVAR    = 100000
SVR     001140
```

961#												
1202#												
1208#	1915	1927*	1939	1945*	1952*	1967	5942	5951	5959	5967	5972	5915

301146

1308#	1815	1837#	1839	1845#	1852#	1867	5842	5851	5859	5867	5872	5915
5962	6041	6152	6214	6263	6417	6427	6492	6555	6804	6821	6898	6928
6942	6957	6983	6995	7007	7049	7350	7364	7367	7382	7389	7446	7528

SWREG 000176

8342	8357	8383	8435	7007	7049	7333	7384	7387	7382	7389	7448	7528
7535	7547	7551	7629	7927	7966	8021*	8242	8262*				
1239#	1845	1867	7927	7966	7989							

SWC = 000001

12348
10148

E15

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 187
 DZR6EA.P11 28-SEP-76 15:31 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0196

SW00 = 000001	1004#	1014							
SW01 = 000002	1003#	1013							
SW02 = 000004	1002#	1012							
SW03 = 000010	1001#	1011							
SW04 = 000020	1000#	1010							
SW05 = 000040	999#	1009							
SW06 = 000100	998#	1008							
SW07 = 000200	997#	1007							
SW08 = 000400	996#	1006							
SW09 = 001000	995#	1005							
SW1 = 000002	1013#								
SW10 = 002000	994#								
SW11 = 004000	993#								
SW12 = 010000	992#	7629							
SW13 = 020000	991#								
SW14 = 040000	990#								
SW15 = 100000	989#								
SW2 = 000004	1012#								
SW3 = 000010	1011#								
SW4 = 000020	1010#								
SW5 = 000040	1009#								
SW6 = 000100	1008#								
SW7 = 000200	1007#								
SW8 = 000400	1006#								
SW9 = 001000	1005#	6898	6928	6942	6957	6983	6995	7007	7049
S.CLR = 000400	1227#								7446
S.FMT = 001000	1228#								
S.PACK = 004000	1230#								
S.RECL = 000040	1224#								
S.RTC = 000200	1226#								
S.SEEK = 000020	1223#								
S.STSP = 000100	1225#								
S.UNLD = 002000	1229#								
TBITVE = 000014	1047#								
TKVEC = 000060	1054#	7904*	7905*						
TPVEC = 000064	1055#								
TRAPPC = 002310	1771#	7046*	8317						
TRAPVE = 000034	1053#	1823*	1824*						
TRTVEC = 000014	1048#								
TSTBY1 043014	8610#								
TSTBY2 043024	8612#								
TST1 002424	1935	1954#	7409						
TST10 007142	2740	2745	2750	2776#	7416				
TST11 007412	2859#	7417							
TST12 007654	2942#	7418							
TST13 010124	3026#	7419							
TST14 010366	3109#	7420							
TST15 010702	3216#	7421							
TST16 011117	3304#	7422							
TST17 011722	3435#	7423							
TST2 004036	2034	2039	2044	2072#	7410				
TST20 012450	3567#	7424							
TST21 013176	3699#	7425							
TST22 013724	3830#	7426							
TST23 014460	3961#	7427							
TST24 015214	4092#	7428							

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 188
 DZR6EA.P11 28-SEP-76 15:31 CROSS REFERENCE TABLE -- USER SYMBOLS

[illegible]

RK611 DISKLESS CONTROLLER DIAGNOSTIC: PS MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 189
 DZR6EA.P11 28-SEP-76 15:31 CROSS REFERENCE TABLE -- USER SYMBOLS

[illegible]

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 190
DZR6EA.P11 28-SEP-76 15:31 CROSS REFERENCE TABLE -- USER SYMBOLS

SERMAX	001115	1297#	1830*	7380	7403*	7408								
SERROR	035666	1821	7523#											
SERRPC	001116	1298#	7532*	7533*	7534	7563	8316	8321	8323	8325	8327	8330	8334	8336
		8339	8341	8344	8346	8349	8351	8353	8355	8357				
SERRTB	001330	1410#	7583											
SERTTL	001112	1295#	5757	5761*	7531*	7563								
SESCAP	001236	1342#	1829*	1932*	3171*	4284*	5840*	5853*	5861*	5869*	5875*	5880*	5901*	5912*
		5914*	5919*	5920*	5936*	5949*	5964*	6037*	6040*	6045*	6046*	6154*	6206*	6213*
		6218*	6219*	6265*	6406*	6420*	6426*	6431*	6432*	6484*	6491*	6496*	6497*	6519*
		6530*	6542*	6557*	6784*	6788*	6792*	6796*	6800*	6808*	6820*	6826*	6828*	6893*
		6923*	6937*	6952*	6959*	6978*	6990*	7002*	7009*	7045*	7048*	7402*	7554	7556
		7563	7647*											
SETABL	001270	1361#												
SETEND	001330	1280	1394#											
SFATAL	001252	1354#	7492*											
SFFLG	035664	7455*	7458*	7486	7495*	7503#								
SFILLC	001156	1316#	7707	7738										
SFILLS	001155	1315#	7738											
SGOADR	001120	1299#												
SGDDAT	001124	1301#												
SGET42	025032	5762#												
SGTSWR	037766	7988#	8306											
SHO =	000000	935												
SHIBTS	001000	1275#												
SHIOCT	040774	8179*	8190#											
SICNT	001104	1292#	7393*	7394	7396*	7407								
SINTAG	001135	1306#	7985*	8004	8024	8137								
SITEMB	001114	1296#	7534*	7542	7563	7577								
SKTNEX	034650	7287	7312#											
SKTOUT	034640	7304	7309#											
SKT11	034506	7285#	7289*	7312*										
SLF	001246	1346#	7563	7738	8120	8130	8191							
SLFLG	035663	7496*	7502#											
SLPADR	001106	1293#	1831*	7375*	7384*	7400*	7405	7407	8264					
SLPERR	001110	1294#	1832*	5354*	5451*	5540*	5882	7014	7054	7384	7401*	7407	7450	7553
SLSTAD	034752	7325*	7332#											
SLSTBK	034754	7326*	7333#											
SMADR1	001302	1379#												
SMADR2	001306	1383#												
SMADR3	001312	1386#												
SMADR4	001316	1389#												
SMAIL	001250	1276	1280	1352#	1849	1865	7399	7540	7682					
SMAMS1	001300													

```

SMTYP4      001315
SMXCNT      035266
$NULL       001154
$NWTST=     000001

```

[illegible]

J15

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 192
DZR6EA.P11 28-SEP-76 15:31 CROSS REFERENCE TABLE -- USER SYMBOLS

SEC 0191

		8323	8325	8327	8330	8334	8336	8339	8341	8344	8346	8349	8351	8353
STIMES	001234	8323	8325	8327	8330	8334	8336	8339	8341	8344	8346	8349	8351	8353
		1341*	1828*	1955*	2073*	2190*	2307*	2424*	2541*	2658*	2777*	2860*	2943*	3027*
		3110*	3217*	3305*	3436*	3568*	3700*	3831*	3962*	4093*	4223*	4328*	4416*	4548*
		4679*	4774*	4909*	5004*	5139*	5230*	5335*	5432*	5525*	5624*	5737*	7387*	7394
		7397*	7407											
STKB	001146	1311*	7885	7906	7917	7942	7970	7997						
STKCN	037400	7886*	7901*	7931	7948*	8062	8064*							
STKINT	037410	1810	7901*	7922	7983									
STKQEN=	037407	7890*	7956	8067										
STKQIN	037402	7887*	7902*	7903	7954*	7955*	7956	7958*						
STKQOU	037404	7888*	7903*	8065	8066*	8067	8069*							
STKQSR	037406	7889*	7902	7958	8069									
STKS	001144	1310*	7885	7907*	7938*	7940	7946*	7968	7984*	7994	8006*	8026*		
STKSRV	037460	7904	7917*											
STMPC	001202	1328*	1886*	1888	1890	1896*	1898	1901	1913*	1915	1917	1919*	1920*	1921*
		1922*	1923*	1925										
STMP1	001204	1329*	6024*	6061	6064*	6067	6070*	6073	6077*	6080	6107*			
STMP10	001222	1336*	6628*	6795										
STMP11	001224	1337*	6630*	6799	6803									
STMP12	001226	1338*	6638*	6642*	6688	6714								
STMP13	001230	1339*	6639*	6643*	6723									
STMP14	001232	1340*	6640*	6644*	6726	6739	6744							
STMP2	001206	1330*												
STMP3	001210	1331*	6624*	6819										
STMP4	001212	1332*	6626*	6817										
STMP5	001214	1333*	3375*	3398	3403*	3505*	3528	3533*	3638*	3661	3666*	3769*	3792	3797*
		3900*	3923	3928*	4031*	4054	4059*	4163*	4186	4191*	4486*	4509	4514*	4617*
		4640	4645*	5079*	5102	5107*	6303*	6330	6341	6345*	6348*	6351*	6369	6376
		6378*	6635*	6677	6681*	6683*	6695*	6762	6766*	6769	6773*			
STMP6	001216	1334*	6295*	6409	6412	6415	6620*	6783	6791					
STMP7	001220	1335*	3172*	3191	3193*	4285*	4304	4306*	6297*	6396	6405	6622*	6787	

K15

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 193
DZR6EA.P11 28-SEP-76 15:31 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0192

[illegible]

M15

RK611 DISKLESS COM* PULLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 196
 DZR6EA.P11 28-SEP-76 15:31 CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0194

GETSWR	1057*	1862*													
LDLPER	1259*														
MSG	1939*	1941	2057*	2059	2175*	2177	2292*	2294	2409*	2411	2526*	2528	2643*	2645	2763*
	2765	2847*	2849	2929*	2931	3014*	3016	3097*	3099	3204*	3206	3289*	3291	3421*	3423
	3552*	3554	3685*	3687	3816*	3818	3947*	3949	4078*	4080	4210*	4212	4315*	4317	4400*
	4402	4533*	4535	4666*	4668	4759*	4761	4896*	4898	4988*	4990	5126*	5128	5216*	5218
	5315*	5317	5412*	5414	5510*	5512	5610*	5612							
MULT	1057*														
NEWTST	1057*	1939	2057	2175	2292	2409	2526	2643	2763	2847	2929	3014	3097	3204	3289
	3421	3552	3685	3816	3947	4078	4210	4315	4400	4533	4666	4759	4896	4988	5126
	5216	5315	5412	5510	5610										
OSECRC	1765*	3115	3222	3310	3440	3573	3706	3836	3967	4098	4228	4333	4421	4553	4684
	4779														
OSECWC	1766*	4914	5008	5143	5234	5339	5436	5628							
POP	1057*	5876	5921	6048	6220	6433	6498	6810	7118	7187	7221	7498	7499	7868	8180
	8229														
PUSH	1057*	5799	5898	5991	6180	6285	6458	6614	7068	7169	7205	7459	7461	7482	7827
	8154	8209													
REPORT	1057*														
SCOPE	952*	1954	2072	2189	2306	2423	2540	2657	2776	2859	2942	3026	3109	3216	3304
	3435	3567	3699	3830	3961	4092	4222	4327	4415	4547	4678	4773	4908	5003	5138
	5229	5334	5431	5524	5623	5735									
SETPRI	1057*	8058													
SETTRA	8292*	8301	8302	8303	8304	8306	8308	8309	8310	8311	8312	8313	8314		
SETUP	1057*	1811													
SKIP	1057*	2034	2039	2044	2152	2157	2162	2269	2274	2279	2386	2391	2396	2503	2508
	2513	2620	2625	2630	2740	2745	2750	4863	4870	4885					
SLASH	1057*														
SPACE	1057*														
SSECHT	1765*	2782	2865	2949	3032										
STARS	1057*	1247	1261	1263	1270	1283	1347	1350	1939	1953	2057	2071	2175	2188	2292
	2305	2409	2422	2526	2539	2643	2656	2763	2775	2847	2858	2929	2941	3014	3025
	3097	3108	3204	3215	3289	3303	3421	3434	3552	3566	3685	3698	3816	3829	3947
	3960	4078	4091	4210	4221	4315	4326	4400	4414	4533	4546	4666	4677	4759	4772
	4896	4907	4988	5002	5126	5137	5216	5228	5315	5333	5412	5430	5510	5523	5610
	5622	5727	7259	7336	7443	7454	7511	7564	7573	7661	7740	7817	7884	7961	7976
	8047	9071	8140	8193	8239	8248	8271								
SWRSU	1057*	.833*													
TRMTRP	8292*														
TYPBIN	1057*														
TYPDEC	1057*	5750	5757												
TYPNAM	1057*	1855													
TYPNUM	1057*														
TYPOCS	1057*														
TYPOCT	1057*	1882	7989												
TYPTXT	1057*	5746	5753												
\$\$CMRE	1281*	1320	1321	1322	1323	1324	1325	1326	1327						
\$\$CHTM	1281*	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	
\$\$ESCA	1057*														
\$\$NEWT	1057*	1939	2057	2175	2292	2409	2526	2643	2763	2847	2929	3014	3097	3204	3289
	3421	3552	3685	3816	3947	4078	4210	4315	4400	4533	4666	4759	4896	4988	5126
	5216	5315	5412	5510	5610										
\$\$SET	8292*	8301	8302	8303	8304	8306	8308	8309	8310	8311	8312	8313	8314		
\$\$SETM	1849*														
\$\$SKIP	1057*	2034	2039	2044	2152	2157	2162	2269	2274	2279	2386	2391	2396	2503	2508
	2513	2620	2625	2630	2740	2745	2750	4863	4870	4885					

N15

RK611 DISKLESS CONTROLLER DIAGNOSTIC: P5 MD-11-DZR6E-A MACY11 27(1006) 05-OCT-76 09:12 PAGE 197
DZR6EA.P11 28-SEP-76 15:31 CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0195

.EQUAT	923#	947
.HEADE	923#	925
.KT11	923#	1057
.SETUP	923#	1796
.SWRHI	923#	935
.SWRLO	923#	947#
.SACT1	923#	1245
.SAPT8	1348#	
.SAPTH	923#	1259
.SAPTY	923#	7452
.SCATC	923#	1231
.SCMTA	923#	1281
.SEOP	923#	5725
.SERRO	923#	7509
.SERRT	923#	
.SPOWE	923#	
.SRDOC	923#	8138
.SREAD	923#	7882
.SSAVE	923#	8191
.SSCOP	923#	7334
.SSIZE	923#	7257
.STRAP	923#	8269
.STYPD	923#	7815
.STYPE	923#	7659
.STYPO	923#	7738

. ABS. 052450 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DSKZ:DZR6EA,DSKZ:DZR6EA.SEQ/SOL/DOC/CRF=DZR6EA.P11
RUN-TIME: 86 81 8 SECONDS
RUN-TIME RATIO: 246/177=1.3
CORE USED: 31K (61 PAGES)

DOCUMENT PAGES: 195

Specular runtime 33 Seconds, 138 KCS, 861 disk reads, 0 disk writes, 195 pages

[illegible]