

M8743, M7981 MS11-L/M/P MEM DIAG  
M8722 CZMSPA0

AH-T157A-MC  
FICHE 1 OF 3

JUL 1982  
COPYRIGHT © 1982  
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	1001	1002	1003	1004	1005
1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020
1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035
1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050
1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065
1066	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080
1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094	1095
1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110
1111	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121	1122	1123	1124	1125
1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140
1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155
1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170
1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185
1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199	1200
1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215
1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230
1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245
1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260
1261	1262	1263	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275
1276	1277	1278	1279	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290
1291	1292	1293	1294	1295	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305
1306	1307	1308	1309	1310	1311	1312	1313	1314	1315	1316	1317	1318	1319	1320
1321	1322	1323	1324	1325	1326	1327	1328	1329	1330	1331	1332	1333	1334	1335
1336	1337	1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1348	1349	1350
1351	1352	1353	1354	1355	1356	1357	1358	1359	1360	1361	1362	1363	1364	1365
1366	1367	1368	1369	1370	1371	1372	1373	1374	1375	1376	1377	1378	1379	1380
1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391	1392	1393	1394	1395
1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407	1408	1409	1410
1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423	1424	1425
1426	1427	1428	1429	1430	143									



M8743, M7981 MS11-L/M/P MEM DIAG  
M8722 CZMSPA0

AH-T157A-MC  
FICHE 2 OF 3

JUL 1982  
COPYRIGHT © 1982  
MADE IN USA



Table with 15 columns and 15 rows of data. The data is organized into a grid, with each cell containing a small, complex diagram or schematic. The diagrams appear to be related to memory or system architecture, as indicated by the header text. The diagrams are arranged in a regular pattern, with each row and column containing a series of similar but distinct diagrams. The diagrams are small and detailed, showing various components and connections. The overall layout is a dense grid of these diagrams, filling the majority of the page below the header.



M8743, M7981 MS11-L/M/P MEM DIAG  
M8722 CZMSPA0

AH-T157A-MC  
FICHE 3 OF 3

JUL 1982  
COPYRIGHT © 1982  
MADE IN USA





CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 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

.TITLE CZMSPA0 MS11-L/M/P MEMORY DIAG.  
.REM

IDENTIFICATION  
-----

PRODUCT CODE: AC-T156A-MC  
PRODUCT NAME: CZMSPA0 MS11-L/M/P MEMORY DIAG  
PRODUCT DATE: MAY 1982  
MAINTAINER: STORAGE SYSTEMS S/W TEST APPLICATIONS  
  
COPYRIGHT(C): 1982

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE  
PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 2  
TABLE OF CONTENTS

110-	4559	DEFINE	TRAPS
111-	4675	DEFINE	BASIC PDP11 STUFF
111-	4758	DEFINE	CACHE REGISTERS
111-	4765	DEFINE	CPU REGISTERS
111-	4768	DEFINE	MEMORY MANAGEMENT REGISTERS
113-	4898	DEFINE	UNIBUS MAP REGISTERS
113-	4966	DEFINE	SOFTWARE SWITCH & DISPLAY REGISTERS
113-	4970	DEFINE	CONTROL STATUS REGISTERS
113-	4973	DEFINE	PARAMETERS
115-	4980	MACRO	FATAL
115-	5000	MACRO	TYPE
117-	5018	MACRO	NEWTST
119-	5068	MACRO	\$\$NEWTEST
119-	5089	MACRO	SUBTST
119-	5108	MACRO	\$\$SUBTST
121-	5123	MACRO	TYPOCT
123-	5163	MACRO	TYPOCS
125-	5219	MACRO	TYPDEC
126-	5261	MACRO	BMOV
128-	5327	MACRO	MAP
130-	5366	MACRO	SUPERVISOR
130-	5387	MACRO	USER
131-	5409	MACRO	TESTAREA
133-	5431	MACRO	SET4 & RES4
135-	5476	MACRO	DLEFT
137-	5500		TRAP CATCHER
137-	5508		ACT11 HOOKS
137-	5527		APT11 HOOKS
139-	5545	VARIABLES	INITIALIZED TO ZERO
141-	5717	VARIABLES	INITIALIZED TO NON ZERO
143-	5760		CONFIGURATION TABLE
144-	5787		***** MAIN *****
144-	5788		INITIALIZE VARIABLES TO ZERO
144-	5800		CLEAR NON-PROGRAM SPACE
145-	5813		TYPE OF SYSTEM SIZER
147-	5863		INITIALIZE VARIABLES TO NON ZERO
147-	5873		INITIALIZE VECTORS
149-	5893		INITIALIZE PATTERNS
149-	5922	SUBR	PLUG IN NULL PATTERNS
151-	5933		CLEAR THE CONFIGURATION TABLE
151-	5945		SIZE FOR A HARDWARE SWITCH REGISTER
153-	5963		SETUP ACT, APT, & XXDP
154-	5968		PROTECT PROGRAM & LOADERS
154-	6000		CHECK SYSTEM FOR CACHE
155-	6032		SETUP USER & SUPERVISOR STACK
155-	6050		GET SOFTWARE SWITCH REGISTER IF NECESSARY
155-	6057		GET MEMORY MANAGEMENT READY
157-	6063	T1	BIT TEST OF ALL CSR'S
158-	6120		DETERMINE TYPE OF ECC MEMORY
159-	6140		PRINT CSR REGISTER MAP
160-	6181		READ AND WRITE ALL CSR BITS
164-	6419		CLEAR ALL MEMORY SPACE FROM BANK 2 ON
166-	6449		MATCH ALL CSR'S WITH MEMORY
167-	6694	T2	TEST BANK 0 ACCESSES
167-	6723		ENABLE ECC FOR CORRECT TRAPS
169-	6731	T3	TEST BANKS 1-200 (OCTAL) FOR ZEROS & ONES
170-	6856		FIND SHADOW INHIBIT MODE POINTERS



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 2-1  
TABLE OF CONTENTS

172- 6879	T4	ECC INHIBIT MODE POINTER TEST	
182- 7036		LEGAL CONFIGURATION CHECK	
184- 7150		PRINT CONFIGURATION DETAILS	
186- 7207		CHECK APT SIZING	
187- 7242	T5	DIAGNOSTIC MODE DISPATCH ROUTINE	
187- 7259	T6	UNIQUE BANK TEST	
187- 7273		FLUSH OUT DBE'S	
189- 7277		END OF PASS ROUTINE	
191- 7339		WRITE BACKGROUND PATTERNS	
193- 7353		MTEST MODES	
193- 7355		BANKS FORWARD,PATTERNS FORWARD	**RECURSIVE**
195- 7385		BANKS FORWARD,PATTERNS REVERSE	**RECURSIVE**
197- 7415		BANKS WORST FIRST,PATTERNS FORWARD	**RECURSIVE**
199- 7452		BANKS WORST FIRST,PATTERNS REVERSE	**RECURSIVE**
201- 7489		PATTERNS FORWARD,BANKS FORWARD	**RECURSIVE**
203- 7527		PATTERNS FORWARD,BANKS WORST FIRST	**RECURSIVE**
205- 7572		PATTERNS REVERSE,BANKS FORWARD	**RECURSIVE**
207- 7610		PATTERNS REVERSE,BANKS WORST FIRST	**RECURSIVE**
209- 7655	SUBR	SETUP MEMORY TEST	
211- 7675	SUBR	TEST ECC CSR LOGIC DISPATCH	
213- 7766		CHECK FOR SBE FREE LOCATIONS	
215- 7861		CSR PATTERN CASE STATEMENT	
217- 7907	SUBR	ECC TEST DISPATCH	
219- 7962	SUBR	PARITY TEST DISPATCH	
220- 8009		PATTERNS	
220- 8011		MEMORY TEST SETUP ROUTINES	
220- 8012	MT0000	SETUP DATA PATTERN TEST	
220- 8025	MT0001	SETUP ADDRESS TEST	
220- 8047	MT0002	SETUP COMPLEMENT ADDRESS TEST	
222- 8075	MT0003	SETUP 3 XOR 9 WORST CASE NOISE TEST	
222- 8111	MT0004	SETUP ROTATING ZEROS TEST	
222- 8129	MT0005	SETUP ROTATING ONES TEST	
224- 8151	MT0006	SETUP INITIAL DATA TEST	
224- 8158	MT0007	SETUP ADDRESS BIT TEST	
224- 8168	MT0010	SETUP BYTE ADDRESSING TEST	
226- 8177	MT0011	SETUP CREATE SINGLE BIT ERROR TEST	
226- 8186	MT0012	SETUP WRITE BYTE CLEARS SBE TEST	
226- 8201	MT0013	SETUP CREATE DOUBLE BIT ERROR TEST	
227- 8212	MT0014	SETUP BASIC DOUBLE BIT ERROR TEST	
229- 8226	MT0015	SETUP WRITE INHIBIT OF BYTE WITH DBE	
229- 8235	MT0016	SETUP WRITE INHIBIT OF WORD WITH DBE	
229- 8244	MT0017	SETUP HOLDING 1'S & 0'S	
231- 8251	MT0020	SETUP SYNDROMES TO CSR ON SINGLE BIT ERROR	
232- 8264	MT0021	SETUP MARCHING 0'S & 1'S TEST	
233- 8311	MT0022	SETUP REFRESH & SHIFTING DIAGONAL TEST	
233- 8319	MT0023	SHIFTING DIAGONAL TEST	
234- 8329	MT0024	SETUP FAST GALLOPING PATTERN TEST	
234- 8371	MT0025	SETUP INTERRUPT ENABLE TEST	
236- 8382	MT0026	SETUP RANDOM DATA TEST	
238- 8429	MT0027	UNIQUE BANK TEST	
240- 8500	MT0030	SETUP FLUSH OUT DBE'S TEST	
242- 8545	MT0031	SETUP SOB-A-LONG TEST	
244- 8574	MT0032	SETUP WRITE RECOVERY TEST	
246- 8638	MT0033	SETUP BRANCH GOBBLE TEST	
246- 8668	MT0034	SOFT ERROR - BACKGROUND PATTERN TEST	
246- 8698	MT0035	SETUP WORST CASE NOISE PARITY TEST	
247- 8719	MT0036	SETUP CORRECTION CODE TEST	



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 2-2  
TABLE OF CONTENTS

248- 8731	MT0037	SETUP ECC DISABLE TEST
248- 8743	MT0040	
249- 8746	MT0041	SETUP ADDRESS TO CSR ON DOUBLE BIT ERROR TEST
250- 8755	MT0042	SETUP EXTENDED UNIBUS ADDRESS TO CSR TEST
251- 8765	MT0043	SETUP WRITE BYTE CLEARS SBE TEST
251- 8773	MT0044	SETUP SHIFTING 1/0'S THROUGH THE CHECK BITS TEST
251- 8781	MT0045	SETUP SYNDROMES TO CSR ON DOUBLE BIT ERROR
251- 8789	MT0046	SETUP CHECK SINGLE BIT ERRORS WITH ECC DISABLED TET
251- 8797	MT0047	SETUP NO CSR UPDATE ON SBE WITH EXSISTING DBE TEST
253- 8808	MT0999	SETUP NULL TEST
253- 8813		CHECK FOR KAMIKAZE MODE
255- 8821	SUBR	EXECUTE PATTERN IN SUPERVISOR
259- 8892	MEMORY	TEST PATTERN ROUTINES
259- 8902	MTP000	BASIC DATA TEST
259- 8913	MTP001	ADDRESS TEST
259- 8925	MTP002	COMPLEMENT ADDRESS TEST (WRITE DOWN, READ UP)
261- 8939	MTPA03	3 XOR 9 WORST CASE NOISE TEST (WRITE)
261- 8962	MTPB03	3 XOR 9 WORST CASE NOISE TEST (READ)
263- 8980	MTPC03	TEST DATA SUBPROGRAM
263- 8988	MTPD03	TEST DATA SUBSUBPROGRAM
265- 8998	MTPA04	ROTATING ZEROS TEST
265- 9011	MTPB04	SUBR ROTATING BIT
265- 9020	MTP005	ROTATION ONES TEST
267- 9034	MTP006	INITIAL DATA TEST
269- 9077	MTP007	ADDRESS BIT TEST
271- 9117	MTP010	BYTE ADDRESSING TEST
273- 9153	MTP011	SINGLE BIT ERROR TEST
275- 9294	MTP012	WRITE BYTE CLEARS SBE TEST
277- 9376	MTP013	CREATE DOUBLE BIT ERROR TEST
282- 9467	MTP014	BASIC DOUBLE BIT ERROR TEST
283- 9515	MTP015	WRITE INHIBIT OF BYTE WITH DBE
285- 9614	MTP016	WRITE INHIBIT OF WORD WITH DBE
289- 9716	MTP017	HOLDING 1'S & 0'S TEST
291- 9750	MTP020	SYNDROMES TO CSR ON SINGLE BIT ERROR TEST
292- 9818	MTPA21	MARCHING 1'S & 0'S PATTERN TEST
296- 9888	MTP022	REFRESH & SHIFTING DIAGONAL TEST
297- 9960	SUBR	REFRESH DELAY
299- 9982	MTPA24	FAST GALLOPING PATTERN TEST
301-10026	MTPB24	FAST GALLOP PART B
301-10034	MTPC24	FAST GALLOP PART C
303-10044	MTP025	INTERRUPT ENABLE TEST
307-10138	MTPA26	RANDOM DATA (WRITE)
307-10145	MTPB26	RANDOM DATA (READ)
307-10163	RANDOM	NUMBER SUBPROGRAM
307-10176	RANDOM	NUMBER SUBSUBPROGRAM
309-10184	MT0030	FLUSH OUT DBE'S
309-10190	MTP031	SOB-A-LONG TEST
311-10241	MTP032	WRITE RECOVERY TEST
313-10261	MTP033	BRANCH GOBBLE TEST
314-10307	MTP034	SOFT ERROR - BACKGROUND PATTERN TEST
315-10319	MTP035	WORST CASE NOISE PARITY TEST
316-10351	MTP036	CORRECTION CODE TEST
317-10404	MTP037	CHECK ECC DISABLE TEST
319-10425	MTP041	ADDRESS TO CSR ON DOUBLE BIT ERROR TEST
321-10466	MTP042	EXTENDED ADDRESS TO CSR ON ERROR TEST
322-10519	MTP043	WRITE BYTE CLEARS SINGLE BIT ERROR TEST
323-10560	MTP044	SHIFTING CHECK BITS THROUGH THE CSR TEST



CZASPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 2-3  
TABLE OF CONTENTS

324-10640	MTP045	SYNDROMES TO CSR ON DOUBLE BIT ERROR TEST
325-10680	MTP046	CHECK SINGLE BIT ERRORS WITH ECC DISABLED
326-10756	MTP047	NO CSR UPDATE ON SBE WITH EXSISTING DBE
327-10800	MISC	SUBROUTINES
327-10802	SUBR	COPY R0 TO R4, R1 TO R3, & R2 TO R5
327-10808	FLIP	WARNING CONSTANTS IN WORST CASE NOISE TESTS
328-10835	SUBR	WRITE BACKGROUND
329-10854	SUBR	GET CSR INFORMATION FROM CONFIGURATION TABLE
331-10868	SUBR	PRINT CONFIGURATION MAP
333-10920	SUBR	TYPE CONFIGURATION
337-11050	TRAP	PARITY ERROR HANDLER
339-11082	TRAP	NON-EXISTANT MEMORY (HOLES) HANDLER
339-11102	TRAP	TIMEOUT (TRAP TO 4) HANDLER
339-11106	TRAP	MEMORY MANAGEMENT (TRAP TO 250) HANDLER
339-11109	TRAP	RESERVED INSTRUCTION HANDLER
339-11119	FIND	BAD SP, PC, & PSW FROM STACK
341-11127	TRAP	KERNEL TRAP HANDLER
341-11135	TRAP	ENERGIZE TRAP HANDLER
341-11139	TRAP	DEENERGIZE TRAP HANDLER
341-11143	TRAP	CACHON TRAP HANDLER
341-11150	TRAP	CACHOFF TRAP HANDLER
343-11158	TRAP	LOAD CSR TRAP HANDLER
343-11177	TRAP	READ CSR TRAP HANDLER
344-11185	TRAP	TEST (R1) & READ CSR CAREFULLY
346-11222	TRAP	ECC DISABLE ALL CSR'S TRAP HANDLER
346-11226	TRAP	ECC DISABLE OF 1 SELECTED CSR TRAP HANDLER
346-11230	TRAP	INITIALIZE ALL CSR'S TRAP HANDLER
346-11234	TRAP	INITIALIZE 1 SELECTED CSR TRAP HANDLER
346-11238	TRAP	ENABLE SBE PARITY TRAPS ON ALL CSR'S
346-11242	TRAP	ENABLE SBE PARITY TRAPS ON 1 SELECTED CSR
346-11246	TRAP	WRITE CHECKBITS THRU ALL CSR'S TRAP HANDLER
346-11251	TRAP	WRITE CHECKBITS THRU 1 SELECTED CSR TRAP HANDLER
348-11258	TRAP	WAS THERE A SBE ON ANY CSR TRAP HANDLER
348-11283	TRAP	WAS THERE A SBE IN 1 SELECTED CSR TRAP HANDLER
350-11293	TRAP	WAS THERE A DBE ON ANY CSR TRAP HANDLER
350-11318	TRAP	WAS THERE A DBE ON 1 SELECTED CSR TRAP HANDLER
352-11329	TRAP	CLEAR ALL ECC CSR'S TRAP HANDLER
352-11333	TRAP	CLEAR 1 SELECTED CSR TRAP HANDLER
352-11337	TRAP	ECC DISABLE, CHECK MODE, & WRITE CHECKBITS IN ALL CSR'S TRAP HANDLER
352-11342	TRAP	ECC DISABLE, CHECK MODE, & WRITE CHECKBITS IN 1 SELECTED CSR
354-11349	SUBR	WRITE IN ALL CSR'S
354-11364	TRAP	INVALIDATE BACKGROUND PATTERN
355-11373	TRAP	GENERATE AND TEST ERROR ADDRESS
355-11427	TRAP	ENABLE CHECK/SYNDROME BIT REGISTER
357-11434	SUBR	GENERATE CHECK BITS
361-11503	SUBR	MAPPER
361-11588	TRAP	MAP KERNEL (ALMOST 1 TO 1) TRAP HANDLER
363-11611		RELOCATE PROGRAM
365-11716		UNRELOCATE PROGRAM
365-11761		SETUP LOWER 16K OF UNIBUS MAP
367-11774		MOVE BANKS
369-11822	SUBR	MAP USER TO NEW BANK
369-11842	SUBR	SETUP KERNEL PAR'S FOR NEW BANK
369-11855	SUBR	MAP KERNAL PARS 4 AND 5 TO A BANK
369-11865	SUBR	SETUP KERNEL PAR'S FOR NEW LOADER BANK
369-11876	SUBR	UNMAP KERNAL PAR'S 4 AND 5
371-11883	SUBR	EXAMINE BANK



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 2-4

## TABLE OF CONTENTS

373-11963	SUBR	BANK OK?
373-11974	SUBR	INCREMENT PATTERN TESTING
373-11982	SUBR	SET HIGHEST PATTERN TESTING TYPE
373-11986	SUBR	INCREMENT BANK & TEST
375-11993		BOOTSTRAP ROUTINE
377-12022		HALT PROGRAM
377-12031		SHUTDOWN DIAGNOSTIC
377-12058		APT SHUTDOWN SEQUENCE
379-12068		BLOCK MOVE SUBROUTINE
380-12095		FIELD SERVICE MODE
380-12097	SUBR	FIELD SERVICE COMMAND MODE
382-12147	COMMAND 0	EXIT
382-12169	FS	COMMAND 1 READ CSR
384-12184	FS	COMMAND 2 LOAD CSR
386-12208	FS	COMMAND 3 EXAMINE MEMORY
388-12250	FS	COMMAND 4 MODIFY MEMORY
390-12302	FS	COMMAND 5 SELECT BANK & PATTERN
391-12433	FS	COMMAND 6 TYPE CONFIGURATION MAP
393-12439	FS	COMMAND 7 SOB-A-LONG TEST
395-12480	FS	COMMAND 8 ERROR SUMMARY
397-12510	FS	COMMAND 9 REFRESH TEST
399-12551	FS	COMMAND 10 SET FILL COUNT
399-12561	FS	COMMAND 11 ENTER KAMIKAZE MODE
399-12566	FS	COMMAND 12 EXIT KAMIKAZE MODE
399-12572	FS	COMMAND 13 TURN CACHE OFF
399-12579	FS	COMMAND 14 TURN CACHE ON
400-12598	FS	COMMAND 15 TEST ONLY SELECTED BANKS
400-12618	FS	COMMAND 16 RESUME TESTING ALL BANKS
402-12632	FS	COMMAND 17 ENABLE TRACE
404-12638	FS	COMMAND 18 DISABLE TRACE
406-12644	SUBR	DETERMINE CORRECT CSR
421-13212		ERROR DATA (SUPERVISOR) SETUP STUFF
421-13226		DATA WAS 3 WORDS
423-13267		GET DATA FROM ABORTED AREA IF POSSIBLE
425-13283		POWER FAIL AUTO RESTART
425-13284		ROUTINE POWER DOWN AND UP
430-13472		POWER FAIL WHILE RELOCATED
432-13499		POWER UP FROM BANK 0 TO RELOCATION
434-13539		IO SUBROUTINES
434-13541		ROUTINE TYPE
449-14334		ERROR DATA SETUP
454-14583		DATA WAS A WORD
454-14595		DATA WAS A BYTE
456-14608		DATA WAS A 7 BIT BYTE
456-14623		DETERMINE XOR OF GOOD & BAD
458-14632		LOG ERROR ON BAD BANK
462-14721		ROUTINE SCOPE HANDLER
463-14791	SUBR	DISPLAY
465-14808		ROUTINE ERROR HANDLER
468-14923		ROUTINE ERROR MESSAGE TYPEOUT
476-15150	SUBR	DETAILED ERROR REPORT
481-15292		ROUTINE BINARY TO OCTAL (ASCII) AND TYPE
482-15370		ROUTINE CONVERT BINARY TO DECIMAL AND TYPE
483-15427		ROUTINE TTY INPUT
485-15522		CONTROL T
485-15547		CONTROL S & CONTROL Q
487-15667		ROUTINE READ AN OCTAL NUMBER FROM THE TTY



CZMSPA0 MS11-I /M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 2-5  
TABLE OF CONTENTS

487-15716	ROUTINE READ A DECIMAL NUMBER FROM THE TTY
488-15775	ROUTINE SAVE AND RESTORE R0-R5
489-15811	ROUTINE RANDOM NUMBER GENERATOR
491-15841	ROUTINE DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT
492-15883	TABLES
492-15885	APT MAILBOX-ETABLE
494-15967	ROUTINE TRAP DECODER
496-15994	TRAP TABLE
500-16097	TABLE ERROR POINTER
509-16385	ERROR DATA TAGS (DT)
511-16412	ERROR DATA FORMATS (DF)
513-16429	ERROR MESSAGES (EM)
515-16475	ERROR DATA HEADERS (DH)
517-16502	MESSAGES



41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56

REVISION HISTORY  
=====

REVISION =====	DATE =====	AUTHOR =====	CHANGES =====
CZMSPA	1-JUN-82	IRA CHAVIS	NONE - NEW PROGRAM



58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79

OPERATIONAL SWITCH SETTINGS  
SWITCH REGISTER DEFINITIONS

★	★	★
★	SWITCH	USE
★	-----	-----
★	15	HALT ON ERROR
★	14	LOOP ON TEST
★	13	INHIBIT ERROR TYPEOUTS
★	12	INHIBIT RELOCATION
★	11	QUICK VERIFY
★	10	BELL ON ERROR
★	9	LOOP ON ERROR
★	8	HALT PROGRAM (UNRELOCATED RESTORE LOADERS)
★	7	DETAILED ERROR REPORTS
★	6	INHIBIT CONFIGURATION MAP
★	5	LIMIT MAX ERRORS PER BANK
★	4	FAT TERMINAL (132 COLUMNS OR BETTER)
★	3	TEST MODE - SEE DOCUMENT
★	2	TEST MODE - SEE DOCUMENT
★	1	TEST MODE - SEE DOCUMENT
★	0	DETECT SINGLE BIT ERRORS

## TABLE OF CONTENTS

## 1.0 GENERAL PROGRAM INFORMATION

- 1.1 PROGRAM PURPOSE (ABSTRACT)
- 1.2 SYSTEM REQUIREMENTS
- 1.3 RELATED DOCUMENTS AND STANDARDS
- 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
- 1.5 ASSUMPTIONS

## 2.0 OPERATING INSTRUCTIONS

- 2.1 LOADING AND STARTING PROCEDURES
- 2.2 DEFAULT TEST SEQUENCE
- 2.3 SPECIAL ENVIRONMENTS
- 2.4 PROGRAM OPTIONS
- 2.5 EXECUTION TIMES

## 3.0 ERROR INFORMATION

- 3.1 ERROR REPORTING
- 3.2 ERROR ABBREVIATIONS
- 3.3 ERROR HALTS

## 4.0 PROGRESS REPORTS

## 5.0 CSR INFORMATION TABLES

- 5.1 MS11-P CSR
- 5.2 MS11-L CSR
- 5.3 MS11-M CSR

## 6.0 SUB-TEST SUMMARIES

- 6.1 TESTS
- 6.2 PATTERNS

## 7.0 PROGRAM FEATURES

- 7.1 FAST DATA ACCESS RATES
- 7.2 BANK ZERO TESTING
- 7.3 MEMORY CONFIGURATION: MAP
- 7.4 EVERYTHING YOU'VE ALWAYS WANTED TO KNOW ABOUT SUPERMAC ...
- 7.5 MEMORY MANAGEMENT MAPPING



## 1.0 GENERAL PROGRAM INFORMATION

## 1.1 PROGRAM PURPOSE (ABSTRACT)

- A. INTENDED FOR USE ON ALL PDP-11/24/44'S WHICH MEET THE CONDITIONS IN 1.2.1.
- B. THIS PROGRAM WILL BE USED BY SYSTEM MANAGERS AND OPERATORS TO DETERMINE THE CORRECT OPERATION OF MAIN MEMORY AND ALSO IT WILL BE PRIMARILY USED BY FIELD SERVICE AND MANUFACTURING TO ISOLATE FAILURES TO THE MEMORY AND TO ISOLATE FAILURES WITHIN THE MEMORY TO THE CORRECT CARD.
- C. THE OBJECT OF THIS SOFTWARE IS TO FUNCTIONALLY TEST AND VERIFY ALL MAIN MEMORY FUNCTIONS AS FAST AS POSSIBLE.
- D. THERE IS THE CAPABILITY OF TESTING MIXED CONFIGURATIONS (MS11-L, MS11-M AND MS11-P) ON THE SYSTEM.
- E. IT HAS SPECIAL A MAINTENANCE MODE (FIELD SERVICE MODE) TO PROVIDE SPECIFIC FUNCTIONAL CAPABILITIES.

## 1.2 SYSTEM REQUIREMENTS

## 1.2.1 HARDWARE REQUIREMENTS -

PDP-11-24/44 CPU WITH 22 BIT ADDRESSING AND AT LEAST 64K (16 BIT WORDS) OF MEMORY AND MEMORY MANAGEMENT.

## NOTE

- 1. LIKE MEMORY TYPES MUST BE ON 16K WORD BOUNDARIES STARTING AT PHYSICAL ADDRESS 0.
- 2. PDP-11 SERIES 16/18 PIT PROCESSORS ARE NOT SUPPORTED.

## 1.2.2 SOFTWARE REQUIREMENTS -

THIS PROGRAM IS DESIGNED TO RUN STAND ALONE OR UNDER ANY OF THE FOLLOWING MONITORS:

XXDP  
ACT  
APT

## 1.3 RELATED DOCUMENTS AND STANDARDS

1. PDP-11/04/24/34/44/70 PROCESSOR HANDBOOK (EB-19402)
2. PDP-11/44 USER'S GUIDE (EK-11044-UG)
3. MS11-M USER'S GUIDE (EK-MS11M-UG-001)
4. MS11-L USERS GUIDE (EK-MS11L-UG-001)
5. MS11-P TECHNICAL MANUAL (EK-MS11P-TM-001)

## 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

IF THE PROGRAM IN ANY WAY MISBEHAVES, THEN:

1. TRY IT AGAIN WITH CACHE OFF (REFERENCE SECTION 2.4.3.1)
2. INHIBIT RELOCATION (REFERENCE SECTION 2.4.1)
3. TRY CPU DIAGNOSTICS
4. TRY MEMORY MANAGEMENT DIAGNOSTICS
5. TRY CACHE DIAGNOSTICS (WHERE APPLICABLE)
6. TRY UNIBUS MAP DIAGNOSTICS (WHERE APPLICABLE)



## 216 1.5 ASSUMPTIONS

217 THIS PROGRAM ASSUMES THE CORRECT OPERATION OF THE CPU, MEMORY  
218 MANAGEMENT, CACHE, AND THE UNIBUS MAP. THIS PROGRAM OCCUPIES  
219 (INITIALLY) BANK 0 (0-16K). THE XXDP LOADERS ARE IN BANK 1.  
220  
221  
222  
223

## 224 2.0 OPERATING INSTRUCTIONS

## 225 2.1 LOADING STARTING PROCEDURES

## 226 2.1.1 QUICK STARTING -

- 227 1. LOAD ADDRESS 200  
228 2. SET SWITCH REGISTER FOR OPTIONS (NORMALLY 0)  
229 3. START  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240

## 241 NOTE

242 IF ON AN 11/24 USING MS11-L MEMORY BE  
243 SURE THAT THE PERIPHERAL PAGE JUMPER IS  
244 IN PLACE; FAILURE TO DO SO SENDS THE  
245 DIAGNOSTIC TO NEVER-NEVER LAND.  
246  
247  
248  
249  
250

## 251 2.1.2 STOPPING -

- 252 1. SET SW8, AND/OR  
253 2. TYPE CONTROL "C" (REFERENCE SECTION 2.4.4.1).  
254  
255  
256  
257  
258  
259  
260

## 261 2.1.3 RESTARTING (PRESERVE CONFIGURATION TABLE) -

- 262 1. LOAD ADDRESS 202  
263 2. SET SWITCH REGISTER FOR OPTIONS (NORMALLY 0)  
264 3. START  
265  
266  
267  
268

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

## 2.1.4 SWITCH REGISTER OPTIONS -

SWITCH	USE
-----	-----
15	HALT ON ERROR
14	LOOP ON TEST
13	INHIBIT ERROR TYPEOUTS
12	INHIBIT RELOCATION
11	QUICK VERIFY
10	BELL ON ERROR
9	LOOP ON ERROR
8	HALT PROGRAM (UNRELOCATE RESTORE LOADERS)
7	DETAILED ERROR REPORTS
6	INHIBIT CONFIGURATION MAP
5	LIMIT MAX ERRORS PER BANK
4	FAT TERMINAL (132 COLUMNS OR BETTER)
3	TEST MODE - SEE DOCUMENT
2	TEST MODE - SEE DOCUMENT
1	TEST MODE - SEE DOCUMENT
0	DETECT SINGLE BIT ERRORS



## 2.2 DEFAULT TEST SEQUENCE

THE FOLLOWING TWO LISTS GIVE THE TEST PROTOCOL FOR PARITY AND ECC MEMORY. TESTS MARKED WITH A '\*' ARE NOT NORMALLY RUN EXCEPT UNDER ACT OR APT, OR THROUGH A FIELD SERVICE COMMAND (REFERENCE SECTION 2.4.4.8).

## 2.2.1 TEST PROTOCOL FOR MS11-L PARITY MEMORY -

TEST	TEST NAME	TIME (SEC/16K)
34	SOFT ERROR TEST	<1
6	INITIAL DATA TEST	<1
17	HOLDING 1'S AND 0'S TEST	<1
7	ADDRESS BIT TEST	<1
1	ADDRESS TEST	<1
2	COMPLEMENT ADDRESS TEST	<1
3	3 XOR 9 TEST	1
4	ROTATING 0'S TEST	1
5	ROTATING 1'S TEST	1
21	MARCHING 1'S AND 0'S TEST	1
35	WORST CASE NOISE PARITY TEST	N/A
* 22	REFRESH TEST	10
* 23	SHIFTING DIAGONAL TEST	10
26	RANDOM DATA TEST	<1
* 24	FAST GALLOPING PATTERN TEST	20
* 31	SOB-A-LONG TEST	3
* 32	WRITE RECOVERY TEST	<1
* 33	BRANCH GOBBLE TEST	35
34	SOFT ERROR TEST	<1

## 2.2.2 TEST PROTOCOL FOR MS11-M ECC MEMORY -

TEST	TEST NAME	TIME (SEC/16K)
5	ROTATING 1'S TEST	1
@ 25	INTERRUPT ENABLE TEST	<1
+@ 11	SINGLE BIT ERROR TEST	<2
+@ 12	WRITE BYTE CLEARS SBE TEST	<1
+@ 13	CREATE DOUBLE BIT ERROR TEST	1
+@ 15	WRITE INHIBIT OF BYTE W/DBE TEST	1
+@ 16	WRITE INH. OF WORD W/DBE TEST	<1
34	SOFT ERROR TEST	<1
6	INITIAL DATA TEST	<1
10	BYTE ADDRESS TEST	<1
17	HOLDING 1'S AND 0'S TEST	<1
7	ADDRESS BIT TEST	<1
1	ADDRESS TEST	<1
2	COMPLEMENT ADDRESS TEST	<1
4	ROTATING 0'S TEST	1
5	ROTATING 1'S TEST	1
21	MARCHING 0'S AND 1'S TEST	1
* 22	REFRESH TEST	10
26	RANDOM DATA TEST	<1
* 24	FAST GALLOPING PATTERN TEST	20
* 31	SOB-A-LONG TEST	3
* 32	WRITE RECOVERY TEST	<1
* 33	BRANCH GOBBLE TEST	35
34	SOFT ERROR TEST	<1

@ - RUN ONLY ON THE FIRST PASS WHEN UNDER ACT OR APT

+ - RUN TWICE FOR EACH 16K BANK IF INTERLEAVED

AT THE END OF EACH PASS THE PROGRAM WILL RUN CLEANUP PATTERNS  
#30, AND #27 FOR ALL BANKS.



## 2.2.3 TEST PROTOCOL FOR MS11-P ECC MEMEORY

PATTERN	PATTERN NAME	TIME (SEC/16K)
5	ROTATING 1'S TEST	1
34	SOFT ERROR TEST	<1
6	INITIAL DATA TEST	<1
44	SHIFTING CHECK BITS THRU	1
14	BASIC DBE TEST CSR TEST	<1
45	SYNDROMES IN CSR ON DBE TEST	<1
36	CORRECTION CODE TEST	1
20	SYNDROMES IN CSR ON SBE TEST	1
37	CHECK ECC DISABLE TEST	<1
41	ADDRESS TO CSR ON DBE TEST	1
42	EXTENDED ADDRESS TO CSR TEST	<1
43	BYTE WRITE TEST	<1
46	CHECK SBE WITH ECC DISABLE TEST	<1
47	NO CSR UPDATE ON SBE WITH DBE TEST	<1
10	BYTE ADDRESS TEST	<1
17	HOLDING 1'S AND 0'S TEST	<1
7	ADDRESS BIT TEST	<1
1	ADDRESS TEST	<1
2	COMPLEMENT ADDRESS TEST	<1
4	ROTATING 0'S TEST	1
5	ROTATING 1'S TEST	1
21	MARCHING 0'S AND 1'S TEST	1
* 22	REFRESH TEST	10
26	RANDOM DATA TEST	<1
* 24	FAST GALLOPING PATTERN TEST	20
* 31	SOB-A-LONG TEST	3
* 32	WRITE RECOVERY TEST	<1
* 33	BRANCH GOBBLE TEST	35
34	SOFT ERROR TEST	<1

a - RUN ONLY ON THE FIRST PASS WHEN UNDER ACT OR APT

AT THE END OF EACH PASS THE PROGRAM WILL RUN CLEANUP PATTERNS  
#30, AND #27 FOR ALL BANKS.

## 2.3 SPECIAL ENVIRONMENTS

## 2.3.1 XXDP -

THE FIRST PASS WILL BE A QUICK VERIFY PASS IF AND ONLY IF IT IS IN CHAIN MODE.

## 2.3.2 ACT APT AUTOMATIC MODE -

THE PROGRAM WILL NOT CREATE DOUBLE BIT ERRORS (DBE'S) AFTER THE 1ST PASS.

## 2.3.2.1 APT EXECUTION TIMES -

HERE ARE SOME MEASURED EXECUTION TIMES FOR AN 11/44 WITH CACHE UNDER APT

	1ST QV PASS	2ND PASS ONWARD
128K MS11-M (NON-INTERLEAVED)	10 MIN 15 SEC	7 MIN 40 SEC
128K MS11-L	9 MIN 50 SEC	7 MIN 30 SEC
256K MS11-M (INTERLEAVED)	19 MIN 50 SEC	14 MIN 45 SEC
512K MS11-P	NOT ESTABLISHED AT RELEASE TIME	

THE FIRST PASS WILL BE A QUICK VERIFY PASS

## NOTE

EVEN THOUGH THE FIRST PASS IS A QV PASS IT TAKES LONGER THAN THE SUBSEQUENT NON-QV PASSES DUE TO THE FACT THAT IT IS RUNNING MORE PATTERNS, SOME OF WHICH (PATTERNS #24 AND #33 FOR EXAMPLE) CAN BE EXTREMELY TIME CONSUMING.

## 2.3.2.2 APT ENVIRONMENT TABLE -

THE FOLLOWING TABLE GIVES SOME OF THE STANDARD SETTINGS FOR THE APT E-TABLE. THEY MAY BE MODIFIED AS NOTED AS THE USER SEES FIT.

## FIRST PASS RUN TIME:

THIS PARAMETER SHOULD BE SET ACCORDING TO THE AMOUNT AND TYPE OF MEMORY TO BE TESTED. THE ABOVE TABLE (APT EXECUTION TIMES) GIVES SOME MEASURED TIMES. FOR ANY PATTERNS DELETED (THROUGH USE OF THE DEVICE DESCRIPTOR WORDS) REFERENCE SECTION 2.2 FOR INDIVIDUAL PATTERN TIMES.

## NOTE

THE TIMES GIVEN IN SECTION 2.2 ARE FOR 16K CHUNKS OF MEMORY, NOT 128K BOARDS!

## LONGEST TEST TIME:

THIS PARAMETER SHOULD BE SET TO THE EXECUTION TIME OF THE LONGEST PATTERN BEING RUN. FOR THE DEFAULT CASE THIS IS 35 SECONDS FOR PATTERN #33.

## ADDITIONAL RUN TIME:

NOT USED BY PROGRAM.

## SOFTWARE ENVIRONMENT:

FOR APT AUTO MODE THIS PARAMETER SHOULD BE SET TO A '1'. FOR DUMP MODE SET THIS TO A '0'.

## ENVIRONMENT MODE:

WHEN THIS PARAMETER IS SET TO A '0' THE PROGRAM DOES IT'S OWN SIZING. IF THE USERS SETS BIT #7 HOWEVER, HE MUST SPECIFY THE TYPES AND AMOUNTS OF MEMORY TO BE TESTED.

## SWITCH 1:

THE DEFAULT SETTING OF THIS SWITCH IS '101'. APT USES THIS AS THE SWITCH REGISTER FOR THE PROGRAM. REFERENCE SECTION 2.4.1 FOR MORE INFORMATION ON SWITCH SETTINGS.

## SWITCH 2:

THIS SWITCH, IF SET TO ANY NON-ZERO NUMBER, IS USED TO LIMIT THE AMOUNT OF PASSES APT WILL MAKE. THE PROGRAM WILL HANG AFTER THIS COUNT HAS BEEN REACHED.

## CPU OPTIONS:

NOT USED BY PROGRAM.

## MEMORY TYPE N (N=1 TO 4)

IF BIT #7 OF ENVIRONMENT MODE IS SET THESE FOUR WORDS ARE USED TO LOG THE DIFFERENT TYPES OF MEMORY TO BE TESTED. IF BIT #7 IS NOT SET THESE LOCATION ARE NOT USED.

## MAXIMUM ADDRESS N (N=1 TO 4)

THESE FOUR WORDS ARE USED IN CONJUNCTION WITH THE CORRESPONDING



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 15-1

554 MEMORY TYPE WORDS TO INDICATE THE HIGHEST ADDRESS THAT MEMORY TYPE  
555 OCCUPIES.  
556

## NOTE

557  
558 THE ABOVE TWO PARAMETERS DO NOT ACTUALLY  
559 HAVE TO REPRESENT AN ACCURATE  
560 CONFIGURATION OF MEMORY. ALL THE  
561 PROGRAM LOOKS FOR IS AN ACCURATE TALLY  
562 OF MEMORY AMOUNT!  
563  
564  
565  
566

567 INTERRUPT VECTOR N (N=1 TO 2)  
568 NOT USED BY PROGRAM.  
569

570 BUS PRIORITY N (N=1 TO 2)  
571 NOT USED BY PROGRAM.  
572

573 BASE ADDRESS:  
574 NOT USED BY PROGRAM.  
575

576 DEVICE MAP:  
577 NOT USED BY PROGRAM.  
578

579 CONTROLLER DESCRIPTOR CODE N (N=1 TO 2)  
580 NOT USED BY PROGRAM.  
581

582 DEVICE DESCRIPTOR CODES:  
583 THE DEVICE DESCRIPTOR CODES ARE USED BY THE PROGRAM TO DETERMINE  
584 WHICH PATTERNS IT WILL RUN. THE DEFAULT VALUES OF THESE WORDS ARE ALL  
585 '1'S, INDICATING THAT ALL OF THE PATTERNS SHOWN IN SECTION 2.2 ARE  
586 EXECUTED (SAVE FOR EXCEPTIONS AS NOTED THERE). EACH SET OF WORDS  
587 CONTROLS A TABLE IN THE PROGRAM AS FOLLOWS:  
588

589 DD WORDS PROGRAM TABLE (SYMBOLIC LOCATION)  
590  
591 WORDS 0-1 MKCSRT  
592  
593 WORDS 2-3 MKPAT  
594  
595 WORDS 4-5 MJPAT  
596  
597

598 BIT #0 SET IN THE FIRST WORD INDICATES THAT THE FIRST PATTERN IN THE  
599 TABLE WILL BE EXECUTED, BIT #1 THE SECOND, BIT #2 THE THIRD,... BIT  
600 #0 OF THE SECOND WORD INDICATES THAT THE 17TH ENTRY IN THE TABLE WILL  
601 BE EXECUTED, AND SO ON.  
602

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 16

604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623

### 2.3.3 NO SBE FREE BANKS -

IF THE PROGRAM CANNOT FIND ANY SBE (SINGLE BIT ERROR) FREE LOCATIONS (IN NON-PROTECTED ECC MEMORY) IT WILL PRINT OUT AN ERROR MESSAGE AND CONTINUE TESTING BY-PASSING THE ECC LOGIC TESTS.

### 2.3.4 MIXED PARITY ECC CONFIGURATIONS -

THE PROGRAM WILL FUNCTION NORMALLY IN MIXED ENVIRONMENTS. THE SEQUENCE OF TESTING MAY SEEM STRANGE DUE TO THE RECURSIVE TEST MODE ALGORITHM (REFERENCE SECTIONS 2.4.1.1, 2.4.1.2, 2.4.1.3).

## 2.4 PROGRAM OPTIONS

## 2.4.1 SWITCH REGISTER DETAILS -

IF A HARDWARE SWITCH REGISTER IS NOT AVAILABLE THEN THE SOFTWARE SWITCH REGISTER IS IN LOCATION 176. IF UNDER APT IF BIT7 IS SET IN THE E-TABLE SYMBOLIC LOCATION '\$ENVM' THE APT SOFTWARE SWITCH REGISTER WILL BE USED (LOCATION \$SWREG).

TO CHANGE THE SOFTWARE SWITCH REGISTER CONTENTS: TYPE 'CONTROL G'. THIS WILL CAUSE DISPLAY THE CURRENT VALUE OF THE SWR AND PROMPT FOR THE OCTAL INPUT OF THE NEW SWR VALUE FROM THE TERMINAL. THIS ROUTINE WILL IGNORE YOU (NOT RESPOND TO CONTROL 'G') IF YOU HAVE A HARDWARE SWITCH REGISTER.

SW15 = HALT ON ERROR  
(100000)

CONTINUING FROM THIS HALT WILL FIRST CHECK FOR A CHANGE IN THE SOFTWARE SWITCH REGISTER ('CONTROL G' IN THE TTY INPUT BUFFER) THEN IT WILL CONTINUE TESTING.

SW14 = LOOP ON TEST  
(40000)

THIS WILL CAUSE LOOPING ON THE PRESENT TEST OR PATTERN (BACK TO LAST SCOPE TRAP). IF IN A PATTERN THEN THE LOOPING WILL BE FOR AN ENTIRE BANK OF 16K ADDRESSES.

SW13 = INHIBIT ERROR TYPEOUTS  
(20000)

THIS WILL CAUSE RETURNS FROM THE ERROR ROUTINE WITHOUT THE TYPED MESSAGES. OTHER ON ERROR FUNCTIONS ARE NOT AFFECTED.

SW12 = INHIBIT RELOCATION  
(10000)

THIS PREVENTS THE PROGRAM FROM MOVING AND CONSEQUENTLY PREVENTS THE PROGRAM FROM TESTING AT LEAST 32K OF MEMORY.

SW11 = QUICK VERIFY  
(4000)

IF THIS SWITCH IS SELECTED APPROXIMATELY ONE 64TH OF THE POSSIBLE COMBINATIONS OF SBE'S DBE'S ARE TESTED.

EACH PASS COMPLETE TYPEOUT WILL INDICATE THIS MODE BY PRECEDING THE PASS NUMBER WITH 'QV'.



679 SW10 = BELL ON ERROR  
680 (2000)  
681  
682 THIS CAUSES A BELL (OR BEEP OR CLICK) ON EACH ERROR  
683 TRAP  
684  
685 SW9 = LOOP ON ERROR  
686 (1000)  
687  
688 THIS WILL CAUSE LOOPING FROM FAILURE POINT BACK TO THE  
689 LAST CORRECTLY INITIALIZED AREA OF THE CURRENT TEST.  
690  
691 SW8 = HALT PROGRAM  
692 (400)  
693  
694 THIS INITIATES THE FOLLOWING SEQUENCE:  
695  
696 1. IF PROGRAM IS RELOCATED IT MOVES BACK TO BANK ZERO.  
697  
698 2. FLUSH OUT ALL POSSIBLE DBE'S.  
699  
700 3. TURNS OFF MEMORY MANAGEMENT.  
701  
702 4. RESTORE LOADERS.  
703  
704 5. UNMAP THE UNIBUS MAP (IF THERE IS ONE).  
705  
706 6. HALT IF UNDER APT OR ACT BRANCH SEL.  
707  
708  
709  
710 SW7 = DETAILED ERROR REPORTS  
711 (200)  
712  
713 AFTER ANY NORMAL ERROR REPORT IS TYPED THIS OPTION  
714 CAUSES THE CONTENTS OF THE FOLLOWING REGISTERS TO BE  
715 TYPED:  
716 R0, R1, R2, R3, R4, R5, SP, 'CONTROL', 'CPUERR'  
717  
718 SW6 = INHIBIT CONFIGURATION MAP  
719 (100)  
720  
721 THIS INHIBITS THE PRINTING OF A MAP SHOWING THE MEMORY  
722 CONFIGURATION - REFERENCE SECTION 7.3  
723  
724 SW5 = LIMIT MAX ERRORS PER BANK  
725 (40)  
726  
727 THIS WILL LIMIT THE NUMBER OF ERROR TYPEOUTS PER BANK.  
728 THE DEFAULT IS 10. DECIMAL, HOWEVER THIS CAN BE  
729 CHANGED BY CHANGING LOCATION 'ERRMAX' MANUALLY.  
730

SW4 = FAT TERMINAL  
(20)

THIS INFORMS THE PROGRAM THAT THE CONSOLE TERMINAL HAS  
A WIDTH OF AT LEAST 132 COLUMNS (LA36 WITH WIDE PAPER).

SW3-1 = TEST MODE

TEST MODES DETERMINE THE RECURSION ALGORITHM TO BE USED  
DURING PATTERN TESTS.

#### MODE NAME DESCRIPTION

(0)	0	BAFPAF	BANKS FORWARD, PATTERNS FORWARD
(2)	1	BAFPAF	BANKS FORWARD, PATTERNS REVERSE
(4)	2	BAWPAF	BANKS WORST FIRST, PATTERNS FORWARD.
(6)	3	BAWPAF	BANKS WORST FIRST, PATTERNS REVERSE.
(10)	4	PAFBAF	PATTERNS FORWARD, BANKS FORWARD
(12)	5	PAFBAW	PATTERNS FORWARD, BANKS WORST FIRST
(14)	6	PARBAF	PATTERNS REVERSE, BANKS FORWARD
(16)	7	PARBAW	PATTERNS REVERSE, BANKS WORST FIRST.

FOR MORE DETAILS REFERENCE SECTION 2.4.1.1, 2.4.1.2 AND  
2.4.1.3.

SW0 = DETECT SINGLE BIT ERRORS (SBI'S)  
(1)

FOR MANUFACTURING PURPOSES THIS SWITCH SHOULD ALWAYS BE  
ON. FOR FIELD SERVICE PURPOSES THIS SWITCH SHOULD  
ALWAYS BE OFF.

THIS SWITCH WILL ALLOW ALL ECC SINGLE BIT ERRORS TO BE  
REPORTED BY DISABLING ERROR CORRECTION.

ERROR PRINTOUTS OF SBE'S ARE NOT DISTINGUISHABLE FROM  
DBE'S.

#### NOTE

IF DOUBLE BIT ERRORS ARE FOUND IN THE MEMORY,  
THIS SWITCH SHOULD BE SET TO MAKE SURE THAT NEW  
DATA CAN BE WRITTEN TO THE DBE LOCATIONS.

#### 2.4.1.1 TEST MODE EXAMPLE -

EXAMPLE ANALYSIS OF MODE 5 'PAFBAW'. ASSUME BANKS 0 1 ARE MS11-L  
AND BANKS 2,3,4, 5 ARE MS11-M.

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 20

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  
837ASSUME ALSO THAT BANK 3 IS KNOWN BAD BY THE PROGRAM VIA THE SIZING  
ROUTINE OR PREVIOUS RUNS THE TESTING SEQUENCE WOULD BE AS FOLLOWS:;TEST MS11-M MEMORY TYPES FIRST  
;TEST KNOWN BAD MEMORY (BANK 3)TEST 17. BANK 3  
TEST 7. BANK 3  
TEST 1. BANK 3  
TEST 2. BANK 3  
TEST 4. BANK 3  
TEST 5. BANK 3  
TEST 21. BANK 3  
TEST 20. BANK 3  
TEST 22. BANK 3  
TEST 26. BANK 3

;TEST PRESUMED GOOD MEMORY (BANKS 2,4,5)

TEST 17. BANK 2  
TEST 7. BANK 2  
TEST 1. BANK 2  
TEST 2. BANK 2  
TEST 4. BANK 2  
TEST 5. BANK 2  
TEST 21. BANK 2  
TEST 20. BANK 2  
TEST 22. BANK 2  
TEST 26. BANK 2  
TEST 17. BANK 4  
TEST 7. BANK 4  
TEST 1. BANK 4  
TEST 2. BANK 4  
TEST 4. BANK 4  
TEST 5. BANK 4  
TEST 21. BANK 4  
TEST 20. BANK 4  
TEST 22. BANK 4  
TEST 26. BANK 4  
TEST 17. BANK 5  
TEST 7. BANK 5  
TEST 1. BANK 5  
TEST 2. BANK 5  
TEST 4. BANK 5  
TEST 5. BANK 5  
TEST 21. BANK 5  
TEST 20. BANK 5  
TEST 22. BANK 5  
TEST 26. BANK 5

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

;RELOCATE TEST PROGRAM SPACE (BANK 0 & 1)

TEST 1, BANK 0  
TEST 2, BANK 0  
TEST 3, BANK 0  
TEST 4, BANK 0  
TEST 5, BANK 0  
TEST 26, BANK 0  
TEST 1, BANK 1  
TEST 2, BANK 1  
TEST 3, BANK 1  
TEST 4, BANK 1  
TEST 5, BANK 1  
TEST 26, BANK 1

#### NOTE

THIS IS AN EXAMPLE NOT AN ACTUAL  
SEQUENCE.

THE TEST SEQUENCE WAS FORWARD (THE SIMPLE PATTERNS FIRST, COMPLEX  
TESTS LAST) SEQUENCE OF PATTERNS (MS11-M = 17, 7, 1, 2, 4, 5, 21,  
20, 22, 26) (MS11-L = 1, 2, 3, 4, 5, 26).

IF THE BANK SELECTION IS FORWARD THE BANKS WILL BE TESTED IN THE  
FOLLOWING ORDER:

1. ECC BANKS THAT ARE NOT PROTECTED OR PROGRAM SPACE (FROM 0 TO 167).
2. PARITY BANKS THAT ARE NOT PROGRAM SPACE (FROM 0 TO 167).
3. THE PROGRAM NOW RELOCATES TESTS:
4. ECC BANKS THAT WERE PROTECTED OR PROGRAM SPACE (FROM 0 TO 167).
5. PARITY BANKS THAT WERE PROGRAM SPACE (FROM 0 TO 167).

IF BANK SELECTION IS WORST FIRST THE CONFIGURATION TABLE WILL BE  
CONSULTED AND BANKS WILL BE TESTED IN THE FOLLOWING ORDER.

1. ECC BANKS THAT ARE KNOWN BAD AND ARE NOT PROTECTED OR PROGRAM  
SPACE (FROM 0 TO 167).
2. PARITY BANKS THAT ARE KNOWN BAD AND ARE NOT PROGRAM SPACE  
(FROM 0 TO 167).
3. ECC BANKS THAT ARE PRESUMED GOOD AND ARE NOT PROTECTED OR



892 PROGRAM SPACE (FROM 0 TO 167).

- 893
- 894 4. PARITY BANKS THAT ARE PRESUMED GOOD AND ARE NOT PROGRAM SPACE
- 895 (FROM 0 TO 167).
- 896
- 897 5. THE PROGRAM NOW RELOCATES TESTS:
- 898
- 899 6. ECC BANKS THAT ARE KNOWN BAD AND WERE PROTECTED OR PROGRAM
- 900 SPACE (FROM 0 TO 167).
- 901
- 902 7. PARITY BANKS THAT ARE KNOWN BAD AND WERE PROGRAM SPACE (FROM
- 903 0 TO 167).
- 904
- 905 8. ECC BANKS THAT ARE PRESUMED GOOD AND WERE PROTECTED OR
- 906 PROGRAM SPACE (FROM 0 TO 167).
- 907
- 908 9. PARITY BANKS THAT ARE PRESUMED GOOD AND WERE PROGRAM SPACE
- 909 (FROM 0 TO 167).
- 910
- 911
- 912

913 2.4.1.2 TEST MODE DETAILS -

914 MODE 0 = 'BAFPF' BANKS FORWARD, PATTERNS FORWARD

915 THIS IS THE DEFAULT AND SIMPLEST MODE.

916 THIS MODE TESTS EACH BANK COMPLETELY FROM 0 TO 167

917 EXCEPT THOSE REQUIRING RELOCATION\*.

918 WHILE TESTING EACH BANK THE PATTERNS ARE RUN WITH THE

919 SIMPLE ONES FIRST BUILDING TO THE MORE COMPLEX.

920 MODE 1 = 'BAFPA' = BANKS FORWARD, PATTERNS REVERSE

921 THIS MODE TESTS EACH BANK COMPLETELY FROM 0 TO 167

922 EXCEPT THOSE REQUIRING RELOCATION\*.

923 WHILE TESTING EACH BANK THE PATTERNS ARE RUN WITH THE

924 MOST COMPLEX ONES FIRST, WORKING TO THE SIMPLE ONES.

925 MODE 2 = 'BAWPAF' = BANKS WORST FIRST, PATTERNS FORWARD

926 THIS MODE FIRST TESTS EACH KNOWN BAD BANK COMPLETELY

927 FROM 0 TO 167 EXCEPT THOSE REQUIRING RELOCATION\*, THEN

928 PRESUMED GOOD BANKS ARE TESTED FROM 0 TO 167 EXCEPT

929 THOSE REQUIRING RELOCATION\*.

930 WHILE TESTING EACH BANK THE PATTERNS ARE RUN WITH THE

931 SIMPLE ONES FIRST, BUILDING TO THE MORE COMPLEX.

932 MODE 3 = 'BAWPA' = BANKS WORST FIRST, PATTERNS REVERSE

933 THIS MODE FIRST TESTS EACH KNOWN BAD BANK COMPLETELY

934

935

936

937

938

939

940

941

942

943

944

945

947 FROM 0 TO 167 EXCEPT THOSE REQUIRING RELOCATION\*, THEN  
948 PRESUMED GOOD BANKS ARE TESTED FROM 0 TO 167 EXCEPT  
949 THOSE REQUIRING RELOCATION\*.

950 WHILE TESTING EACH BANK THE PATTERNS ARE RUN WITH THE  
951 MOST COMPLEX ONES FIRST, WORKING TO THE SIMPLE ONES.

952  
953 MODE 4 = 'PAFBAF' = PATTERNS FORWARD, BANKS FORWARD

954 THIS MODE TESTS EACH PATTERN COMPLETELY WITH THE SIMPLE  
955 ONES FIRST, BUILDING TO THE MORE COMPLEX.

956 WHILE TESTING EACH PATTERN THE BANKS ARE RUN FROM 0 TO  
957 167 EXCEPT THOSE REQUIRING RELOCATION\*.

958  
959 MODE 5 = 'PAFBAW' = PATTERNS FORWARD, BANKS WORST FIRST

960 THIS MODE TESTS EACH PATTERN COMPLETELY WITH THE SIMPLE  
961 ONES FIRST, BUILDING TO THE MORE COMPLEX.

962 WHILE TESTING EACH PATTERN FIRST EACH KNOWN BAD BANK  
963 FROM 0 TO 167 EXCEPT THOSE REQUIRING RELOCATION\* IS  
964 RUN, THEN PRESUMED GOOD BANKS ARE RUN FROM 0 TO 167  
965 EXCEPT THOSE REQUIRING RELOCATION\*.

966  
967 MODE 6 = 'PARBAF' = PATTERNS REVERSE, BANKS FORWARD

968 THIS MODE TESTS EACH PATTERN COMPLETELY WITH THE MOST  
969 COMPLEX ONES FIRST, WORKING TO THE SIMPLE ONES.

970 WHILE TESTING EACH PATTERN THE BANKS ARE RUN FROM 0 TO  
971 167 EXCEPT THOSE REQUIRING RELOCATION\*.

972  
973 MODE 7 = 'PARBAW' = PATTERNS REVERSE, BANKS WORST FIRST

974 THIS MODE TESTS EACH PATTERN COMPLETELY WITH THE MOST  
975 COMPLEX ONES FIRST, WORKING TO THE SIMPLE ONES.

976 WHILE TESTING EACH PATTERN FIRST EACH KNOWN BAD BANK  
977 FROM 0 TO 167 EXCEPT THOSE THAT REQUIRE RELOCATION\* IS  
978 RUN, THEN PRESUMED GOOD BANKS ARE RUN FROM 0 TO 167  
979 EXCEPT THOSE REQUIRING RELOCATION\*.

980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999

#### NOTE

\* RELOCATION IS REQUIRED TO TEST THE  
BANK(S) IN PROGRAM SPACE AND ALSO TO  
TEST ANY ECC BANKS PROTECTED BY  
DIAGNOSTIC CHECKMODE WITH THE INHIBIT  
MODE POINTER OFF (ZERO)!

1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040

#### 2.4.1.3 TEST MODE APPLICATIONS -

1. TO VERIFY CORRECT OPERATION OF THE MEMORY SYSTEM USE MODE 0 'BAFPAF'.

ADVANTAGES: EASY TO UNDERSTAND.

DISADVANTAGES: IN CASE OF A FAILING BANK, IT MAY TAKE A LONG TIME TO FIND THE FAILURE.

2. TO GET DETAILED ERROR INFORMATION ON KNOWN BAD BANKS (FOUND BY SIZING ROUTINE) USE MODE 2 'BAWPAF'.

ADVANTAGES: SEEKS BAD BANKS. EASY TO UNDERSTAND.

DISADVANTAGES: FAILURES OTHER THAN ZEROS ONES MAY TAKE A LONG TIME TO FIND.

3. TO GET GOOD ERROR INFO ON ANY MEMORY PROBLEM FAST USE MODE 4 'PAFBAF'.

ADVANTAGES: COVERS ALL BANKS FAST. EASY TO UNDERSTAND.

DISADVANTAGES: FAILURES FROM ONLY COMPLEX PATTERNS MAY TAKE A LONG TIME TO FIND.

4. TO FIND ANY PROBLEM FAST USE MODE 7 'PARBAW'.

ADVANTAGES: COVERS ALL BANKS FAST.

DISADVANTAGES: DIFFICULT TO UNDERSTAND FAILURES REPORTED ARE NOT NECESSARILY THE MOST BASIC FAILURE MODES.

1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092

#### 2.4.2 DISPLAY REGISTER -

A SOFTWARE DISPLAY REGISTER EXISTS IN LOCATION 174 IN ADDITION TO ANY HARDWARE DISPLAY EXISTENCE.

DISPLAY FIELDS ARE AS FOLLOWS:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
RELOCATED	BANK #						NOT USED			PATTERN #				
=====														

PATTERN # = THE NUMBER OF THE PATTERN PRESENTLY BEING RUN. ALL PATTERNS ARE DESCRIBED IN SECTION 6.2. ANY PATTERN CAN BE FOUND IN THE DIAGNOSTIC BY LOOKING UP THE SYMBOLIC TAGS 'MTOONN' AND 'MTPONN' - WHERE 'NN' IS THE TEST NUMBER. MTOONN REFERS TO THE ROUTINE THAT SETS UP FOR THE TEST PATTERN WHEREAS MTPONN IS THE ACTUAL PATTERN ITSELF.

#### NOTE

THE PATTERN # IS NOT NECESSARILY AN INDICATION OF DEGREE OF DIFFICULTY.

BANK = THE NUMBER OF THE BANK (16K) OF MEMORY UNDER TEST (0-167). THESE BITS DIRECTLY MAP TO PHYSICAL ADDRESS BITS (21:15).

RELOCATED = THIS BIT INDICATES THAT THE PROGRAM IS RELOCATED AND NO LONGER IN BANK 0. IT WILL BE RELOCATED TO THE FIRST KNOWN GOOD NON-PROTECTED MEMORY BANK INDICATED ON THE CONFIGURATION MAP (REFERENCE SECTION 7.3).

#### NOTE

ANOTHER WAY TO OBTAIN THIS INFORMATION IS TO TYPE A CONTROL/T AT THE CONSOLE (REFERENCE SECTION 2.4.4.5).



1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150

### 2.4.3 SPECIAL MEMORY LOCATIONS -

#### 2.4.3.1 CACHE CONSTANT -

THE CACHE CONSTANT IS LOCATED AT SYMBOLIC LOCATION "CACHK" AND IS USED TO ENABLE CACHE.

#### NOTE

BIT 0 IN THE CACHE CONSTANT HAS NO EFFECT SINCE IT IS UNCONDITIONALLY SET BY THE PROGRAM WHENEVER IT TRIES TO ENABLE CACHE.

#### 2.4.3.2 CONFIGURATION TABLE

THE CONFIGURATION TABLE IS LOCATED AT SYMBOLIC LOCATION "CONFIG" AND HAS THE FOLLOWING FORMAT:

CONFIG: FIRST 16K CONFIGURATION WORDS (2 EACH)  
2ND 16K CONFIGURATION WORDS (2 EACH)  
200TH 16K CONFIGURATION WORDS (2 EACH)

#### CONFIGURATION WORDS:

LOW:	BIT 0	ERRORS PRESENT
	BIT 1	MEMORY EXISTS
	BIT 2-4	RESERVED
	BIT 5	SKIP ECC LOGIC TESTS FLAG (1=SKIP)
	BIT 6	PROTECTED REGION OF AN ECC MEMORY
	BIT 7	PROTECTED (PROGRAM SPACE)
	BIT 8-11	CSR CODE
	BIT 12-15	INTERLEAVED CSR CODE
MED:	BIT 0-7	NUMBER OF ERRORS
	BIT 8-10	MEMORY TYPE
	BIT 11	CSR TESTED OK
	BIT 12	INTERLEAVE ENABLED
	BIT 13	"BACKGROUND PATTERN VALID" FLAG
	BIT 14	BANK SELECTED FOR TEST BY FIELD SERVICE MODE
	BIT 15	LOADERS HOME BANK

THIS TABLE IS USED AS THE SOURCE FOR THE CONFIGURATION MAP (REFERENCE. SECTION 7.3).

1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208

#### 2.4.4 TERMINAL COMMANDS -

##### 2.4.4.1 CONTROL 'C'

THIS COMMAND WILL:

1. IF SWITCH 8 (HALT PROGRAM) IN THE SWITCH REGISTER IS SET HALT THE PROGRAM.
2. IF SWITCH 8 IS NOT SET, UNRELOCATE IF PROGRAM WAS RELOCATED.
3. FLUSH OUT ANY DBE'S.
4. TURN OFF MEMORY MANAGEMENT.
5. ATTEMPT TO BOOT RK05 DRIVE 0.
6. FAILING 4, ATTEMPT TO BOOT RK04 DRIVE 1.
7. FAILING 5, GO TO 4.

THIS COMMAND WILL ONLY BE RECOGNIZED AT THE COMPLETION OF THE CURRENT TEST OR PATTERN, OR AT THE END OF A LINE OF AN ERROR MESSAGE.

##### 2.4.4.2 CONTROL 'K' (KILL ERROR PRINTOUT AND SKIP PATTERN)

THIS COMMAND WILL ALLOW YOU TO STOP AN ERROR PRINTOUT AND SKIP TO THE NEXT PATTERN. THIS IS HANDY, FOR EXAMPLE, WHEN YOU HAVE A WHOLE BANK FULL OF ERRORS, HAVE GOTTEN ENOUGH INFORMATION, AND WISH TO SKIP TO THE NEXT PATTERN.

##### 2.4.4.3 CONTROL 'T' (TELL ME WHAT'S HAPPENING)

THIS COMMAND WILL PRINT OUT THE INFORMATION ENCODED IN THE DISPLAY REGISTER. THIS IS MAINLY INTENDED FOR CPU'S WITHOUT A HARDWARE DISPLAY REGISTER.

EXAMPLE:

BANK = 17 TEST = 46  
RELOCATED BANK= 0 PAT= 26

BY USE OF FIELD SERVICE COMMAND 17 'TRACE' CAN BE SET SO THAT IT WILL AUTOMATICALLY TYPE OUT THE BANK AND PATTERN NUMBERS AS EACH PATTERN IS RUN. (REFERENCE SECTION 2.4.4.8.18).

1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254

#### 2.4.4.4 CONTROL 'S' (STOP)

THIS COMMAND WILL STOP TYPEOUT (SOON) AND WILL WAIT FOR A CONTROL 'Q'.

#### 2.4.4.5 CONTROL 'Q' (QUINTINUE)

THIS COMMAND WILL CONTINUE TYPING THAT HAS BEEN STOPPED BY CONTROL 'S'. IF THERE HAS BEEN NO CONTROL 'S' TYPED THEN THIS COMMAND IS IGNORED.

#### 2.4.4.6 CONTROL 'F' (FIELD SERVICE MODE)

THIS COMMAND WILL CAUSE YOU TO ENTER A MODE WHICH LOOKS FOR SUB COMMANDS.

WHEN THE PROGRAM IS LOOKING FOR A SUB COMMAND ANY NUMBER THAT IS NOT A LEGAL COMMAND WILL CAUSE A MINI HELP MESSAGE TO BE TYPED. THEREFORE WHEN IN DOUBT TYPE 99 (CR) AND YOU WILL GET HELP.

#### NOTE

TYPING JUST CARRIAGE RETURN IS A DEFAULT COMMAND 0.

#### 2.4.4.7.1 FIELD SERVICE COMMAND 0 (EXIT)

THIS COMMAND WILL EXIT FIELD SERVICES MODE AND RETURN TO WHATEVER TASK IT WAS IN PRIOR TO TYPING CONTROL 'F'. NOTE TYPING JUST CARRIAGE RETURN IS A DEFAULT COMMAND 0.

1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305

#### 2.4.4.7.2 FIELD SERVICE COMMAND 1 (READ CSR)

THIS COMMAND WILL TYPEOUT THE CONTENTS OF THE CSR.

IF THERE IS MORE THAN ONE CSR ON THE CPU (OR IF THE PROGRAM HAS NOT DETERMINED THE CSR STATUS YET), IT WILL ASK YOU 'WHICH CSR(0-F)' TO WHICH YOU MUST RESPOND WITH AN HEXIDECIMAL NUMBER FROM 0 TO F. NOTE TYPING JUST CARRIAGE RETURN IS A DEFAULT 0.

IF THE CSR YOU SELECT CAUSES A TRAP TO 4 THE PROGRAM WILL TYPE 'THIS CSR DOES NOT EXIST'.

#### NOTE

CSR REFERENCES ARE DONE IN ACCORDANCE WITH SECTION 5.0.

#### 2.4.4.7.3 FIELD SERVICE COMMAND 2 (LOAD CSR)

THIS COMMAND WILL ENABLE YOU TO LOAD THE CSR.

IF THERE IS MORE THAN ONE CSR ON THE CPU (OR IF THE PROGRAM HAS NOT YET DETERMINED THE CSR STATUS YET) IT WILL ASK YOU 'WHICH CSR(0-F)' TO WHICH YOU MUST RESPOND WITH AN HEXIDECIMAL NUMBER FROM 0 TO F. NOTE TYPING JUST CARRIAGE RETURN IS A DEFAULT 0.

IF THE CSR YOU SELECT CAUSES A TRAP TO 4 THE PROGRAM WILL TYPE 'THIS CSR DOES NOT EXIST'.

THE CSR WILL BE READ AND DISPLAYED AS IN COMMAND 1.

THE PROGRAM WILL THEN ASK YOU FOR THE 'CSR?' TO WHICH YOU MUST RESPOND WITH AN OCTAL NUMBER. NOTE TYPING JUST CARRIAGE RETURN IS A DEFAULT 0.

THE PROGRAM WILL THEN LOAD THE CSR AND READ IT AGAIN DISPLAYING ITS NEW CONTENTS.

1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357

#### 2.4.4.7.4 FIELD SERVICE COMMAND 3 (EXAMINE MEMORY)

THIS COMMAND WILL ALLOW YOU TO EXAMINE ANY PHYSICAL ADDRESS AND DOES THE NECESSARY MEMORY MANAGEMENT MAPPING FOR YOU.

THE PROGRAM WILL ASK YOU FOR THE 'PHYSICAL ADDRESS (0-17757776)' TO WHICH YOU MUST RESPOND WITH AN OCTAL NUMBER.

IF THE ADDRESS ACCESS CAUSES A TRAP TO 4 THE PROGRAM WILL TYPE 'TIMEOUT TRAP'. IF THE ADDRESS ACCESS CAUSES A TRAP TO 114 THE PROGRAM WILL TYPE 'PARITY ABORT'.

THE CONTENTS OF YOUR PHYSICAL ADDRESS WILL BE TYPED.

#### 2.4.4.7.5 FIELD SERVICE COMMAND 4 (MODIFY MEMORY)

THIS COMMAND ALLOWS YOU TO MODIFY ANY PHYSICAL ADDRESS AND DOES THE NECESSARY MEMORY MANAGEMENT MAPPING FOR YOU.

THE PROGRAM WILL ASK YOU FOR THE 'PHYSICAL ADDRESS (0-17757776)' TO WHICH YOU MUST RESPOND WITH AN OCTAL NUMBER.

IF THE ADDRESS ACCESS CAUSES A TRAP TO 4 THE PROGRAM WILL TYPE 'TIMEOUT TRAP'. IF THE ADDRESS ACCESS CAUSES A TRAP TO 114 THE PROGRAM WILL TYPE 'PARITY ABORT'.

THE PROGRAM WILL TYPE 'OLD DATA WAS' AND THE CONTENTS OF YOUR PHYSICAL ADDRESS.

THE PROGRAM WILL THEN TYPE 'INPUT NEW DATA' TO WHICH YOU MUST RESPOND WITH AN OCTAL NUMBER. NOTE TYPING JUST CARRIAGE RETURN IS A DEFAULT 0.

THE PROGRAM WILL ATTEMPT TO WRITE THIS NEW DATA INTO YOUR PHYSICAL ADDRESS AFTER WHICH IT WILL READ IT AGAIN AND TYPE 'DATA IS NOW' AND THE NEW CONTENTS OF YOUR PHYSICAL ADDRESS.

#### NOTE

IF YOU CAN'T CHANGE THE DATA, THAT WOULD INDICATE THAT YOU HAVE A DOUBLE BIT ERROR IN THAT DOUBLE WORD PAIR.



1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408

#### 2.4.4.7.6 FIELD SERVICE COMMAND 5 (SELECT BANK TEST)

THIS COMMAND ALLOWS YOU TO RUN ANY BANK WITH ANY PATTERN FOREVER.

THE PROGRAM WILL ASK YOU 'BANK(0-167)' TO WHICH YOU MUST RESPOND WITH AN OCTAL NUMBER. IF THE BANK IS NOT ACCESSIBLE. THE PROGRAM WILL TYPE 'BANK NOT ACCESSIBLE' AND ASK QUESTION OVER.

THE PROGRAM WILL THEN ASK 'TEST (0-47)' TO WHICH YOU MUST RESPOND WITH AN OCTAL NUMBER.

#### NOTE

ANY PATTERN CAN BE RUN INCLUDING THOSE THAT ARE NOT PART OF THE APT E-TABLE DEFAULTS (REFERENCE SECTION 6.2.1). IF YOU SELECT PATTERN 0, THE PROGRAM WILL ASK 'TEST 0 DATA IS?' TO WHICH YOU MUST RESPOND WITH AN OCTAL NUMBER.

IF THE BANK YOU SELECTED REQUIRES RELOCATION THE PROGRAM WILL TYPE 'BANK REQUIRES RELOCATION' AND EXIT THIS COMMAND. NOTE NORMALLY THIS IS TRUE FOR BANK 0.

THE PROGRAM WILL THEN ATTEND THE CONSOLE KEYBOARD FOR INTERRUPTS AND TYPE 'TO ESCAPE TYPE ANY KEY!'

THE TEST PATTERN WILL BE ENTERED AND RUN UNTIL A CONSOLE KEY IS DEPRESSED TO ESCAPE THIS LOOP.

#### 2.4.4.7.7 FIELD SERVICE COMMAND 6 (TYPE CONFIGURATION MAP)

THIS COMMAND TYPES THE CONFIGURATION MAP.

THIS IS USEFUL AFTER A LONG RUN (OVERNIGHT) TO SEE ALL THE BANKS THAT ARE MARKED AS BAD. (ESPECIALLY IF YOUR CONSOLE IS A VIDEO TERMINAL).

FOR A DETAILED EXPLANATION OF THE MAP REFERENCE SECTION 7.3.

1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462

#### 2.4.4.7.8 FIELD SERVICE COMMAND 7 (SOB-A-LONG TEST)

THIS COMMAND ALLOWS EXECUTION OF THE SOB-A-LONG TEST ON ALL NON-PROTECTED BANKS REFERENCE SECTION 6.2.2.26. OPERATION IS IDENTICAL TO COMMAND 5 EXCEPT THAT NO PATTERN OR BANK IS ENTERED AND EACH PASS CAUSES A BELL.

#### 2.4.4.7.9 FIELD SERVICE COMMAND 8 (ERROR SUMMARY)

THIS COMMAND TYPES OUT THE NUMBER OF PASSES AND THE TOTAL NUMBER OF ERRORS. IF THERE WERE 7 ERRORS IT WILL TYPE OUT THE BANKS AND THE NUMBER OF ERRORS PER BANK UP TO 255 DECIMAL.

THIS BECOMES USEFUL AFTER LONG RUNS (ALL NIGHT) ON SYSTEMS WITH A VIDEO CONSOLE TERMINAL.

#### 2.4.4.7.10 FIELD SERVICE COMMAND 9 (REFRESH TEST)

THIS COMMAND ALLOWS EXECUTION OF THE REFRESH TEST ON ALL NON-PROTECTED BANKS REFERENCE SECTION 6.2.2.19. OPERATION IS IDENTICAL TO COMMAND 5 EXCEPT THAT NO PATTERN OR BANK IS ENTERED AND EACH PASS CAUSES A BELL.

#### 2.4.4.7.11 FIELD SERVICE COMMAND 10 (SET FILL COUNT)

THIS COMMAND ALLOWS SETTING OF THE TERMINAL FILL COUNT (NECESSARY FOR LA30'S, ASR3'S, AND VT05'S). IT IS NORMALLY SET TO ZERO FOR LA36'S, VT52'S, VT100'S, ETC.

#### 2.4.4.7.12 FIELD SERVICE COMMAND 11 (ENTER KAMIKAZE MODE)

THIS COMMAND ALLOWS YOU TO RUN PATTERNS THAT ARE NORMALLY NOT EXECUTED UNLESS UNDER APT OR ACT. THEY ARE USUALLY VERY TIME CONSUMING AND CAN RESULT IN FAILURES THAT ARE FATAL TO THE PROGRAM. IN EFFECT YOU ARE TRYING TO FIND A HARDWARE FAILURE REGARDLESS OF THE CONSEQUENCES. NOTE THAT MOST CRASHES DO NOT WIPE OUT THE DISPLAY INFORMATION WHICH IS TELLING YOU WHAT THE PROGRAM WAS DOING JUST PRIOR TO FAILURE. THERE ARE TWO WAYS TO DIE HERE - IMPATIENCE AND CRASHES.

1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519

#### 2.4.4.7.13 FIELD SERVICE COMMAND 12 (EXIT KAMIKAZE MODE)

RETURN TO THE DEFAULT MODE OF TESTING (UNDO COMMAND 12).

#### 2.4.4.7.14 FIELD SERVICE COMMAND 13 (TURN CACHE OFF)

THIS CHANGES THE CACHE CONSTANT TO BYPASS CACHE (REFERENCE SECTION 2.4.3.1).

#### 2.4.4.8.15 FIELD SERVICE COMMAND 14 (TURN CACHE ON)

THIS CHANGES THE CACHE CONSTANT TO USE CACHE (REFERENCE SECTION 2.4.3.1).

#### 2.4.4.7.16 FIELD SERVICE COMMAND 15 (TEST ONLY SELECTED BANKS)

THIS COMMAND ALLOWS YOU TO CENTER THE TEST EFFORT ON ONLY THOSE BANKS THAT YOU ARE TROUBLESHOOTING. YOU MAY ALSO TEST BANKS THAT REQUIRE RELOCATION AND WERE INACCESSABLE VIA COMMAND 5.

#### 2.4.4.7.17 FIELD SERVICE COMMAND 16 (RESUME TESTING ALL BANKS)

RETURN TO THE DEFAULT MO OF TESTING (UNDO COMMAND 15).

#### 2.4.4.7.18 FIELD SERVICE COMMAND 17 (RESUME TESTING ALL BANKS)

ENABLE 'TRACE'. AFTER EXITING FIELD SERVICE MODE, THE PROGRAM WILL TYPE OUT THE BANK AND PATTERN NUMBERS AS EACH PATTERN IS RUN.

#### 2.4.4.7.19 FIELD SERVICE COMMAND 18 (RESUME TESTING ALL BANKS)

DISABLE 'TRACE'. (UNDO COMMAND 17).

1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577

## 2.5 EXECUTION TIMES

### 2.5.1 TYPICAL (SYSTEM) -

EXECUTION TIME DEPENDS ON MANY VARIABLES; HOWEVER HERE ARE SOME MEASURED TIMES ON AN 11/44 WITH CACHE:

#### 128K WORDS OF MS11-L MEMORY

NORMAL PASS	0 MIN	50 SEC
QUICK VERIFY	0 MIN	50 SEC
KAMIKAZE MODE	10 MIN	5 SEC
KAMIKAZE QV	10 MIN	5 SEC

#### 128K WORDS OF MS11-M MEMORY (NON-INTERLEAVED)

NORMAL PASS	2 MIN	25 SEC
QUICK VERIFY	1 MIN	0 SEC
KAMIKAZE MODE	11 MIN	0 SEC
KAMIKAZE QV	10 MIN	30 SEC

#### 128K WORDS OF MS11-M MEMORY (INTERLEAVED)

NORMAL PASS	3 MIN	55 SEC
QUICK VERIFY	1 MIN	50 SEC
KAMIKAZE MODE	22 MIN	0 SEC
KAMIKAZE QV	20 MIN	5 SEC

#### 512K WORDS OF MS11-P MEMORY

NORMAL PASS	3 MIN	55 SEC
QUICK VERIFY	3 MIN	25 SEC
KAMIKAZE MODE	42 MIN	30 SEC
KAMIKAZE QV	37 MIN	0 SEC

### 2.5.2 CALCULATIONS (SYSTEM)

#### NORMAL PASS

ADD	18 SEC PER 16K BANK OF NON-INTEREAVED MS11-M
ADD	15 SEC PER 16K BANK OF INTERLEAVED MS11-M
ADD	6 SEC PER 16K BANK OF MS11-L
ADD	22 SEC PER 64K BANK OF MS11-P

#### QUICK VERIFY PASS

ADD	8 SEC PER 16K BANK OF NON-INTERLEAVED MS11-M
ADD	7 SEC PER 16K BANK OF INTERLEAVED MS11-M
ADD	6 SEC PER 16K BANK OF MS11-L
ADD	20 SEC PER 64K BANK OF MS11-P

#### KAMIKAZE MODE

ADD 10 MIN. PER 128K WORDS FOR APPROXIMATE PASS TIMES.

## 2.5.3 TYPICAL (TESTS)

TEST TIME  
-----DESCRIPTION  
-----

MT0000	:<1 SEC	DATA PATTERN TEST
MT0001	:<1 SEC	ADDRESS TEST
MT0002	:<1 SEC	COMPLEMENT ADDRESS TEST
MT0003	: 1 SEC	3 XOR 9 WORST CASE NOISE TEST
MT0004	: 1 SEC	ROTATING ZEROS TEST
MT0005	: 1 SEC	ROTATING ONES TEST
MT0006	:<1 SEC	INITIAL DATA TEST
MT0007	:<1 SEC	ADDRESS BIT TEST
MT0010	:<1 SEC	BYTE ADDRESSING TEST
MT0011	:<2 SEC	CREATE SINGLE BIT ERROR TEST
MT0012	:<1 SEC	WRITE BYTE CLEARS SBE TEST
MT0013	: 1 SEC	CREATE DOUBLE BIT ERROR TEST
MT0014	: 1 SEC	BASIC DOUBLE BIT ERROR TEST
MT0015	: 1 SEC	WRITE INHIBIT OF BYTE WITH DBE
MT0016	:<1 SEC	WRITE INHIBIT OF WORD WITH DBE
MT0017	:<1 SEC	HOLDING 1'S 0'S TEST
MT0020	: 1 SEC	SYNDROMES TO CSR ON SINGLE BIT ERROR TEST
MT0021	: 1 SEC	MARCHING 0'S 1'S TEST
MT0022	:10 SEC	REFRESH TEST
MT0023	:10 SEC	SHIFTING DIAGONAL TEST
MT0024	:20 SEC	FAST GALLOPING PATTERN TEST
MT0025	:<1 SEC	INTERRUPT ENABLE TEST
MT0026	:<1 SEC	RANDOM DATA TEST
MT0027	: 1 SEC	UNIQUE BANK TEST
MT0030	: 1 SEC	FLUSH OUT DBE'S TEST
MT0031	: 3 SEC	SOB-A-LONG TEST
MT0032	:<1 SEC	WRITE RECOVERY TEST
MT0033	:35 SEC	BRANCH GOBBLE TEST
MT0034	:<1 SEC	SOFT ERROR TEST
MT0035	:<1 SEC	WORST CASE PARITY TEST
MT0036	: 1 SEC	CORRECTION CORE TEST
MT0037	:<1 SEC	CHECK ECC DISABLE TEST
MT0041	: 1 SEC	ADDRESS TO CSR ON DBC TEST
MT0042	:<1 SEC	EXTENDED ADDRESS TO CSR ON ERROR TEST
MT0043	:<1 SEC	WRITE BYTE TEST
MT0044	: 1 SEC	SHIFTING CHECKBITS THROUGH CSR TEST
MT0045	:<1 SEC	SYNDROME BITS TO THE CSR ON A DBE TEST
MT0046	: 1 SEC	CHECK SINGLE BIT ERRORS WITH ECC DISABLED TEST
MT0047	:<1 SEC	NO CSR UPDATE WITH EXISTING DBE TEST



## 3.0 ERROR INFORMATION

## 3.1 ERROR REPORTING

MOST ERRORS ARE REPORTED USING THE EMT TRAP AND HANDLER PROVIDED BY SYSMAC.SML. MOST ERRORS WILL BE OF THE 'MEMORY DATA ERROR' TYPE WHICH WILL BE DESCRIBED HERE. MEMORY DATA ERRORS WILL ALSO CAUSE THE BANK TO BE MARKED AS BAD IN THE CONFIGURATION TABLE.

OTHER ERRORS ARE BEST EXPLAINED BY REFERENCING THE SPECIFIC TYPEOUT AND IF NECESSARY THE PROGRAM LISTING.

## EXAMPLE 1:

## MEMORY DATA ERROR

PC	BANK	VADD	PADD	GOOD	BAD	XOR	CSR	MTYP	INT	PAT
022132	37	060006	03700006	000000	000100	000100	0		-	06
022132	37	060006	03700006	000000	000100	000100	0		-	06
022132	37	060006	03700006	000000	000100	000100	0		-	06
022132	37	060006	03700006	000000	000100	000100	0		-	06

WHILE TESTING BANK 37 AT VIRTUAL ADDRESS 60006 (VIRTUAL ADDRESSES ARE ALWAYS BETWEEN 60000 AND 157776 FOR MAPPING PURPOSES), PHYSICAL ADDRESS 3700006 (THAT'S BANK 37 PHYSICAL 6 WITHIN THE BANK) WITH PATTERN 6 (INITIAL DATA TEST), THE GOOD DATA EXPECTED WAS 0 BUT THE DATA ACTUALLY READ (BAD) WAS 100, THE EXCLUSIVE OR AT GOOD BAD YIELDS 100 WHICH INDICATES ONLY FAILING BIT(S) (BIT 6). IT IS AN MS11-P (ECC) MEMORY AND IT'S NOT INTERLEAVED. THE CSR IS LOCATED AT 172000.

## EXAMPLE 2:

## MEMORY DATA ERROR

PC	BANK	VADD	PADD	GOOD	BAD	XOR	CSR	MTYP	INT	PAT
022132	35	060000	03500000	000000	000001	000001	0	M	1	06
022132	35	060002	03500002	000000	000100	000100	0	M	1	06
022132	35	060006	03500006	000000	000100	000100	0	M	1	06

WHILE TESTING BANK 35, VIRTUAL ADDRESS 60000, PHYSICAL ADDRESS 3700000 WITH PATTERN 6 (INITIAL DATA TEST), THE GOOD DATA EXPECTED WAS 0 BUT THE DATA ACTUALLY READ (BAD) WAS 1, THE EXCLUSIVE OR AT GOOD BAD YIELDS 1 WHICH INDICATES ONLY FAILING BIT(S) (BIT 0). IT IS AN MS11-M (ECC) MEMORY AND IT'S INTERLEAVED; SO SINCE ADDRESS BIT 1 WAS NOT ASSERTED, THE CSR IS LOCATED AT 172000.

WHILE ALSO IN BANK 35, VIRTUAL ADDRESSES 60002 AND 60006 WERE EXPECTED TO HAVE 0, BUT THE DATA READ WAS 100, THE EXCLUSIVE OR OF GOOD BAD YIELDS 100 WHICH INDICATES ONE FAILING BIT (BIT 6). SINCE IT IS INTERLEAVED MS11-M MEMORY, AND ADDRESS BIT 1 IS ASSERTED, THE CSR IS LOCATED AT 172102 (CSR NUMBER 1 UNDER THE INT COLUMN)

## NOTE

SUBSEQUENT ERRORS OF THE SAME TEST DO NOT TYPE A NEW HEADING.

## 3.2 ERROR ABBREVIATIONS

THE FOLLOWING IS A LIST OF ALL ABBREVIATIONS USED IN ERROR REPORTS.

# OF ERRORS	NUMBER OF ERRORS THAT WERE DETECTED.
1ST ADD	FIRST ADDRESS THAT FAILED.
ARRAY	THE ARRAY NUMBER THAT WAS LOCKED UP IN THE MS11-M CSR.
APT#	THE # OF CPU'S APT EXPECTS ON THE SYSTEM.
APTCORE	APT CORE SIZE.
APTMOS	APT MOS SIZE.
BAD	BAD DATA.
BAD-WD1	BAD WORD #1 OF A DOUBLE WORD DATA VALUE.
BAD-WD2	BAD WORD #2 OF A DOUBLE WORD DATA VALUE.
BAD-CHK	BAD CHECK CODE BITS.
BANK	THE BANK NUMBER. BANKS ARE 16K WORDS LONG.
BD-CC	BAD CHECK CODE BITS.
CHKBITS	THE 7 BIT VALUE OF THE CHECK CODE BITS.
CONTRL	THE CACHE CONTROL REGISTER.
CPUERR	CPU ERROR REGISTER.
CSR	CONTROL AND STATUS REGISTER.
CSRNO	CSR NUMBER (0-F HEXIDEcimal).
DATARG	THE CACHE DATA REGISTER.
DBE	DOUBLE BIT ERROR (UNCORRECTABLE ERROR).
DEV ADD	DEVICE ADDRESS.
ECC	ERROR CORRECTABLE CODE.
GD-CC	GOOD CHECK CODE BITS.
GD-CHK	GOOD CHECK CODE BITS.
GD-WD1	GOOD WORD #1 OF A DOUBLE WORD DATA VALUE.
GD-WD2	GOOD WORD #2 OF A DOUBLE WORD DATA VALUE.
GOOD	GOOD DATA.
INT	INTERLEAVED (ADDRESS BIT 1 ASSERTED) CSR NUMBER.
LSIZE	MS11-L SIZE.
MEMERR	MEMORY ERROR REGISTER.
MMR0	MEMORY MANAGEMENT REGISTER #0.
MMR1	MEMORY MANAGEMENT REGISTER #1.
MMR2	MEMORY MANAGEMENT REGISTER #2.
MMR3	MEMORY MANAGEMENT REGISTER #3.
MSIZE	MS11-M SIZE.
MTYP	MEMORY TYPE (MS11-L, MS11-M, OR MS11-P).
PADD	PHYSICAL ADDRESS (ASSERTED BY THE PROGRAM AFTER MAPPING).
PAT	PATTERN NUMBER.
PC	PROGRAM COUNTER AT THE TIME THE ERROR OCCURRED.
SBE	SINGLE BIT ERROR (CORRECTABLE ERROR).
VADD	VIRTUAL ADDRESS (ASSERTED BY THE PROGRAM BEFORE MAPPING).
WROTE1	THE DATA THAT WAS WRITTEN INTO THE 1ST HALF OF A DOUBLE WORD.
WROTE2	THE DATA THAT WAS WRITTEN INTO THE 2ND HALF OF A DOUBLE WORD.
XOR	EXCLUSIVE OR OF THE GOOD AND BAD DATA. SHOWS THE BAD BITS.
AUT	ADDRESS UNDER TEST

1737  
173E  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760

### 3.3 ERROR HALTS

THERE ARE SEVERAL HALTS IN THE PROGRAM.

ALL UNUSED TRAP VECTORS CONTAIN A TRAP CATCHER (.WORD .+2,HALT).

AN UNDEFINED TRAP INSTRUCTION HALTS AT SYMBOLIC LOCATION '\$HALT2'.

THE APT DOWN LOAD SEQUENCE WILL HALT AT SYMBOLIC LOCATION 'APTHLT'.

HALT ON ERROR OPTION (SW15 SET) AT SYMBOLIC LOCATION '\$HALT'.

HALT PROGRAM (SW8 SET) AT SYMBOLIC LOCATION '\$EXHALT'.

POWER FAIL WILL NORMALLY HALT AT THE END OF THE SHUT DOWN SEQUENCE (SYMBOLIC LOCATION '\$DOWN').

POWER FAIL HAS A FATAL HALT AT SYMBOLIC LOCATION '\$ILLUP' WHICH CAN BE CAUSED BY POWER UP OCCURRING BEFORE POWER DOWN SEQUENCE COMPLETED OR BY POWER DOWN BEFORE A POWER UP SEQUENCE IS COMPLETED.

1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813

#### 4.0 PROGRESS REPORTS

PASS COMPLETE TYPEOUTS AS FOLLOWS:

END PASS	#	0
END PASS	#	1
END PASS	#QV	2

#### NOTE

PASS 2 WAS FLAGGED AS A QUICK VERIFY  
PASS. (BECAUSE OF A CHANGE IN SW5)

TO OBTAIN PROGRESS REPORTS WHILE EXECUTING, TYPING A CONTROL ';' WILL  
PRINT OUT THE INFORMATION ENCODED IN THE DISPLAY REGISTER.

EXAMPLE:

BANK= 2 TEST= 34

REFERENCE SECTION 2.4.4.7.18 FOR MORE INFORMATION ON TRACING.

#### 5.0 CSR INFORMATION TABLES

THE FOLLOWING IS A PICTURE VIEW OF THE CURRENT CONTROL STATUS  
REGISTERS WHICH CAN BE TESTED BY THIS PROGRAM. IT SHOWS BIT  
ASSIGNMENTS AND DEFINITIONS TO PROVIDE A HANDY REFERENCE, AND  
SHOWS THE SIMILARITIES AND DIFFERENCES BETWEEN EACH ONE:

#### NOTE

ALL UNUSED BITS IN EACH CSR ARE EQUAL TO ZERO.

1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868

## 5.1 MS11-P CSR

(I)

```

-----
I I I I I I I I I I I I I I I I
DE EA ADDRESS SE IP DC EC EE
I I I I I I I I I I I I I I I I
-----
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00

```

(II)

```

-----
I I I I I I I I I I I I I I I I
DE EA SI CHECK BITS SE IP DC EC EE
I I I I I I I I I I I I I I I I
-----
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00

```

(III)

```

-----
I I I I I I I I I I I I I I I I
DE EA SI SYNDROMES SE IP DC EC EE
I I I I I I I I I I I I I I I I
-----
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00

```

BIT ASSIGNMENTS ARE DEFINED  
AS FOLLOWS:

BIT15 UNCORRECTABLE ERROR  
ON A READ TO MEMORY (ECC DISABLE  
BIT = 0), THIS BIT IS SET IF  
A DOUBLE ERROR OCCURS. THE  
ERROR ADDRESS IS STORED IN THE  
CSR. SETTING THIS BIT ALSO  
TURNS ON A RED LED AT THE REAR OF  
THE CARD FOR A VISUAL INDICATION.  
THIS BIT IS ALSO SET IN ECC  
DISABLE MODE IF A SERR OR DERR  
OCCURS.

BIT14 EUB ERROR ADDRESS  
WITH BIT 14 = 1, A READ TO THE CSR  
WILL FETCH ADDRESS A21  
THROUGH A18. WHEN BIT 14 = 1,  
DIAGNOSTIC DATA MAY NOT BE  
LOADED INTO THE SYNDROME  
REGISTER.



1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923

BIT13 SET INHIBIT MODE  
WHEN THIS BIT IS SET TO  
A '1', IT ENABLES THE INHIBIT  
MODE POINTER TO INHIBIT  
EITHER THE FIRST OR SECOND  
16K FROM EVER GOING INTO THE  
DIAG CHECK OR ECC DISABLE  
MODE.

BITS05-10 CHECK BIT STORAGE (CSR 11)  
CHECK BIT STORAGE (DIAG CK  
BIT 2 = 1)  
WHEN IN THE DIAGNOSTIC CHECK  
MODE THESE BITS ARE USED TO STORE  
THE CHECK BITS TO BE WRITTEN  
INTO MEMORY OR THE CHECK BITS  
READ FROM MEMORY. IF A DOUBLE  
ERROR OR SINGLE ERROR OCCURS  
WHEN IN THE DIAGNOSTIC CHECK  
MODE AND ECC DISABLE BIT 1 = 0,  
THEN THE CHECK BITS ARE STORED  
IN THE CSR TOGETHER WITH  
THE DOUBLE OR SINGLE ERROR  
BIT. THESE BITS ARE WRITEABLE  
IN DIAGNOSTIC MODE. A '1'  
IS STORED IN BIT 11 IF CSR  
02, CSR 13, AND CSR 14 ARE  
SET TO INDICATE THAT THE  
MEMORY UNDER TEST IS A MS11-P.

BITS05-11 UNIBUS ADDRESS STORAGE (CSR 1)  
(DIAG CK BITS 2 = 0, ECC  
DISABLE BIT 1 = 0)

IF A DOUBLE OR SINGLE ERROR  
OCCURS ON A READ CYCLE, THEN  
ADDRESS BITS ALL THROUGH  
A17 ARE STORED IN THESE BITS  
THESE BITS ARE READ ONLY ON  
THE CONDITION THAT SERR (CSR 4)  
OR DERR (CSR 15) IS SET BUT  
CSR 14 IS NOT SET.

EUB ADDRESS STORAGE (DIAG CK  
BIT 2 = 0), ECC DISABLE BIT  
1 = 0 OR 1).

IF A DOUBLE OR SINGLE ERROR  
OCCURS ON A READ CYCLE,  
ADDRESS BITS A17 THROUGH  
A11 ARE STORED IN CSR BITS  
11 THROUGH 5 AND ADDRESS  
BITS A21 THROUGH A18 ARE

1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978

STORED IN A BACKUP REGISTER.  
THE EUB ERROR ADDRESS  
RETRIEVAL BIT (CSR 14) IS  
USED TO OBTAIN THE TOTAL  
ERROR ADDRESS AS FOLLOWS:

WITH CSR BIT 14 = 0 A READ TO  
THE CSR WILL OBTAIN A17  
THROUGH A11 FROM CSR BITS 11  
THROUGH 5.

CSR BIT 14 CAN THEN BE SET  
TO A '1' AND A READ TO THE  
CSR WILL THEN READ A21  
THROUGH A18 FROM CSR BITS 8  
THROUGH 5 AND 0'S FROM CSR  
BITS 11 THROUGH 9.

ADDRESS BITS A21 THROUGH  
A11 ARE OBTAINED  
TO LOCATE THE DOUBLE  
ERROR TO A 1K SEGMENT OF  
MEMORY.

THE EUB ADDRESS A21  
THROUGH A18 IS READ ONLY  
WHENEVER CSR 14 = 1.

BIT05-10 SYNDROME STORAGE (CSR 111)  
IF A DOUBLE OR SINGLE ERROR  
OCCURS ON A READ OR WRITE  
BYTE CYCLE, AND IF CSR BIT  
2 IS SET TO A '0' SYNDROME  
BITS X, 0, 1, 2, 4 AND 8  
AND STORED IN CSR BITS 5  
THROUGH 10. TO READ THE  
SYNDROME BITS FROM CSR, BIT  
YOU MUST READ THE ERROR  
ADDRESS, THEN SET 2 OF  
THE CSR MUST BE SET TO  
A '1' (DIAGNOSTIC MODE) AND  
THE CSR READ AGAIN. THIS OPERATION  
WILL ALLOW SYNDROME BITS  
FOR A SINGLE OR DOUBLE  
FAILURE TO BE READ INSTEAD  
OF THE ADDRESS BITS NORMALLY  
READ WHEN CSR 02 IS SET TO '0'.

BIT04 SINGLE ERROR  
IF ON A READ TO MEMORY A  
SBE OCCURS, THE ERROR  
ADDRESS A21-A11 AND  
THE ERROR SYNDROMES WILL

1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032

BE LOGGED IN CSR BITS 5-11  
UNLESS THE UNCORRECTABLE ERROR  
CSR 15 IS SET. THE ERROR  
ADDRESS WILL BE LOGGED  
UNCONDITIONALLY IN THE ECC  
DISABLE MODE. THIS BIT IS NOT  
SET IF INHIBIT MODE (BIT  
13 = 1) IS SET AND DIAGNOSTIC  
MODE (BIT 02 = 1) IS SET.

BIT03 INHIBIT MODE POINTER  
THIS BIT WORKS IN CONJUNCTION  
WITH THE SET INHIBIT MODE  
(BIT 13). WHEN BIT 13 IS SET  
TO A 1, A 16K PORTION OF  
MEMORY IS INHIBITED FROM  
OPERATING IN THE ECC DISABLE  
MODE OR DIAGNOSTIC CHECK MODE.

THE INHIBIT MODE POINTER  
INDICATES WHICH 16K IS BEING  
INHIBITED, I.E., BIT 3 = 0  
THE FIRST 16K OF MEMORY IS  
INHIBITED, BIT 3 = 1, THE  
SECOND 16K OF MEMORY IS  
INHIBITED.

WITH BIT 13 SET TO A 0, BIT  
3 BECOMES INOPERATIVE.

BIT03, IN CONJUNCTION WITH  
BIT 13, THEREFORE ALLOWS A 16K  
CHUNK OF MEMORY TO ALWAYS HAVE  
ECC COVERAGE. THE SYSTEMS  
DIAGNOSTIC CAN THEREFORE  
RESIDE IN THIS PROTECTED  
PORTION OF MEMORY AND CAN  
DISABLE ECC AND/OR RUN THE  
DIAGNOSTIC CHECK MODE IN THE  
REST OF MEMORY WITHOUT ITSELF  
BECOMING VULNERABLE TO SINGLE  
ERROR. THIS BIT IS A READ/WRITE  
BIT RESET BY POWER UP AND BUS  
INIT.

BIT02 DIAGNOSTIC CHECK MODE  
THIS MODE ALLOWS A MEANS OF  
FORCING A SINGLE OR DOUBLE  
ERROR IN A DESIRED LOCATION.  
IT ALSO PROVIDES A MEANS OF  
EXAMINING THE CHECK BITS AND  
THE SYNDROME IN A GIVEN  
LOCATION.

2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086

THE CHECK BITS DESIRED FOR A GIVEN DATA PATTERN ARE WRITTEN INTO BITS 5 THROUGH 11 OF THE CSR. A WORD OR WRITE BYTE MEMORY WILL WRITE THE CHECK BITS FROM THE CSR TO THE MOS ARRAY (CSR 2 = 1) INSTEAD OF THE CHECK BITS GENERATED ON THE DATA TO BE WRITTEN. SINGLE ERRORS ON THE READ PORTION OF THE DATOB CYCLE ARE CORRECTED.

A READ TO THE MEMORY WILL READ THE CHECK BITS STORED IN MEMORY AND CLOCK THEM INTO THE CSR.

IF A DOUBLE ERROR OR SINGLE ERROR OCCURS THE DERR OR SERR BIT IN THE CSR IS SET AND THE ERROR SYNDROME BITS READ FROM ECC ARE STORED IN CSR BITS 10-5 AS WELL AS THE ADDRESS BITS. IN DIAGNOSTIC CHECK MODE THE ERROR SYNDROME BITS WILL BE READ WHEN CSR BITS 10-5 ARE READ.

THIS BIT IS A READ/WRITE BIT AND IS RESET ON POWER UP AND BUS INIT.

BIT01 DISABLE CORRECTION MODE  
IF THIS BIT IS SET, NO SINGLE ERRORS WILL BE CORRECTED. A SINGLE ERROR WILL SET CSR 4 AND CSR 15 OR A DOUBLE ERROR WILL SET CSR 15 AND ASSERT BUS PBL IF CSR 00 IS ASSERTED. THE 1K BLOCK OF ADDRESS WHERE THE ERROR OCCURS WILL ALSO BE STORED IN THE CSR. THE PRIORITY OF A SERR AND DERR WILL BE THE SAME, I.E., THE LAST ERROR INFORMATION WILL ALWAYS BE STORED UNLESS A DERR PRECEDES A SERR. IF A DOUBLE ERROR OCCURS DURING A WRITE BYTE CYCLE, THE WRITE PORTION OF THE CYCLE WILL NOT BE ABORTED. THE CHECK BITS WRITTEN WILL

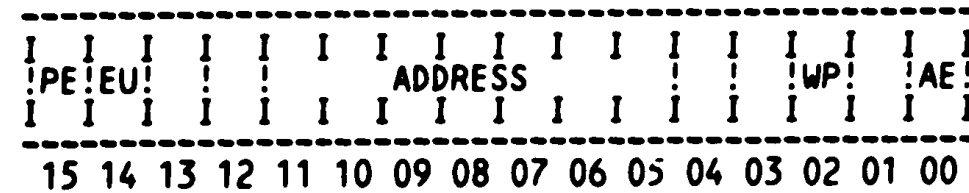
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111

HAVE BEEN GENERATED ON THE  
DATA WRITTEN. THIS MEANS THAT  
IF A SINGLE OR DOUBLE ERROR  
EXISTED IN THE LOCATION ACCESSED,  
IT WOULD BE CLEARED (UNLESS THE  
ERRORS WERE HARD).

THIS BIT IS A DIAGNOSTIC AID  
TO ALLOW WRITING AND READING  
DATA FROM MEMORY WITHOUT  
INTERFERENCE FROM THE ERROR  
CORRECTION LOGIC.

BIT00 UNCORRECTABLE ERROR  
INDICATION ENABLE  
IF A DOUBLE ERROR OCCURS WITH  
ECC ENABLED OR A SINGLE ERROR  
OR DOUBLE ERROR WITH ECC  
DISABLED, ON A READ CYCLE TO THE  
MEMORY AND THIS BIT IS SET,  
THEN BUS PBL WILL BE ASSERTED.

## 5.2 MS11-L CSR



BIT ASSIGNMENTS ARE DEFINED AS FOLLOWS:

## BIT15 PARITY ERROR

BIT14 EUB ERROR RETRIEVAL IF THE MEMORY IS ON AN EXTENDED UNIBUS, WHEN BIT14 IS ZERO, THE LOW ORDER FAILING ADDRESSES ARE AVAILABLE (BITS 11-17); WHEN BIT14 IS ONE, THE HIGH ORDER FAILING ADDRESSES ARE AVAILABLE (BITS 18-21 OF ADDRESS). IF THE MEMORY IS ON A UNIBUS, A JUMPER DISABLES THIS BIT SO THAT IT IS READ ONLY, AND EQUAL TO ZERO.

BITS 11-5 ERROR ADDRESS WITH BIT14 SET, THEY CONTAIN THE HIGH ORDER PARITY ERROR ADDRESS (BITS 21-18 OF ADDRESS); WITH BIT14 CLEARED, THEY CONTAIN THE LOW ORDER PARITY ERROR ADDRESS (BITS 17-11 OF ADDRESS).

BIT02 WRITE WRONG PARITY NORMAL PARITY (ODD) WHEN CLEAR; OTHER PARITY (EVEN) WHEN SET.

BIT00 ACTION ENABLE NO ACTION WHEN CLEAR; TRAP TO VECTOR 114 WHEN SET.

2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167



2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
2217  
2218  
2219  
2220  
2221  
2222

5.3 MS11-M CSR

-----															
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
!DE	!EU	!SI	!	!			ADDRESS			!SE	!IP	!DC	!EC	!EE	!
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
-----															
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00

BIT ASSIGNMENTS ARE DEFINED AS FOLLOWS:

BIT15 UNCORRECTABLE  
ERROR THIS BIT IS SET  
IF A DBE OCCURS, AND  
THE ERROR ADDRESS IS  
STORED IN THE CSR.  
THIS BIT IS ALSO SET  
IN THE ECC DISABLE  
MODE IF AN SBE OR DBE  
OCCURS.

BIT14 EUB ERROR  
RETRIEVAL IF THE MEM-  
ORY IS ON AN EXTENDED  
UNIBUS, WHEN BIT14 IS  
ZERO AND EITHER BIT4  
OR BIT 15 IS A ONE,  
THE LOW ORDER FAILING  
ADDRESSES ARE AVAILAB-  
LE (BITS 11-17); WHEN  
BIT14 IS ONE, THE HIGH  
ORDER FAILING ADDRESS-  
ES ARE AVAILABLE (BITS  
18-21 OF ADDRESS). IF  
THE MEMORY IS ON A  
UNIBUS, A JUMPER DIS-  
ABLES THIS BIT SO THAT  
IT IS READ ONLY, AND  
EQUAL TO ZERO.

BIT13 SET INH:BIT MODE  
WHEN THIS BIT IS SET  
TO A 1, IT ENABLES THE  
INH MODE POINTER TO  
INHIBIT EITHER THE  
FIRST OR SECOND 16K  
FROM EVER GOING INTO  
THE DIAG. CHECK OR  
ECC DISABLE MODE.  
WHEN THIS BIT IS SET  
TO A 0, IT ALLOWS THE  
DIAG. CHECK MODE

2224  
2225  
2226  
2227  
2228  
2229  
2230  
2231  
2232  
2233  
2234  
2235  
2236  
2237  
2238  
2239  
2240  
2241  
2242  
2243  
2244  
2245  
2246  
2247  
2248  
2249  
2250  
2251  
2252  
2253  
2254  
2255  
2256  
2257  
2258  
2259  
2260  
2261  
2262  
2263  
2264  
2265  
2266  
2267  
2268  
2269  
2270  
2271  
2272  
2273  
2274  
2275  
2276  
2277

AND/OR ECC DISABLE  
MODE TO OPERATE OVER  
THE ENTIRE MEMORY ON  
THE BOARD.

BITS 11-5 ERROR  
ADDRESS WITH BIT02  
CLEARED AND BIT14 SET,  
THEY CONTAIN THE HIGH  
ORDER ERROR ADDRESS  
(BITS 21-18); WHEN  
BIT02 AND BIT14 ARE  
CLEARED, THEY CONTAIN  
THE LOW ORDER ERROR  
ADDRESS (BITS 17-11);  
WHEN BIT02 IS SET THEY  
CONTAINS CHECK BITS  
FOR ECC.

BIT04 SINGLE ERROR SET  
WHENEVER SINGLE ERROR  
OCCURS.

BIT03 INHIBIT MODE  
POINTER THE INHIBIT  
MODE POINTER WORKS IN  
CONJUNCTION WITH THE  
SET INHIBIT MODE BIT.  
WHEN BIT13 IS SET TO A  
1, A 16K PORTION OF  
MEMORY IS INHIBITED  
FROM OPERATING IN THE  
ECC DISABLE MODE OR  
DIAGNOSTIC CHECK MODE.  
THE INHIBIT MODE  
POINTER INDICATES  
WHICH 16K IS BEING IN-  
HIBITED; E.G.-IF BIT3  
=1, THE SECOND 16K OF  
MEMORY IS INHIBITED.  
WHEN BIT13 IS SET TO A  
0, BIT3 BECOMES  
INOPERATIVE.

BIT02 DIAGNOSTIC CHECK  
MODE WHEN SET ENABLES  
READ-WRITE OF CHECK  
BITS(SEE BITS 11-5).  
IF A DBE OCCURS IN  
THIS MODE (WITH BIT1=0  
, BIT15 IS SET, BUT  
THE CHECK BITS READ

2279  
2280  
2281  
2282  
2283  
2284  
2285  
2286  
2287  
2288  
2289  
2290  
2291  
2292  
2293  
2294  
2295  
2296  
2297  
2298  
2299  
2300  
2301  
2302  
2303  
2304  
2305  
2306  
2307  
2308  
2309

ARE STORED IN BITS  
11-5, NOT THE DBE  
ADDRESS BITS.

BIT01 DISABLE ERROR  
CORRECTION WHEN SET NO  
SINGLE ERROR CORRECTI-  
ON TAKES PLACE. A  
SINGLE BIT ERROR WILL  
SET BIT04 AND BIT15  
AND ASSERT BUS PBL L  
IF BIT00 IS ASSERTED;  
A DOUBLE ERROR WILL  
SET SET BIT15 AND AS-  
SERT BUS PBL L IF  
BIT00 IS ASSERTED.  
THE ERROR ADDRESS IS  
STORED IN THE CSR, AND  
CORRECT CHECK BITS ARE  
GENERATED AND STORED  
ON A WRITE.

BIT00 UNCORRECTABLE  
ERROR ENABLE WHEN SET  
ENABLES TRAP TO VECTOR  
114 ON UNCORRECTABLE  
ERROR.

2311  
2312  
2313  
2314  
2315  
2316  
2317  
2318  
2319  
2320  
2321  
2322  
2323  
2324  
2325  
2326  
2327  
2328  
2329  
2330  
2331  
2332  
2333  
2334  
2335  
2336  
2337  
2338  
2339  
2340  
2341  
2342  
2343  
2344  
2345  
2346  
2347  
2348  
2349  
2350  
2351  
2352  
2353  
2354  
2355

## 6.0 SUB-TEST SUMMARIES

## 6.1 TESTS

## TEST 1

BIT TEST OF ALL CSR'S/MATCH ALL CSR'S WITH MEMORY  
(CSR ACCESS MAY CAUSE WRONG TYPE OF TRAPS)

## TEST 2

TEST BANK 0 ACCESSES  
FAILURES ARE FATAL.

## TEST 3

TEST BANKS 1-167 (OCTAL) FOR ZEROS AND ONES  
ERRORS ARE NOT TYPED HERE - ONLY LOGGED IN  
THE CONFIGURATION TABLE

## TEST 4

ECC INHIBIT MODE POINTER TEST

## TEST 5

DIAGNOSTIC MODE DISPATCH ROUTINE  
THIS TEST RUNS ALL THE PATTERNS IN THE  
MODE SELECTED.

## TEST 6

UNIQUE BANK TEST  
PATTERN 27 IS RUN

## 6.2 TESTS

## 6.2.1 GENERAL TEST INFORMATION

ACTUAL TESTS ARE IDENTIFIED BY SYMBOLIC LOCATIONS 'MTPXY' WHERE X MAY BE ANY SUB PROGRAM INDICATOR (A,B,C,ETC) OR 0 AND Y WILL BE THE NUMBER OF THE TEST.

SETUP PROCEDURES FOR EACH TEST ARE IDENTIFIED BY SYMBOLIC LOCATIONS 'MTOOY' WHERE Y WILL BE THE NUMBER OF THE TEST.

TESTS RESIDE IN 4 SCRIPTS THAT ARE SCANNED FOR EXECUTION. SYMBOLIC LOCATION 'MKCSRT' IS A TABLE OF TESTS THAT CAN RUN ONCE FOR EACH ECC BANK (TWICE FOR INTERLEAVED MS11-M'S). SYMBOLIC LOCATION 'MKPAT' IS A TABLE OF TESTS THAT CAN RUN ON EACH BANK OF ECC MEMORY. SYMBOLIC LOCATION 'MJPAT' IS A TABLE OF TESTS THAT CAN RUN ON EACH BANK OF PARITY MEMORY. SYMBOLIC LOCATION 'FSPAT' IS A TABLE OF TESTS THAT CAN BE RUN IN FIELD SERVICE MODE (COMMAND 5).

THE 1ST 3 SCRIPTS ARE COMPLETELY CONTROLLED BY THE APT E-TABLE (EVEN IF NOT RUNNING UNDER APT). MODIFICATIONS TO THIS TABLE CAN BE MADE (1) WITH APT, OR (2) MANUALLY.

## EXAMPLE E-TABLE SEGMENT:

:THE FOLLOWING LOCATIONS SPECIFY WHICH TESTS  
:ARE TO BE RUN FOR PARTICULAR MEMORIES

:REFERENCE THE TABLE LISTED BELOW TO RELATE BITS TO TESTS.  
:BIT0 SET WILL RUN THE FIRST ENTRY IN THE TABLE, BIT0 SET  
:IN THE SECOND WORD WILL RUN THE 17TH ENTRY IN THE TABLE...

:NOTE\*\*NULL TESTS DO NOT TAKE ANY TIME

\$DDW0:	.WORD	177777
\$DDW1:	.WORD	177777
\$DDW2:	.WORD	177777
\$DDW3:	.WORD	177777
\$DDW4:	.WORD	177777
\$DDW5:	.WORD	177777

RECOMMENDED VALUE	
:ECC CSR TESTS	177777 TABLE = MKCSRT:
:ECC CSR TESTS	177777 TABLE = MKCSRT:
:ECC TESTS	103777 TABLE = MKPAT:
:ECC TESTS	177777 TABLE = MKPAT:
:PARITY TESTS	003777 TABLE = MJPAT:
:PARITY TESTS	177774 TABLE = MJPAT:

2404  
2405  
2406  
2407  
2408  
2409  
2410  
2411  
2412  
2413  
2414  
2415  
2416  
2417  
2418  
2419  
2420  
2421  
2422  
2423  
2424  
2425  
2426  
2427  
2428  
2429  
2430  
2431  
2432  
2433

## 6.2.2 SPECIFIC TESTS

### 6.2.2.1 TEST 0 BASIC DATA TEST

WRITES READS R2 INTO A 16K BANK.

THIS IS USED FOR ZEROS AND ONES TESTING AND IN FIELD SERVICE MODE FOR ANY CONSOLE SELECTED TEST.

IT CAN EXECUTE OUT OF THE USER INSTRUCTION PAR'S.

#### NOTE

IT IS FREQUENTLY MODIFIED DYNAMICALLY SUCH THAT (1) IT RETURNS AFTER WRITING ONLY (THE 1ST NOP IS REPLACED WITH A RETURN) OR (2) IT ONLY COUNTS ERRORS (THE CODE PERRO2 AND A IP ARE REPLACED WITH INC @#PATERR).

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 53

2435  
2436  
2437  
2438  
2439  
2440  
2441  
2442  
2443  
2444  
2445  
2446  
2447  
2448  
2449  
2450  
2451  
2452

6.2.2.2 TEST 1 ADDRESS TEST

WRITES READS AN INCREMENTING PATTERN EQUIVALENT TO PHYSICAL  
ADDRESSED INTO A 16K BANK.

IT CAN EXECUTE OUT OF THE USER INSTRUCTION PAR'S.



2454  
2455  
2456  
2457  
2458  
2459  
2460  
2461  
2462  
2463  
2464  
2465  
2466  
2467

#### 6.2.2.3 TEST 2 COMPLEMENT ADDRESS TEST

WRITES THE COMPLEMENT OF THE PHYSICAL ADDRESS FROM HIGH ADDRESSES TO LOW (WRITE DOWN) AND READS FROM LOW ADDRESSES TO HIGH (READ UP).

THIS PROVIDES THE COMPLEMENT OF THE COVERAGE OF TEST 1 IN BOTH DATA PATTERN AND ADDRESSING SEQUENCE.

IT CAN EXECUTE OUT OF THE USER INSTRUCTION PAR'S.

2469  
2470  
2471  
2472  
2473  
2474  
2475  
2476  
2477  
2478  
2479  
2480  
2481  
2482  
2483  
2484  
2485  
2486

#### 6.2.2.4 TEST 3 3 XOR 9

WRITES READS A TEST THAT COMPLEMENTS AS ADDRESS BITS 3 AND 9  
CHANGE.

THIS TEST IS RUN 4 TIMES (1) WITH ZEROS ONES, (2) WITH ONES  
ZEROS, (3) WITH 401 ONES, AND (4) WITH ONES 401. THE TEST OF  
THE 401 IS TO FORCE A THE PARITY BITS TO BECOME INVOLVED.

IT CAN EXECUTE OUT OF THE USER DATA PDR'S, THE USER INSTRUCTION PAR'S,  
THE KERNEL DATA PAR'S AND THE SUPERVISOR DATA PAR'S.

2488  
2489  
2490  
2491  
2492  
2493  
2494  
2495  
2496  
2497  
2498  
2499  
2500  
2501  
2502  
2503  
2504  
2505  
2506  
2507  
2508  
2509  
2510  
2511

#### 6.2.2.5 TEST 4 ROTATING ZEROS TEST

WRITES A BACKGROUND PATTERN OF ONES. ROTATES A ZERO CARRY BIT LEFT THRU EACH PAR OF BYTES (18 TIMES) AND THEN CHECKS THAT THE CARRY IS ZERO AND THE WORD (2 BYTES) IS STILL ALL ONES.

IT CAN EXECUTE OUT OF THE USER DATA PAR'S AND THE KERNEL DATA PAR'S.

#### NOTE

IT IS NOT UNCOMMON TO OBSERVE THE GOOD DATA EQUAL TO THE BAD DATA. THIS INDICATES THAT THE CARRY WAS NOT CLEAR AFTER 18 ROLB'S.

2513  
2514  
2515  
2516  
2517  
2518  
2519  
2520  
2521  
2522  
2523  
2524  
2525  
2526  
2527  
2528  
2529  
2530  
2531  
2532  
2533  
2534  
2535  
2536  
2537

#### 6.2.2.6 TEST 5 ROTATING ONES TEST

WRITES A BACKGROUND PATTERN OF ZEROS. ROTATES A ONE CARRY BIT LEFT THRU EACH PAIR OF BYTES (18 TIMES) AND THEN CHECKS THAT THE CARRY IS A ONE AND THE WORD (2 BYTES) IS STILL ALL ZEROS.

THIS PROVIDES THE COMPLEMENT OF THE COVERAGE OF TEST 4 IN DATA.

IT CAN EXECUTE OUT OF THE USER DATA PAR'S AND THE KERNEL DATA PAR'S.

#### NOTE

IT IS NOT UNCOMMON TO OBSERVE THE GOOD DATA EQUAL THE BAD DATA. THIS INDICATES THAT THE CARRY WAS NOT SET AFTER 18 ROLB'S.

2539  
2540  
2541  
2542  
2543  
2544  
2545  
2546  
2547  
2548  
2549  
2550  
2551  
2552

#### 6.2.2.7 TEST 6 INITIAL DATA TEST

WRITES READS A DOUBLE WORD FIRST WITH ALL BITS 0 EXCEPT 1 (FOR EVERY BIT POSITION), SECOND WITH ALL BITS 1 EXCEPT 1 (FOR EVERY BIT POSITION).

THIS IS A VERY QUICK CHECK OF THE DATA PATHS.

2554  
2555  
2556  
2557  
2558  
2559  
2560  
2561  
2562  
2563  
2564  
2565  
2566  
2567  
2568  
2569  
2570  
2571  
2572  
2573  
2574  
2575  
2576  
2577  
2578

#### 6.2.2.8 TEST 7 ADDRESS BIT TEST

WRITES A BACKGROUND OF ALL ZEROS.

READ ADDRESS 1 FOR A 0 BYTE.

COMPLEMENT ADDRESS 1.

READ ADDRESS 1 FOR A NON 0 BYTE.

FOR EACH ADDRESS BIT POSITION FROM BIT 1:

VIRTUAL (2, 4, 10, 20, 40, 100, 200, 400, 1000, 2000, 4000, 10000, 60000, 20000)

PHYSICAL (60002, 60004, 60010, 60020, 60040, 60100, 60200, 60400, 61000, 62000, 64000, 70000, 140000, 100000)

READ ADDRESS FOR A 0 WORD.

COMPLEMENT ADDRESS CONTENTS.

READ ADDRESS FOR A NON-ZERO WORD.

THIS IS A VERY QUICK CHECK OF THE ADDRESS BIT UNIQUENESS.

2580  
2581  
2582  
2583  
2584  
2585  
2586  
2587  
2588  
2589  
2590  
2591  
2592  
2593  
2594  
2595  
2596  
2597  
2598  
2599  
2600  
2601  
2602  
2603

#### 6.2.2.9 TEST 10 BYTE ADDRESSING TEST

WITH ECC DISABLED.  
WRITES ALL ONES TO A DOUBLE WORD.  
FOR EACH OF THE 4 BYTES IN THE DOUBLE WORD.  
CLEAR ONE BYTE.  
READS ALL 4 BYTES FROM DOUBLE WORD.  
CHECKS FOR ONLY PROPER BYTE CLEAR.  
ALL OTHER BYTES SET TO ALL ONES.

THIS IS ONLY DONE ON ONE DOUBLE WORD ADDRESS.

#### NOTE

THIS IS RUN FOR ECC MEMORY ONLY

## 6.2.2.10 TEST 11 SINGLE BIT ERROR TEST

1. CREATE A SINGLE BIT ERROR.
2. READ DATA UNCORRECTED (WITH ECC DISABLE).
3. CHECK THAT SBE AND DBE FLAGS ARE SET, AND THE ERROR ADDRESS IS LATCHED.
4. READ FIRST WORD OF DATA CORRECTED (WITH ECC ENABLED)
5. CHECK THAT THE CSR SINGLE BIT ERROR FLAG WAS SET, AND THE ERROR ADDRESS WAS LATCHED.
6. CLEAR SBE FLAG.
7. READ SECOND WORD OF DATA CORRECTED (WITH ECC ENABLED).
8. CHECK THAT THE CSR SINGLE BIT ERROR FLAG WAS SET.
9. DO (1-7) FOR A SINGLE BIT ERROR IN EACH OF 32 POSITIONS OF A DOUBLE WORD.  
I.E. (32 TIMES)
10. IF NOT IN QUICK VERIFY MODE THEN DO (1-8) FOR DATA CONSISTING OF 1 BIT SET IN EACH OF 32 POSITIONS OF A DOUBLE WORD.  
I.E. (32 X 32 = 1024 TIMES)
11. DO (1-9) FOR COMPLEMENTED DATA (1 BIT CLEAR IN EACH OF 32 POSITIONS OF A DOUBLE WORD).  
I.E. (1024 X 2 = 2048 TIMES)  
OR (32 X 2 = 64 TIMES (QUICK VERIFY))
12. DO (1-7) FOR A DOUBLE WORD EQUAL TO (000000,000000), AND ALL POSSIBLE SINGLE BIT ERROR COMBINATIONS FORCED INTO THE CHECK BITS (CSR BITS 5-11).
13. CLEAR ANY ERRORS OUT OF TEST LOCATIONS.

THIS INSURES THAT ALL SINGLE BIT ERRORS CAN BE CORRECTED AND DETECTED.

## NOTE

THIS TEST IS RUN FOR MS11-M MEMORY ONLY

2605  
2606  
2607  
2608  
2609  
2610  
2611  
2612  
2613  
2614  
2615  
2616  
2617  
2618  
2619  
2620  
2621  
2622  
2623  
2624  
2625  
2626  
2627  
2628  
2629  
2630  
2631  
2632  
2633  
2634  
2635  
2636  
2637  
2638  
2639  
2640  
2641  
2642  
2643  
2644  
2645  
2646  
2647  
2648  
2649  
2650  
2651  
2652  
2653  
2654  
2655  
2656  
2657  
2658  
2659



2661  
2662  
2663  
2664  
2665  
2666  
2667  
2668  
2669  
2670  
2671  
2672  
2673  
2674  
2675  
2676  
2677  
2678  
2679  
2680  
2681  
2682  
2683  
2684  
2685  
2686  
2687  
2688  
2689  
2690  
2691  
2692  
2693  
2694  
2695  
2696  
2697  
2698  
2699  
2700

#### 6.2.2.11 TEST 12 WRITE BYTE CLEARS SBE TEST

1. CREATE A SINGLE BIT ERROR.
2. WRITE A BYTE OF DOUBLE WORD TO ONES.
3. READ A BYTE OF DOUBLE WORD.
4. IF THIS IS MS11-M, THE SBE FLAG SHOULD BE SET.
5. THE BYTE SHOULD HAVE BEEN EQUAL TO ONES.
6. DO (1-5) FOR EACH OF THE 4 BYTES OF THE DOUBLE WORD
7. DO (1-6) FOR A SINGLE BIT ERROR IN EACH OF 32 POSITIONS OF A DOUBLE WORD  
I.E. (32 TIMES)
8. IF NOT IN QUICK VERIFY MODE THEN DO (1-7) FOR DATA CONSISTING OF 1 BIT SET IN EACH OF 32 POSITIONS OF A DOUBLE WORD.  
I.E. (32 X 32 = 1024 TIMES)
9. CLEAR ANY ERRORS OUT OF TEST LOCATIONS.

THIS INSURES THAT SINGLE BIT ERRORS IN THE DATA PORTION (NOT IN CHECKBITS) CAN BE CLEARED BY WRITING THE CORRESPONDING BYTE AND THAT WRITING ANY OTHER BYTE DOES NOT CHANGE THE EXISTING SINGLE BIT ERROR.

#### NOTE

THIS TEST IS RUN FOR MS11-M MEMORY ONLY.

## 6.2.2.12 TEST 13 CREATE DOUBLE BIT ERROR TEST

1. CREATE A DOUBLE BIT ERROR.
2. ACCESS THE DATA (TST INSTRUCTION).
3. CHECK THAT THE CSR DBE FLAG IS SET, AND THE ERROR ADDRESS IS LATCHED.
4. INITIALIZE CSR TO ALLOW PARITY TRAPS ON DBE'S.
5. ACCESS THE DATA (TST INSTRUCTION).
6. CHECK THAT A PARITY TRAP OCCURRED.
7. DO (1-6) FOR THE 2ND BIT OF EACH DOUBLE BIT ERROR IN EACH OF 32 POSITIONS OF A DOUBLE WORD LESS THE ONE POSITION OF THE 1ST BAD BIT.  
I.E. (31 TIMES)
8. IF NOT IN QUICK VERIFY MODE THEN DO (1-7) FOR THE 1ST BIT OF EACH OF DOUBLE BIT ERROR IN EACH OF 32 POSITIONS OF A DOUBLE WORD.  
I.E. (31 X 32 = 992 TIMES)
9. DO (1-8) FOR COMPLEMENTED DATA (ONES VERSUS ZEROS IN DOUBLE WORD)  
I.E. (992 X 2 = 1984 TIMES)  
OR (31 X 2 = 62 TIMES (QUICK VERIFY))
10. DO (1-6) FOR A DOUBLE WORD EQUAL TO (000000,000000), AND ALL POSSIBLE DOUBLE BIT ERROR COMBINATIONS FORCED INTO EACH OF THE CHECK BITS (CSR BITS 5-11).
11. CLEAR ANY ERRORS OUT OF TEST LOCATIONS.

THIS INSURES THAT ALL DOUBLE BIT ERRORS CAN BE CREATED AND DETECTED AND CAUSE TRAPS.

## NOTE

- 1) THIS TEST IS RUN ON THE MS11-M ONLY.
- 2) THIS TEST IS ONLY RUN DURING THE FIRST (QV) PASS WHEN UNDER ACT OR APT, AND IS RUN FOR ECC MEMORY ONLY.

2756  
2757  
2758  
2759  
2760  
2761  
2762  
2763  
2764  
2765  
2766  
2767  
2768  
2769  
2770  
2771  
2772  
2773  
2774  
2775  
2776  
2777  
2778  
2779  
2780  
2781  
2782  
2783  
2784  
2785  
2786  
2787  
2788  
2789  
2790  
2791  
2792  
2793

## 6.2.2.13

## TEST 14 BASIC DOUBLE BIT ERROR TEST

1. WRITE THE CSR TO ENABLE DIAG MODE WITH A DOUBLE BIT ERROR CHECK BITS OF 110011 AND UNCORRECTABLE ERROR INDICATION ENABLED.
2. WRITE FIRST AUT IN A 16K BANK WITH DATA OF ALL ZERO'S. THIS WILL WRITE THE CHECK BITS IN (1)
3. READ ADDRESS, THIS SHOULD CAUSE A DOUBLE BIT ERROR. BUS PBL IS ASSERTED AND WE CHECK FOR A PARITY TRAP TO OCCUR.
4. READ THE CSR FOR CHECK BITS IN (1) AND UNCORRECTABLE ERROR INDICATOR.
5. WRITE ONES TO THE HIGH BYTE OF THE ADDRESS UNDER TEST. SINCE A DBE EXISTS AT THIS ADDRESS THE WRITE SHOULD BE ABORTED.
6. READ ADDRESS AND CHECK FOR A PARITY TRAP TO OCCUR AS A RESULT OF (5)
7. REPEAT 5 AND 6 FOR DATA OF ONES IN THE LOW BYTE AND CHECK FOR WRITE ABORT AND PARITY TRAP.

THIS TEST CHECKS TO SEE IF A DOUBLE BIT ERROR WILL BE ABORTED AND A BYTE WRITE OF A DOUBLE BIT ERROR WILL BE ABORTED.

## NOTE

THIS TEST IS ONLY RUN FOR THE MS11-P

## 6.2.2.14 TEST 15 WRITE INHIBIT OF BYTE WITH DBE

1. CREATE A DOUBLE BIT ERROR.
2. DO A MOV B IMMEDIATE TO TEST BYTE.
3. CHECK THAT DOUBLE WORD IS STILL BAD (UNCHANGED-WITH DBE).
4. DO (2-3) ON ALL 4 BYTES OF DOUBLE WORD.
5. DO (1-4) FOR THE 2ND BIT OF EACH DOUBLE BIT ERROR IN EACH OF 32 POSITIONS OF A DOUBLE WORD LESS THE ONE POSITION OF THE 1ST BAD BIT.  
I.E. (31 TIMES)
6. IF NOT IN QUICK VERIFY MODE THEN DO (1-5) FOR THE 1ST BIT OF EACH DOUBLE BIT ERROR IN EACH OF 32 POSITIONS OF A DOUBLE WORD.  
I.E. (31 X 32 = 922 TIMES)
7. DO (1-6) FOR COMPLEMENTED DATA (ONES VERSUS ZEROS IN DOUBLE WORD).  
I.E. (992 X 2 = 1984 TIMES)  
OR (31 X 2 = 62 TIMES (QUICK VERIFY))
8. DO (1-4) FOR A DOUBLE WORD EQUAL TO (000000,000000), AND ALL POSSIBLE DOUBLE BIT ERROR COMBINATIONS FORCED INTO THE CHECK BITS (CSR BITS 5-11).
9. CLEAR ANY ERRORS OUT OF TEST LOCATIONS.

THIS INSURES THAT NO DOUBLE BIT ERROR CAN BE CLEARED BY A MOV B TO ANY AFFECTED BYTE.

## NOTE

- 1) THIS TEST IS RUN ON THE MS11-M ONLY.
- 2) THIS TEST IS ONLY RUN DURING THE FIRST (QV) PASS WHEN UNDER ACT OR APT, AND IS RUN FOR ECC MEMORY ONLY.

2848  
2849  
2850  
2851  
2852  
2853  
2854  
2855  
2856  
2857  
2858  
2859  
2860  
2861  
2862  
2863  
2864  
2865  
2866  
2867  
2868  
2869  
2870  
2871  
2872  
2873  
2874  
2875  
2876  
2877  
2878  
2879  
2880  
2881  
2882  
2883  
2884  
2885  
2886  
2887  
2888  
2889  
2890  
2891  
2892  
2893  
2894  
2895  
2896  
2897  
2898  
2899

#### 6.2.2.15 TEST 16 WRITE INHIBIT OF WORD WITH DBE TEST

1. CREATE A DOUBLE BIT ERROR.
2. DO MOV IMMEDIATE ON TEST LOCATION.
3. CHECK THAT DOUBLE WORD IS STILL BAD (UNCHANGED-WITH DBE).
4. DO (2-3) ON BOTH DOUBLE WORDS.
5. DO (1-4) FOR THE 2ND BIT OF EACH DOUBLE BIT ERROR IN EACH OF 32 POSITIONS OF A DOUBLE WORD LESS THE ONE POSITION OF THE 1ST BAD BIT.  
I.E. (31 TIMES)
6. IF NOT IN QUICK VERIFY MODE THEN DO (1-5) FOR THE 1ST BIT OF EACH DOUBLE BIT ERROR IN EACH OF 32 POSITIONS OF A DOUBLE WORD.  
I.E. (32 X 32 = 992 TIMES)
7. DO (1-6) FOR COMPLEMENTED DATA (ONES VERSUS ZEROS IN DOUBLE WORD).  
I.E. (992 X 2 = 1984 TIMES)  
OR (31 X 2 = 62 TIMES (QUICK VERIFY))
8. DO (1-4) FOR A DOUBLE WORD EQUAL TO (000000,000000), AND ALL POSSIBLE DOUBLE BIT ERROR COMBINATIONS FORCED INTO THE CHECK BITS (CSR BITS 5-11).
9. CLEAR ANY ERRORS OUT OF TEST LOCATIONS.

THIS INSURES THAT NO DOUBLE BIT ERROR CAN BE CLEARED BY A MOV TO ANY AFFECTED WORD.

#### NOTE

- 1) THIS TEST IS RUN ON THE MS11-M ONLY
- 2) THIS TEST IS ONLY RUN DURING THE FIRST (QV) PASS WHEN UNDER ACT OR APT, AND IS RUN FOR ECC MEMORY ONLY.

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 67

2901  
2902  
2903  
2904  
2905  
2906  
2907  
2908  
2909  
2910  
2911  
2912  
2913  
2914  
2915  
2916

## 6.2.2.16 TEST 17 HOLDING 1'S 0'S TEST

1. WRITE A 16K BANK WITH ALTERNATING BYTES OF ZEROS ONES  
WRITING A BYTE AT A TIME.
2. READ EACH WORD FOR CORRECT TEST.
3. DO (1-2) AGAIN FOR A COMPLEMENT TEST.

THIS CHECKS THE MEMORY FOR THE CAPABILITY OF HOLDING 0'S 1'S.

2918  
2919  
2920  
2921  
2922  
2923  
2924  
2925  
2926  
2927  
2928  
2929  
2930  
2931  
2932  
2933  
2934  
2935  
2936  
2937  
2938  
2939  
2940  
2941  
2942  
2943  
2944  
2945  
2946  
2947  
2948  
2949  
2950  
2951  
2952  
2953  
2954  
2955

#### 6.2.2.17 TEST 20 SYNDROME BITS TO THE CSR ON A SBE TEST

1. WRITE CSR WITH CHECK BITS TO CORRECT BIT 0 OF THE FIRST AUT 16K BANK FROM A 0 TO A 1 WITH DIAG MODE.
2. WRITE AUT WITH DATA OF 0'S CREATING A SBE.
3. CLEAR CSR.
4. READ THE AUT TO CLOCK THE ADDRESS AND SYNDROMES INTO THE CSR.
5. READ THE CSR FOR THE SBE INDICATOR, BIT 4.
6. WRITE THE CSR TO DIAG MODE TO CLOCK THE SYNDROME BITS INTO CSR BITS 5-11.
7. READ THE CSR FOR THE PROPER SYNDROME BITS.
8. REPEAT 1-7 FOR ALL 16 DATA BITS.
9. REPEAT 1-8 FOR DATA OF ONES SO THAT A CORRECTION WILL OCCUR FROM A 1 TO A ZERO.

THIS TEST CHECKS TO SEE THAT THE EDC CHIP CAN DETECT SINGLE BIT ERRORS FOR ALL 16 DATA BITS BY CHECKING FOR CSR BIT#4 AND THAT THE PROPER SYNDROME BITS ARE PLACED IN THE CSR.

#### NOTE

THIS TEST IS ONLY RUN FOR THE MS11-P

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

#### 6.2.2.18 TEST 21 MARCHING 0'S 1'S TEST

1. WRITE A BACKGROUND OF ALTERNATING BYTES OF ZEROS ONES
2. FOR THE 16K BANK ADDRESSING DOWN
  - (A) READ CHECK A WORD
  - (B) BYTE SWAP A WORD
  - (C) READ CHECK A WORD
3. FOR THE 16K BANK ADDRESSING UP
  - (A) READ CHECK A WORD
  - (B) BYTE SWAP A WORD
  - (C) READ CHECK A WORD
4. FOR THE 16K BANK ADDRESSING UP
  - (A) READ CHECK A WORD
  - (B) BYTE SWAP A WORD
  - (C) READ CHECK A WORD
5. FOR THE 16K BANK ADDRESSING DOWN
  - (A) READ CHECK A WORD
  - (B) BYTE SWAP A WORD
  - (C) READ CHECK A WORD

THIS CHECKS THE INTEGRITY OF THE 32 BIT DOUBLE WORDS.

IT CAN EXECUTE OUT OF THE USER DATA PAR'S.

#### NOTE

IT IS NOT UNCOMMON TO SEE A MISLEADING ERROR TYPEOUT BECAUSE THE SECOND TEST IN EACH CASE IS BASED UPON A BYTESWAP OF THE FIRST TEST WHICH MAY OR MAY NOT HAVE FAILED. IF THE ERROR REPORT INDICATES ERRORS IN PAIRS WITH THE BAD BIT IN THE SECOND REPORT BEING THE SAME BIT POSITION, RELATIVE TO A BYTE THEN YOU SHOULD IGNORE THE SECOND ERROR REPORT.



## 6.2.2.19 TEST 22 REFRESH TEST

1. WRITE A DIAGONAL TEST OF ONES ON EVERY KDIAG(TH) STRIPE  
WRITE ZEROS ELSEWHERE.

THIS TEST IS ON ADDRESSES NOT BIT POSITIONS.

EXAMPLE:

ADDRESS

LSB'S

MSB'S

MSB'S	0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	1	0
0	1	0	0	0	0	1	0	0
1	0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1	0
0	1	0	0	0	0	1	0	0
1	0	0	0	1	0	0	0	0

## NOTE

EXAMPLE USES KDIAG OF VALUE 4 MORE TYPICAL IS A VALUE OF 8. CONSULT THE SYMBOLIC DEFINITION OF 'KDIAG' IN THE PROGRAM LISTING TO BE SURE.

2. DISTURB EACH ROW FOR > 3.2MS
3. READ CHECK DIAGONAL PATTERN
4. DO (1-3) KDIAG TIMES MOVING THE PLACEMENT OF THE DIAGONAL STRIPE TO COVER ALL ADDRESS POSITIONS.
5. DO (1-4) FOR A COMPLEMENT PATTERN  
(ZEROS IN A BACKGROUND OF ONES)

## NOTE

THIS TEST IS NOT NORMALLY EXECUTED EXCEPT UNDER APT OR ACT. IT MAY BE INVOKED VIA FIELD SERVICE COMMAND 13 (KAMIKAZE MODE).

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

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 71

3062  
3063  
3064  
3065  
3066  
3067  
3068  
3069  
3070  
3071  
3072  
3073  
3074  
3075  
3076  
3077  
3078  
3079  
3080  
3081  
3082

6.2.2.20 TEST 23 SHIFTING DIAGONAL TEST

SIMILAR IN OVERALL OPERATION TO TEST 22 EXCEPT IT DOES NOT DELAY  
FOR REFRESH AND DISTURB ROWS.

NOTE

THIS TEST IS NOT NORMALLY EXECUTED  
EXCEPT UNDER APT OR ACT. IT MAY BE  
INVOKED VIA FIELD SERVICE COMMAND 13  
(KAMIKAZE MODE).

3084  
3085  
3086  
3087  
3088  
3089  
3090  
3091  
3092  
3093  
3094  
3095  
3096  
3097  
3098  
3099  
3100  
3101  
3102  
3103

## 6.2.2.21 TEST 24 FAST GALLOPING PATTERN TEST

THIS DOES A CLASSICAL GALLOPING PATTERN EXCEPT THAT ADDRESSING IS  
INCREMENTED BY 400 OCTAL (EVERY 64TH DOUBLE WORD)

## NOTE

THIS TEST IS NOT NORMALLY EXECUTED  
EXCEPT UNDER APT OR ACT. IT MAY BE  
INVOKED VIA FIELD SERVICE COMMAND 13  
(KAMIKAZE MODE).

## 6.2.2.22 TEST 25 INTERRUPT ENABLE TEST

1. SET CSR TO ALLOW UNCORRECTABLE ERROR TRAPS.
2. ACCESS TEST DOUBLE WORDS.
3. CHECK THAT NO UNCORRECTABLE ERROR TRAP OCCURRED.
4. ENABLE CSR FOR SBE TRAPS.
5. ACCESS TEST DOUBLE WORDS.
6. CHECK THAT NO SBE TRAP OCCURRED.
7. WRITE A SBE IN 1 BYTE.
8. DISABLE CSR TRAPS.
9. ACCESS TEST DOUBLE WORDS.
10. CHECK THAT NO TRAPS OCCURRED.
11. ENABLE CSR FOR SBE TRAPS.
12. ACCESS TEST DOUBLE WORDS.
13. CHECK TO INSURE TRAP OCCURRED.
14. DO (7-13) FOR THE 3 OTHER BYTES IN THE DOUBLE WORD.
15. CREATE A DBE IN 1 BYTE.
16. DISABLE CSR TRAPS.
17. ACCESS THE TEST DOUBLE WORD.
18. CHECK THAT NO TRAPS OCCURRED.
19. ENABLE CSR FOR DBE TRAPS.
20. ACCESS THE TEST DOUBLE WORD.
21. CHECK TO INSURE TRAP OCCURRED.
22. ENABLE CSR FOR SBE TRAPS.
23. ACCESS THE TEST DOUBLE WORD.
24. CHECK TO INSURE TRAP OCCURRED.
25. DO (15-24) FOR THE 3 OTHER BYTES IN THE DOUBLE WORD.

THIS INSURES THAT SBE'S DBE'S GIVE THE CORRECT TYPE OF TRAPS.

## NOTE

THIS TEST IS RUN FOR MS11-M MEMORY ONLY.

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

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

#### 6.2.2.23 TEST 26 RANDOM DATA TEST

WRITE RANDOM DATA IN A 16K BANK WHILE INCREMENTING THE ADDRESSES.

READ CHECK RANDOM DATA.

THIS ROUTINE REGENERATES THE SAME RANDOM NUMBERS BY USING THE SAME

SEED AS THE WRITE SEQUENCE. AFTER THE READ CHECK THE SEED IS UPDATED SO THAT THE NEXT USE OF THIS PATTERN WILL NOT INVOKE THE SAME SEQUENCE OF RANDOM NUMBERS.

IF YOU WISH TO CHANGE THE RANDOM SEQUENCE SO THAT IT IS DIFFERENT THAN ANY OTHER RUN IN THE SAME CONFIGURATION THEN THERE ARE 2 WAYS OF DOING SO.

1. MODIFY SYMBOLIC LOCATIONS 'SEEDHI' AND 'SEEDLO' TO ANY NUMBER YOU LIKE.
2. ENTER FIELD SERVICE MODE AND EXECUTE THIS TEST (COMMAND 5) ON SOME (ANY GOOD) BANK FOR A SHORT TIME (30 SEC OR SO).

THIS CAN EXECUTE OUT OF THE USER DATA PAR'S, THE KERNEL DATA PAR'S, AND THE SUPERVISOR DATA PAR'S.

3194  
3195  
3196  
3197  
3198  
3199  
3200  
3201  
3202  
3203  
3204  
3205  
3206  
3207  
3208  
3209  
3210

#### 6.2.2.24 TEST 27 UNIQUE BANK TEST

THIS TEST USES TEST 0 TO WRITE READ THE BANK NUMBER IN EACH BANK.

IT DOES NOT TEST BANKS THAT REQUIRE RELOCATION TO TEST.

IT DOES NOT RUN AS PART OF ANY SCRIPT BUT RATHER IS ALWAYS RUN AFTER NORMAL PATTERN TESTS ARE COMPLETE.

3212  
3213  
3214  
3215  
3216  
3217  
3218  
3219  
3220  
3221  
3222  
3223  
3224  
3225  
3226  
3227  
3228

#### 6.2.2.25 TEST 30 FLUSH OUT DBE'S TEST

THIS READS EACH LOCATION THEN MOVES THE OLD VALUE BACK IN. THIS IS DONE WITH ECC DISABLED AND THEREFORE CORRECTS ANY DBE'S OR SBE'S (IF POSSIBLE).

IT DOES NOT RUN AS PART OF ANY SCRIPT BUT RATHER IS ALWAYS RUN JUST PRIOR TO THE END OF PASS CODE, AS PART OF A CONTROL 'C' (BOOT) COMMAND, AS PART OF END OF PASS SHUTDOWN FOR ACT OR LXDP CHAIN MODE, AS PART OF HANGING SEQUENCE AFTER AN ERROR IF UNDER ACT OR APT, AND AS PART OF A SHUTDOWN SEQUENCE DIRECTED BY SWITCH 8 (HALT PROGRAM).

## 6.2.2.26 TEST 31 SOB-A-LONG TEST

## RATIONALIZATION

IN ORDER TO CONCENTRATE THE MEMORY CYCLES OF A TEST INTO A PARTICULAR ADDRESS, WE MUST CUT THE OVERHEAD CYCLES TO A MINIMUM. FREQUENTLY, THE INSTRUCTION ITSELF MAY PROVIDE ADEQUATE DATA OR SET UP A BACKGROUND IN WHICH ANY COMPLEMENTED BIT MAY FIND IT HARD TO SURVIVE.

THE SOB INSTRUCTION IS THE ONLY PDP-11 INSTRUCTION THAT IS (1) A SINGLE OPERAND, (2) CAN BE RE. EATEDLY EXECUTED AT THE SAME PC AND, (3) CAN ESCAPE THIS REPETITIOUS LOOP.

HENCE, IT CAN BE POSSIBLE TO SOB A MOS CELL TO DEATH (OR AT LEAST BRAIN WASH HIM), AND TO SOB A CORE INTO OVER-HEATING (OR AT LEAST WARM DISCOMFORT).

THE SOB ROUTINE WILL BE LOADED AND CALLED WITH R0 SET EQUAL TO THE SOB CONSTANT "SOBK", R1 SET EQUAL TO THE COMPLEMENT OF A "SOB R0,.." INSTRUCTION "100776".

## SIMPLIFIED SOB EXAMPLE:

1\$:	SOB	R0,1\$	:SOB TILL R0 UNDERFLOWS
	MOV	R1,1\$	:WRITE COMPLEMENT OF SOB
	CMP	R1,1\$	:READ CHECK NOT SOB
	BEQ	2\$	:SKIP IF OK
	SOBFAIL		:TRAP REPORT ERROR
2\$:	SOBMOV1		:CODE TO GET SELF MOVED
	SOBMOV2		:FORWARD 1 WORD AND RUN AGAIN
	SOBMOV3		
	SOBMOV4		
	SOBMOV...		

THE VALUE OF THE SOB CONSTANT CAN BE FOUND AT SYMBOLIC LOCATION "SOBK" (TYPICAL 25 DECIMAL).

THIS TEST IS NOT IN THE NORMAL SCRIPT OF EXECUTION BUT MAY BE ADDED VIA THE APT E-TABLE, REFERENCE SYMBOLIC LOCATIONS "MKPAT", "MJPAT", "\$DDW2-5". FIELD SERVICE MODE COMMAND 8 IS THE NORMAL METHOD OF RUNNING THIS PATTERN.

## NOTE

THIS TEST IS NOT NORMALLY EXECUTED EXCEPT UNDER APT OR ACT. IT MAY BE INVOKED VIA FIELD SERVICE COMMAND 13 (KAMIKAZE MODE).



## 6.2.2.27 TEST 32 WRITE RECOVERY TEST

THIS TEST CAUSES A WRITE, READ, WRITE, READ, ... TO OCCUR IN MEMORY AND IF THE 1ST, 3RD, 5TH, ... READ IS BAD THE PROGRAM MAY BOMB OR IF THE 2ND, 4TH, 6TH, ... READ IS BAD THE PROGRAM WILL GRACEFULLY TYPE OUT THE ERROR.

## WRITE RECOVERY TEST

THIS TEST DIFFERS FROM OTHER TESTS IN THAT IT CONSISTS OF A SMALL TEST PROGRAM ACTUALLY RUNNING IN THE BANK UNDER TEST. THE PROGRAM IS SELF MODIFYING AND MAY BE DIFFICULT TO DEBUG. TO AID IN THE DEBUG, REMEMBER THAT THE BANK AND MARGIN ARE BEING DISPLAYED. THIS WILL ALLOW THE USER TO AT LEAST SEE WHICH MEMORY BANK FAILED.

THE TEST CONSISTS OF 1/2 OF THE BANK STORED WITH 'MOV R2, -(PC)' AND THE OTHER 1/2 CONTAINING '177667'. '177667' IS THE COMPLEMENT OF 'JMP (R0)' INSTRUCTION. R2 CONTAINS 'COM -(R1)' INSTRUCTION ON ENTRY TO THE BANK AND R1 CONTAINS THE HIGHEST TEST ADDRESS IN THAT BANK.

IF YOU UNDERSTAND THIS SO FAR THE REST IS EASY.

THE TEST EXECUTION IS AS FOLLOWS:

1. THE 'MOV R2, -(PC)' INSTRUCTION EXECUTES STORING THE CONTENTS OF R2 IN THE ADDRESS IT VACATED (DUE TO -(PC)).
2. SINCE R2 CONTAINS A 'COM -(R1)' INSTRUCTION IT COMPLEMENTS THE HIGHEST ADDRESS UNDER TEST. THIS ADDRESS CONTAINED '177667' SO AFTER THE COM -(R1) IT EQUALS 110 CLEVERLY THIS IS THE 'JMP (R0)' INSTRUCTION.
3. THIS SEQUENCE CONTINUES UNTIL THE 'MOV R2, -(PC)' INSTRUCTIONS REACH THE MIDDLE OF THE TEST BANK. THEN THE 'JMP (R0)' INSTRUCTION IS MET AND EXECUTED. R0 CONTAINED THE RETURN ADDRESS BACK TO TEST 13.
4. THESE STEPS ARE REPEATED FOR EACH BANK UNDER TEST.

## NOTE

THIS TEST IS NOT NORMALLY EXECUTED EXCEPT UNDER APT OR ACT. IT MAY BE INVOKED VIA FIELD SERVICE COMMAND 13 (KAMIKAZE MODE).

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

## 6.2.2.28 TEST 33 BRANCH GOBBLE TEST

THIS TEST LOADS A SMALL ROUTINE INTO THE MEMORY UNDER TEST. THE ROUTINE MOVES ITSELF ALONG IN MEMORY ONE WORD AFTER EACH PASS SO THAT WHEN IT REACHES THE END EVERY INSTRUCTION HAS EXECUTED FROM EVERY LOCATION WITH THE EXCEPTION OF THE BEGINNING AND END OF EACH TEST AREA.

THE BRANCH GOBBLE'S GENERAL FORMAT AFTER YOU ELIMINATE SETUP CODE AND CODE TO MOVE THE PROGRAM ALONG IS AS FOLLOWS.

BGTEST:	0		;TEST WORD
BRGOBB:	SEC		
	ADCB	BGTEST	;INC LOW BYTE
	BMI	1\$	;END LOOP AFTER 128 TIMES
	INCB	BGTEST+1	;INC HIGH BYTE
	BR	BRGOBB	;LOOP 128 TIMES
1\$:	BVS	2\$	;BRANCH IF V-BIT SET (SHOULD BE)
	ERROR		;ERROR TRAP
2\$:	CLV		;CLEAR V-BIT
	INCB	BGTEST	;INC HIGH BYTE ONE LAST TIME
	BCS	3\$	;BRANCH IF C-BIT SET (SHOULD NOT BE)
	BVC	3\$	;BRANCH IF V-BIT CLEAR (SHOULD NOT BE)
	BMI	4\$	;BRANCH IF N-BIT SET (SHOULD BE)
3\$:	ERROR		;ERROR TRAP
4\$:	RETURN		

THIS CODE ORIGINALLY CAME FROM THE PDP-11 FAMILY INSTRUCTION EXERCISER DZQKA-A. THE FIRST MOS MEMORIES WERE SUSCEPTIBLE TO THIS SECTION OF THAT DIAGNOSTIC AND IT HAS BEEN AN IMPORTANT MEMORY EXERCISER EVER SINCE.

## NOTE

THIS TEST IS NOT NORMALLY EXECUTED EXCEPT UNDER APT OR ACT. IT MAY BE INVOKED VIA FIELD SERVICE COMMAND 13 (KAMIKAZE MODE).

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

#### 6.2.2.29 TEST 34 SOFT ERROR TEST

##### RATIONALIZATION

-----

MOS CHIPS HAVE A FAILURE MODE IN WHICH THEY CAN RANDOMLY PICK OR DROP BITS. THIS IS CAUSED BY ALPHA PARTICLES BOMBARDING THE CELL. IF THE CELL IS VERY SMALL (AND THEY ARE) THEN THE ELECTRONS DISPLACED BY THE ALPHA PARTICLE ARE SUFFICIENT TO CAUSE THE CELL TO CHANGE FROM A ONE TO A ZERO OR FROM A ZERO TO A ONE.

THIS TEST IS CONTROLLED BY THE MAIN PROGRAM SO THAT IT IS USED TO CREATE A TEST OF 125252 AND 52525 ON ALTERNATE PASSES OF THE PROGRAM. THE CONFIGURATION TABLE IS USED TO FLAG BANKS THAT HAVE THE TEST INVALIDATED BECAUSE ANOTHER TEST WAS WRITTEN OVER THIS BACKGROUND.

THIS TEST IS NOTHING MORE THAN A CLEVER USE OF TEST 0.

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

#### 6.2.2.30 TEST 35 WORST CASE PARITY TEST

1. FORCE WRITE WRONG PARITY IN EACH 1K WORD BLOCK OF THE MEMORY UNDER TEST.
2. READ WITH PARITY TRAPPING ENABLED, MAKING SURE THAT A TRAP OCCURRS.
3. MAKE SURE ERROR ADDRESS BITS ARE SET CORRECTLY.
4. WRITE GOOD PARITY WITHOUT TRAPPING, AND MAKE SURE NO TRAP OCCURRS WHEN READ.

#### NOTE

THIS TEST IS RUN FOR PARITY MEMORY WHICH IS NOT CONTROLLED BY THE SAME CSR AS THE PROGRAM.

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

## 6.2.2.31

## TEST 36 CORRECTION CODE TEST

1. WRITE CSR WITH CHECK BITS TO CORRECT BIT 0 OF THE FIRST ADDRESS IN A 16K BANK FROM A 0 TO A 1 WITH DIAG MODE.
2. WRITE AUT WITH DATA OF 0'S.
3. READ AUT FOR CORRECTION OF BIT 0 FROM A 0 TO A 1.
4. REPEAT 1-3 FOR ALL 16 DATA BITS.
5. REPEAT 1-4 FOR DATA OF ONES SO THAT A CORRECTION WILL OCCUR FROM A 1 TO A ZERO.

THIS TEST CHECKS TO SEE THAT THE EDC CHIP CAN CORRECT SINGLE BIT ERRORS FOR ALL 16K DATA BITS FROM A 1 TO A 0 AND VISA VERSA.

## NOTE

THIS TEST IS ONLY RUN FOR THE MS11-P

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

6.2.2.32 TEST 37 CHECK ECC DISABLE TEST

1. WRITE CSR WITH ECC DISABLE, DIAG MODE, AND SBE CHECK BITS OF 000010.
2. WRITE AUT WITH DATA OF ZERO'S. THIS SHOULD WRITE CHECK BITS TO MEMORY.
3. READ AUT FOR DATA OF ZEROS INSURING NO CORRECTION WAS MADE.

NOTE

THIS TEST IS RUN ON THE MS11-P ONLY.

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

6.2.2.33 TEST 41 ADDRESS TO CSR ON DBE TEST

1. WRITE CSR WITH ECC DISABLE, DIAG MODE, AND DOUBLE BIT ERROR  
CHECK BITS OF 010011
2. WRITE AUT WITH DATA OF ZEROS CREATING A DBE.
3. READ AUT TO DETECT DBE AND TO CLOCK ADDRESS INTO CSR
4. READ CSR FOR CORRECT ADDRESS IN BITS 5-11.
5. INCREMENT ADDRESS BY 1K AND REPEAT 1-4 UNTIL 16K IS DONE.

THIS TEST INSURES THAT THE CORRECT ADDRESS APPEARS IN CSR BITS 5-11 ON A DBE

NOTE

THIS TEST IS RUN ON A MS11-P ONLY.

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

#### 6.2.2.34 TEST 42 EXTENDED ADDRESS TO CSR ON ERROR TEST

1. WRITE CSR WITH SBE CHECK BITS OF 000010 WITH DIAGNOSTIC MODE.
2. WRITE LOW ADDRESS IN A 16K BANK WITH DATA OF ZEROS CREATING A SBE.
3. CLEAR THE CSR.
4. READ ADDRESS TO DETECT SBE.
5. READ CSR FOR CORRECT ADDRESS AND THE SBE INDICATOR BIT #4.
6. ENABLE CSR BIT 14 TO CHECK THE EXTENDED ADDRESS BITS.
7. READ CSR FOR CORRECT ADDRESS BITS
8. REPEAT 1-7 WITH A TEST ADDRESS THAT IS THE HIGHEST IN A 16K BANK.

THIS TEST CHECKS TO SEE THAT THE CORRECT ADDRESS BITS APPPEAR IN THE CSR.  
THIS IS ALSO REPEATED FOR THE EXTENDED ADDRESS FUNCTION IN THE CSR.

#### NOTE

THIS TEST IS ONLY RUN FOR THE MS11-P



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

#### 6.2.2.35 TEST 43 WRITE BYTE TEST

1. WRITE CSR TO DIAG MCDE WITH CHECK BITS OF 001100. THESE CORRESPOND TO DATA OF ZEROS.
2. WRITE FIRST AUT WITH DATA OF ONE IN BIT ZERO. THE WRITE EFFECTIVELY CREATES A SBE IN BYTE 0.
3. CLEAR THE CSR
4. WRITE BYTE 1 OF THE AUT WITH DATA OF ALL ONES.
5. READ CSR TO CHECK FOR SBE INDICATION.
6. WRITE THE CSR TO DIAG MODE.
7. READ THE AUT TO CHECK FOR THE CORRECT DATA -- ALL ONES IN HIGH BYTE AND ALL ZEROS IN LOW BYTE.
8. READ THE CSR TO CHECK FOR CORRECT CHECK BITS CORRESPONDING TO THE DATA READ IN (7). THESE CHECK BITS ARE 000110.
9. REPEAT (1)-(8) THIS TIME CREATING AN ERROR IN BYTE 1 (2) AND WRITING BYTE 0 IN (4).

THIS TEST CHECKS TO SEE THAT A SBE WILL BE CORRECTED DURING THE READ PORTION OF THE BYTE WRITE AND THAT CORRECT CHECKBITS WILL BE GENERATED ON THE WRITE.

#### NOTE

THIS TEST IS ONLY RUN FOR THE MS11-P

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

#### 6.2.2.36 TEST 44 SHIFTING CHECKBITS THROUGH THE CSR TEST

1. WRITE CSR TO DIAG MODE TO ENABLE CHECKBIT REGISTER.
2. WRITE CSR WITH CHECK BITS OF 000001, ECC DISABLE AND DIAG MODE.
3. WRITE MEMORY WITH DATA OF ZEROS. THIS SHOULD WRITE THE CHECK BITS INTO MEMORY.
4. COMPLEMENT CHECK BITS PATTERN AND WRITE CSR AS IN (2).
5. READ CSR FOR COMPLMENT CHECK BIT PATTERN.
6. READ MEMORY TO READ CHECK BITS WRITTEN IN (2) INTO CSR.
7. READ CSR FOR CORRET CHECK BITS WRITTEN IN (2).
8. SHIFT CHECK BIT PATTERN AND REPEAT (1-7) TILL CSR BITS 5-10 ARE DONE.
9. COMPLEMENT CHECK BIT PATTERN IN (2) AND REPEAT (1-8) SHIFTING A ZERO THROUGH A FIELD OF ONES.

THIS TEST CHECKS THE ABILITY TO READ CHECK BITS FROM THE CSR TO MEMORY AND BACK. THE TEST IS DONE TWICE. ONCE SHIFTING A FIELD OF A ONE THROUGH A FIELD OF ZEROS AND A ZERO THROUGH A FIELD OF ONES. THIS TESTS THE CHECKBIT/SYNDROME BIT REGISTER AND CHECK BIT RAM'S

#### NOTE

THIS TEST IS ONLY RUN FOR THE MS11-P

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

#### 6.2.2.37 TEST 45 SYNDROME BITS TO THE CSR ON A DBE TEST

1. WRITE CSR WITH DIAG MODE TO ENABLE CHECK/SYNDROME BIT REGISTER.
2. WRITE CSR WITH DBE CHECK BITS OF 110011 WITH DIAG MODE.
3. WRITE MEMORY WITH DATA OF ZEROS CREATING A DBE.
4. CLEAR CSR.
5. READ MEMORY TO DETECT DBE.
6. READ CSR FOR UNCORRECTABLE ERROR INDICATOR.
7. WRITE CSR TO DIAG MODE TO READ SYNDROME BITS INTO CSR.
8. READ CSR FOR CORRECT SYNDROME BITS OF 111111.
9. REPEAT (1-8) WITH MULTIPLE BIT ERROR CHECK BITS OF 111100 AND CORRESPONDING SYNDROME BITS OF 110000.

THIS TEST CHECKS THE ABILITY OF THE CSR TO DETECT A DBE AND READ FOR THE PROPER SYNDROME BITS GENERATED BY THE EDC CHIP. THIS TEST IS THEN REPEATED WITH CHECK BITS CORRESPONDING TO A MULTIPLE BIT ERROR.

#### NOTE

THIS TEST IS ONLY RUN FOR THE MS11-P

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

#### 6.2.2.38 TEST 46 CHECK SINGLE BIT ERRORS WITH ECC DISABLED TEST

1. WRITE CSR WITH CHECK BITS TO CORRECT BIT 0 OF THE FIRST ADDRESS IN A 16K BANK FROM A 0 TO A 1 WITH DIAG MODE AND ECC DISABLED.
2. WRITE AUT WITH DATA OF 0'S THUS CREATING A SBE.
3. WRITE THE CSR TO ECC DISABLE.
4. READ AUT TO DETECT SBE.
5. CHECK TO SEE THAT NO TRAP OCCURED.
6. READ CSR TO SEE THAT UNCORRECTABLE ERROR (CSR15) IS SET.
7. REPEAT 1-6 FOR ALL 16 DATA BITS.
8. REPEAT 1-7 FOR DATA OF ONES SO THAT A CORRECTION WILL OCCUR FROM A 1 TO A ZERO.
9. REPEAT 1-8 EXCEPT IN STEPS (3) THE CSR IS WRITTEN TO ECC DISABLE AND BUS PBL ENABLE AND (5) WE CHECK FOR TRAPS.

THIS TEST CHECKS TO SEE THAT SBE ARE TREATED A UNCORRECTABLE ERRORS WITH ECC DISABLE. THE TEST IS REPEATED 2 TIMES, ONCE WITH TRAPS DISABLED AND AGAIN WITH IT ENABLED. THIS IS DONE FOR ALL 16 POSSIBLE SBE CONDITIONS.

#### NOTE

THIS TEST IS ONLY RUN FOR THE MS11-P

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

6.2.2.39 TEST 47 NO CSR UPDATE ON SBE WITH EXSISTING DBE TEST

1. WRITE THE CSR TO DIAG MODE TO ENABLE CHECKBIT/SYNDROME BIT REGISTER.
2. WRITE THE CSR WITH DBE CHECK BITS OF 110011 AND DIAG MODE.
3. WRITE MEMORY WITH DATA OF ZEROS CREATING A DBE.
4. WRITE CSR WITH SBE CHECK BITS OF 000010 AND DIAG MODE.
5. WRITE MEMORY 4K ABOVE ADDRESS IN (3) CREATING A SBE.
6. CLEAR CSR.
7. READ MEMORY WITH ADDRESS IN (3) TO DETECT DBE.
8. READ CSR FOR CORRECT ADDRESS AND UNCORRECTABLE ERROR INDICATOR
9. READ MEMORY WITH ADDRESS IN (5) TO DETECT SBE.
10. READ CSR FOR SBE INDICATOR AND NO CHANGE IN DBE STATUS IN CSR IN (8)

THIS TEST CHECKS TO SEE THAT NO UPDATE WILL OCCUR IN THE CSR WITH A SBE IN MEMORY WHEN A DBE ALREADY EXISTS.

NOTE

THIS TEST IS ONLY RUN FOR THE MS11-P

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 91

3754  
3755  
3756  
3757  
3758  
3759  
3760  
3761  
3762  
3763  
3764  
3765  
3766

6.2.2.40 TEST 999 NULL TEST

THIS IS AN INSTANT RETURN ADDED TO PRESERVE THE SOFTWARE STRUCTURE.

THIS TEST REPLACES ANY REAL TESTS WHEN THE APT E-TABLE DOES NOT  
SPECIFY A TEST TO BE RUN.

## 7.0 PROGRAM FEATURES

## 7.1 FAST DATA ACCESS RATES

ONE OF THE MAIN AREAS OF CONCERN IN TESTING MEMORY IN SYSTEMS ENVIRONMENTS IS SPEED. ONE OF THE PRIME REASONS THAT SYSTEM PROGRAMS LIKE RSTS, IAS AND MUMPS CAN CRASH DUE TO MEMORY FAILURES NOT DETECTABLE BY MEMORY DIAGNOSTICS (0-124K, 0-2 MEG, ETC.) IS BECAUSE OF MULTIPLE NPR DEVICES CONTENDING FOR THE BUS. AFTER SOME DELAY A NPR DEVICE BECOMES BUS MASTER AND DOES SEVERAL MEMORY TRANSFERS AT MEMORY DATA RATES.

ON THE OTHER HAND MOST DIAGNOSTICS WHEN WRITING READING AND/OR CHECKING PATTERNS SPEND MOST OF THEIR TIME FETCHING INSTRUCTIONS AND OPERANDS OUT OF THEIR PROGRAM SPACE AND PROPORTIONALLY LITTLE TIME ACCESSING THE MEMORY UNDER TEST.

THIS DIAGNOSTIC'S ERROR DETECTING ABILITIES HAVE BEEN OPTIMIZED AROUND THE PRIMARY DESIGN CRITERIA OF SPEED. TO THIS END THE FOLLOWING STEPS HAVE BEEN TAKEN.

## 7.1.1 FAST CITY

UTILIZATION OF MEMORY MANAGEMENT REGISTERS AS NON MEMORY BUS, NON UNIBUS, BIPOLAR MEMORY. SINCE USER MODE IS ONLY USED FOR RELOCATION AND DATA SPACE IS NEVER USED, THEN SUBROUTINES CAN BE EXECUTED FROM THE UIPAR'S, UDPAR'S, KDPAR'S, SDPAR'S AND WITH SOME BIT PATTERN RESTRICTIONS THE UIPDR'S, UDPDR'S, KDPDR'S, AND SDPDR'S.

THE PROGRAM RUNS IN KERNEL MODE AND PATTERNS ARE EXECUTED IN SUPERVISOR MODE FOR MAPPING PURPOSES. ALL CORE PATTERNS AND SOME MOS PATTERNS ARE SUBROUTINES THAT ARE MOVED TO THIS BIPOLAR REGION REFERRED TO IN THE PROGRAM AS FAST CITY.

## NOTE

18-BIT PDP-11'S CANNOT EXECUTE FROM THE PAR'S BECAUSE THEIR PAR'S ARE ONLY 12 BITS WIDE; THEY ALSO HAVE NO SUPERVISOR MODE. THEREFORE, ALL PATTERNS ARE EXECUTED IN MEMORY, USING USER MODE (REFERENCE SECTION 7.5).

## 7.1.2 SOB'S

3823 UTILIZATION OF THE FULL PDP-11 INSTRUCTION SET TO SPEED PATTERN  
3824 ALGORITHMS (PRINCIPALLY THE SOB).  
3825  
3826  
3827  
3828  
3829

3830 7.1.3 CACHE  
3831

3832 CACHE IS USED BETWEEN PATTERN TESTS TO DECREASE PROGRAM PASS TIMES.  
3833 CACHE CAN BE DEFEATED BY THE OPERATOR (REFERENCE SECTION 2.4.3.1).  
3834  
3835

3836 7.2 BANK ZERO TESTING  
3837

3838 BANK ZERO HAS BEEN TRADITIONALLY NEGLECTED BY MEMORY DIAGNOSTICS FOR  
3839 THE FOLLOWING REASON.  
3840

3841 THE VECTOR SPACE EXISTS THERE AND ALL TRAPS MUST NOT ACCESS TEST  
3842 PATTERN DATA. IF THE AREA IS TESTED THE DIAGNOSTIC MUST NOT USE ANY  
3843 TRAPS, AND IT IS AGAINST THE RULES FOR POWER TO FAIL.  
3844

3845 SYSTEMS WITH MEMORY MANAGEMENT CAN OVERCOME THIS BECAUSE ALL TRAPS ARE  
3846 TO KERNEL VIRTUAL SPACE EVEN IF THE POWER SHOULD FAIL (CAUTION MUST BE  
3847 OBSERVED BECAUSE POWER UP GOES TO PHYSICAL ADDRESS 24 (BECAUSE THE  
3848 MEMORY MANAGEMENT UNIT COMES UP OFF)).  
3849

3850 HOWEVER, CATCH 22 IS THAT THE DIAGNOSTIC IS NOT APT COMPATIBLE IN THIS  
3851 MODE BECAUSE APT ACCESSES PHYSICAL MEMORY LOCATIONS.  
3852

3853 THE PDP-11/44 CAN OVER COME THIS BECAUSE THE UNIBUS MAP CAN FOOL APT.  
3854

3855 BECAUSE OF THE PREVIOUS ARGUMENTS THIS PROGRAM DOES NOT RELOCATE IN  
3856 THE TRUE SENSE OF THE WORD (I.E. NO POSITION INDEPENDENT CODE WAS  
3857 WRITTEN (AT LEAST NOT ON PURPOSE)), BUT RATHER THIS PROGRAM MOVES AND  
3858 REMAPS (HEREAFTER REFERRED TO AS RELOCATES). THIS ENABLES THE  
3859 COMPLETE TESTING OF BANK ZERO OR ANY OTHER PROGRAM SPACE OR PRIVILEGED  
3860 SPACE EXACTLY AS ALL OTHER BANKS ARE TESTED. (THE CONDITIONAL TEST TO  
3861 SEE IF A BANK IS PROTECTED IS COMPLEMENTED WHEN RELOCATED).  
3862  
3863

3864 NOTE  
3865

3866 THE PROGRAM WILL RELOCATE ONLY IN THE  
3867 FIRST PASS UNDER APT; AFTER THIS, THE  
3868 PROGRAM WILL REMAIN FIXED IN BANKS 0 AND  
3869 1.  
3870

3871 7.3 MEMORY CONFIGURATION MAP  
3872  
3873  
3874  
3875  
3876



**EXAMPLE:**

16K BANKS

DISPLAYED ARE BANKS 0-73 OCTAL (2 MEG WORDS). IF THE FAT TERMINAL SWITCH WAS SET (REFERENCE SECTION 2.4.1) THEN ALL BANKS (0-167) WOULD BE SHOWN. IF THIS WAS AN 18-BIT PDP-11 (EG - 11/34), ONLY BANKS 0-7 WOULD BE PRINTED.

THE FIELDS:

THE SIZING ROUTINE COULD NOT WRITE ZEROS AND ONES IN BANKS 10 11, HENCE THEY ARE MARKED AS BAD WITH X'S.

THERE IS INTERLEAVING ON BANKS 20-37, WITH CSR 2 (172104) CONTROLLING THE ADDRESS BIT 1 NON-ASSERTED ADDRESSES, AND CSR 3 (172106) CONTROLLING THE ADDRESS BIT 1 ASSERTED ADDRESSES.

## MEMTYPE:

BANKS 0-7 ARE MEMORY TYPE L (MS-11L), AND BANKS 10-37 ARE MEMORY TYPE M (MS11-M) AND BANKS 40-77 ARE MEMORY TYPE P(MS11-P). BANKS 100-167 DO NOT EXIST.

**CSR:**

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

BANKS 0-7 ARE ASSIGNED TO CSR 172100, 10-17 TO CSR 172102,  
AND 20-37 TO INTERLEAVED CSR'S 172104 AND 172106 AND BANKS  
40-77 ARE ASSIGNED CSR 17210.

#### PROTECT:

BANKS 0 AND 1 ARE PROTECTED BECAUSE THEY ARE PROGRAM SPACE.  
BANK 0 AND 1 CAN ALSO BE PROTECTED BECAUSE THEY ARE IN THE  
BOTTOM 16K OF AN MS11-M CSR. THE PROTECTION IS HIERARCHICAL  
AND PROGRAM SPACE OVERSHADOWS MS11-M PROTECTION. BANKS 0  
AND 1 WILL NOT BE TESTED UNTIL THE PROGRAM RELOCATES. IF

ANY BANK IS PROTECTED BY MS11-M AND NOT BECAUSE  
IT IS IN PROGRAM SPACE IT WILL HAVE AN 'I' TYPED IN THIS  
ROW. THIS IS TO POINT OUT WHERE THE PROTECTED BANKS START  
FOR EACH ECC CSR. NOTE THE 'P' AT BANK 30; THIS POINTS OUT  
THE 'SHADOW' PROTECTION WHICH OCCURS WHEN TWO MS11-M  
MEMORIES ARE INTERLEAVED. THEREFORE, BANK 30 WILL NOT BE  
TESTED UNTIL THE PROGRAM HAS RELOCATED.

#### 7.4 EVERYTHING YOU'VE ALWAYS WANTED TO KNOW ABOUT SUPERMAC ...

SUPER-MAC IS A SET OF STRUCTURED PROGRAMMING MACROS THAT ALLOWS  
PROGRAMS TO BE WRITTEN IN A HIGH LEVEL, EASILY UNDERSTOOD LANGUAGE.

AS A GENERAL RULE, MOST SUPER-MAC STATEMENTS CAN BE SINGLE-LINE  
STATEMENTS OR MULTIPLE-LINE (NESTED) BLOCK STATEMENTS. A SINGLE-LINE  
STATEMENT MUST BE COMPLETED ON ONE SOURCE LINE; NO CONTINUATION LINES  
ARE ALLOWED. SINGLE-LINE STATEMENTS SHOULD BE AS SHORT AND SIMPLE AS  
POSSIBLE. COMMENTS MAY ALSO BE INCLUDED ON A SOURCE LINE. ALL THE  
GENERAL RULES, CONDITIONS, ETC., THAT GOVERN MACRO-11 ALSO GOVERN  
SUPER-MAC. SPACING ON A SOURCE LINE IS VERY IMPORTANT. THE ELEMENTS  
SHOULD BE SEPARATED BY A COMMA OR A SPACE. TABS SHOULD NEVER BE USED  
FOR SPACING. FOR EXAMPLE: THE EXPRESSION A+B IS INTERPRETED  
DIFFERENT THAN A + B.

ALL THE CONDITIONAL STATEMENTS CAN BE WRITTEN AS MULTIPLE-LINE NESTED  
BLOCKS. EACH LEVEL OF NESTING WITHIN A BLOCK MUST BE TERMINATED WITH  
AN ASSOCIATED END STATEMENT. EACH LEVEL OF NESTING SHOULD BE INDENTED  
TWO SPACES.

USER WRITTEN MACROS OR ASSEMBLY LANGUAGE INSTRUCTIONS MAY BE INCLUDED  
IN A PROGRAM IF DESIRED. AS A DEBUGGING AID, IF THE SYMBOL LST\$\$ IS  
DEFINED, IT WILL CAUSE GENERATED CODE AND LABELS TO BE LISTED. ALL  
PROGRAMS MUST BEGIN WITH THE MACRO CALL SMACIT. THIS CALL INITIALIZES  
SUPER-MAC. ALL LEGAL PDP-11 SOURCE AND DESTINATION OPERANDS ARE LEGAL  
IN SUPER-MAC.

```

3985
3986      7.4.1 SAMPLE SOURCE FILE -
3987      .ENABL ABS
3988      .ENABL AMA
3989      .MCALL .SUPER
3990      .SUPER
3991      .LST$$=0
3992      BIT5=40
3993      A:      0
3994      B:      0
3995      C:      0
3996      D:      0
3997      E:      0
3998      F:      0
3999      G:      0
4000      H:      0
4001      I:      0
4002      J:      0
4003      .PAGE
4004      ;LET EXAMPLES
4005      LET RO := A
4006      LET B := C + D
4007      LET E := F + 1
4008      LET G := H + 2
4009      LET J := J + 01
4010      LET A :B= B
4011      ;IF EXAMPLES
4012      IF A IS TRUE
4013      MOV 23,D
4014      END ;OF IF A
4015      IF B IS FALSE
4016      MOV 34,E
4017      END ;OF IF B
4018      IF A EQ B THEN LET C := D
4019      IF A LT B
4020      MOV C,D
4021      ELSE
4022      MOV E,D
4023      END ;OF IF A
4024      IF A EQ B AND C NE D
4025      MOV F,G
4026      END ;OF IF A
4027      IF A EQ B OR C NE D
4028      MOV F,G
4029      END ;OF IF A
4030      IFB A EQ B AND C EQ 1
4031      MOV H,J
4032      ELSE
4033      MOV E,J
4034      END ;OF IFB A
4035      IFB A EQ B ANDB C EQ 1
4036      MOV H,J
4037      ELSE
4038      MOV E,J

```

```

4040      END ;OF IFB A
4041      IF RESULT IS EQ
4042      MOV A,B
4043      END ;OF IF RESULT
4044      IF BITS SET.IN A
4045      MOV B,C
4046      END ;OF IF BITS
4047      IF BITS OFF.IN A
4048      MOV C,D
4049      END ;OF IF BITS
4050 ;ON.ERROR IS LIKE AN IF STATEMENT ON THE C-BIT
4051 ;ON.ERROR EXAMPLES
4052 ON.ERROR
4053     MOV A,B
4054 ELSE
4055     MOV C,B
4056 END ;OF ON.ERROR
4057 ON.NOERROR
4058
4059     MOV C,B
4060 ELSE
4061     MOV A,B
4062 END ;OF ON.NOERROR
4063 ON.ERROR THEN LET A :B= B
4064 ;FOR EXAMPLES
4065 FOR I := -5 TO 23
4066     INC A
4067 END ;OF FOR I
4068 FOR R0 := 0 TO 140 BY 4
4069     DEC A(R0)
4070 END ;OF FOR R0
4071 FOR I := 133 DOWNT0 3 BY 2
4072     ADD A,B
4073 END ;OF FOR I
4074 ;BEGIN EXAMPLES
4075 BEGIN ALPHA
4076     FOR R0 := 0 TO 167
4077         MOV B A(R0),B
4078         IF B LT 0 THEN LEAVE ALPHA
4079     END ;OF FOR R0
4080     FOR R0 := 400 TO 567
4081         IF B GE 0 THEN LEAVE ALPHA
4082     END ;OF FOR R0
4083 END ALPHA
4084 ;$RETURN EXAMPLES
4085 $RETURN
4086 $RETURN ERROR
4087 $RETURN NOERROR
4088 ;CASE EXAMPLES
4089 MOV A,R0
4090 CASE R0
4091     A
4092

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 98

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

B  
C  
D  
E  
F  
END ;OF CASE R0  
.END

7.4.2 SAMPLE LISTING FILE (WITH NO EXPANDED MACROS) - -  
.MAIN. MACRO M1111 01-APR-79 16:41 PAGE 2

1	000000				.ENABL ABS
2					.ENABL AMA
3					.MCALL .SUPER
4	000000				.SUPER
5					.LST\$\$=0
6		000040			BIT5=40
7	000000	000000	A:	0	
8	000002	000000	B:	0	
9	000004	000000	C:	0	
10	000006	000000	D:	0	
11	000010	000000	E:	0	
12	000012	000000	F:	0	
13	000014	000000	G:	0	
14	000016	000000	H:	0	
15	000020	000000	I:	0	
16	000022	000000	J:	0	

.MAIN. MACRO M1111 01-APR-79 16:41 PAGE 3

18					;LET EXAMPLES
19	000024				LET R0 := A
20	000030				LET B := C + D
21	000044				LET E := F + 1
22	000056				LET G := H + 2
23	000072				LET J := J + 01
24	000100				LET A :B= B
25					;IF EXAMPLES
26	000106				IF A IS TRUE
27	000114	012737	000023	000006	MOV 23,D
28	000122				END ;OF IF A
29	000122				IF B IS FALSE
30	000130	012737	000034	000010	MOV 34,E
31	000136				END ;OF IF B
32	000136				IF A EQ B THEN LET C := D
33	000154				IF A LT B
34	000164	013737	000004	000006	MOV C,D
35	000172				ELSE

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 99

4148	36	000174	013737	000010	000006
4149	37	000202			
4150	38	000202			
4151	39	000222	013737	000012	000014
4152	40	000230			
4153	41	000230			
4154	42	000250	013737	000012	000014
4155	43	000256			
4156	44	000256			
4157	45	000276	013737	000016	000022
4158	46	000304			
4159	47	000306	013737	000010	000022
4160	48	000314			
4161	49	000314			
4162	50	000334	013737	000016	000022
4163	51	000342			
4164	52	000344	013737	000010	000022
4165	53	000352			
4166	54	000352			
4167	55	000354	013737	000000	000002
4168	56	000362			
4169	57	000362			
4170	58	000372	013737	000002	000004
4171	59	000400			
4172	60	000400			
4173	61	000410	013737	000004	000006
4174	62	000416			
4175	63				
4176	64				
4177	65	000416			
4178	66	000420	013737	000000	000002
4179	67	000426			
4180	68	000430	013737	000004	000002
4181	69	000436			
4182	70	000436			
4183	71	000440	013737	000004	000002
4184	72	000446			
4185	73	000450	013737	000000	000002
4186	74	000456			

```

MOV E,D
END ;OF IF A
IF A EQ B AND C NE D
MOV F,G
END ;OF IF A
IF A EQ B OR C NE D
MOV F,G
END ;OF IF A
IFB A EQ B AND C EQ 1
MOV H,J
ELSE
MOV E,J
END ;OF IFB A
IFB A EQ B ANDB C EQ 1
MOV H,J
ELSE
MOV E,J
END ;OF IFB A
IF RESULT IS EQ
MOV A,B
END ;OF IF RESULT
IF BITS SET.IN A
MOV B,C
END ;OF IF BITS
IF BITS OFF.IN A
MOV C,D
END ;OF IF BITS

```

;ON.ERROR IS LIKE AN IF STATEMENT ON THE C-BIT  
;ON.ERROR EXAMPLES

```

ON.ERROR
MOV A,B
ELSE
MOV C,B
END ;OF ON.ERROR
ON.NOERROR
MOV C,B
ELSE
MOV A,B
END ;OF ON.NOERROR

```

.MAIN. MACRO M1111 01-APR-79 16:41 PAGE 3-1

4188					
4189					
4190					
4191					
4192	75	000456			
4193	76				
4194	77	000466			
4195	78	000474	005237	000000	
4196	79	000500			
4197	80	000514			
4198	81	000516	005360	000000	
4199	82	000522			
4200	83	000534			

```

ON.ERROR THEN LET A :B= B
;FOR EXAMPLES
FOR I := -5 TO 23
INC A
END ;OF FOR I
FOR RO := 0 TO 140 BY 4
DEC A(RO)
END ;OF FOR RO
FOR I := 133 DOWNT0 3 BY 2

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 100

4202	84	000542	063737	000000	000002	ADD A,B
4203	85	000550				END ;OF FOR I
4204	86					;BEGIN EXAMPLES
4205	87	000566				BEGIN ALPHA
4206	88	000566				FOR RO := 0 TO 167
4207	89	000570	116037	000000	000002	MOVB A(RO),B
4208	90	000576				IF B LT 0 THEN LEAVE ALPHA
4209	91	000604				END ;OF FOR RO
4210	92	000614				FOR RO := 400 TO 567
4211	93	000620				IF B GE 0 THEN LEAVE ALPHA
4212	94	000626				END ;OF FOR RO
4213	95	000636				END ALPHA
4214	96					;SRETURN EXAMPLES
4215	97	000636				\$RETURN
4216	98	000640				\$RETURN ERROR
4217	99	000644				\$RETURN NOERROR
4218	100					;CASE EXAMPLES
4219	101	000650	013700	000000		MOV A,RO
4220	102	000654				CASE RO
4221	103	000664	000000			A
4222	104	000666	000002			B
4223	105	000670	000004			C
4224	106	000672	000006			D
4225	107	000674	000010			E
4226	108	000676	000012			F
4227	109	000700				END ;OF CASE RO
4228	110					
4229	111		000001			.END

7.4.3 SAMPLE LISTING FILE (WITH EXPANDED MACROS) - -  
 .MAIN. MACRO M1111 01-APR-79 16:10 PAGE 2

4234	1	000000				.ENABL ABS
4235	2					.ENABL AMA
4236	3					.MCALL .SUPER
4237	4	000000				.SUPER
4238	5		000000			LST\$\$=0
4239	6		000040			BIT5=40
4240	7	000000	000000		A:	0
4241	8	000002	000000		B:	0
4242	9	000004	000000		C:	0
4243	10	000006	000000		D:	0
4244	11	000010	000000		E:	0
4245	12	000012	000000		F:	0
4246	13	000014	000000		G:	0
4247	14	000016	000000		H:	0
4248	15	000020	000000		I:	0
4249	16	000022	000000		J:	0

.MAIN. MACRO M1111 01-APR-79 16:10 PAGE 3

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 101

4255	18			
4256	19	000024	013700	000000
4257		000024		
4258	20	000030		
4259		000030	013737	000004
4260		000036	063737	000006
4261	21	000044		
4262		000044	013737	000012
4263		000052	005237	000010
4264	22	000056		
4265		000056	013737	000016
4266		000064	062737	000002
4267	23	000072		
4268		000072	062737	000001
4269	24	000100		
4270		000100	113737	000002
4271	25			
4272	26	000106		
4273		000106	005737	000000
4274		000112	001403	
4275	27	000114	012737	000023
4276	28	000122		
4277		000122		
4278	29	000122		
4279		000122	005737	000002
4280		000126	001003	
4281	30	000130	012737	000034
4282	31	000136		
4283		000136		
4284	32	000136		
4285		000136	023737	000000
4286		000144	001003	
4287		000146	013737	000006
4288		000154		
4289	33	000154		
4290		000154	023737	000000
4291		000162	002004	
4292	34	000164	013737	000004
4293	35	000172		
4294		000172	000403	
4295		000174		
4296	36	000174	013737	000010
4297	37	000202		
4298		000202		
4299	38	000202		
4300		000202	023737	000000
4301		000210	001007	
4302		000212	023737	000004
4303		000220	001403	
4304	39	000222	013737	000012
4305	40	000230		

```

;LET EXAMPLES
LET R0 := A
MOV A,R0
LET B := C + D
MOV C,B
ADD D,B
LET E := F + 1
MOV F,E
INC E
LET G := H + 2
MOV H,G
ADD 2,G
LET J := J + 01
ADD 01,J
LET A := B = B
MOVB B,A

;IF EXAMPLES
IF A IS TRUE
TST A
BEQ L0
MOV 23,D
END ;OF IF A

L0:
IF B IS FALSE
TST B
BNE L1
MOV 34,E
END ;OF IF B

L1:
IF A EQ B THEN LET C := D
CMP A,B
BNE L2
MOV D,C

L2:
IF A LT B
CMP A,B
BGE L3
MOV C,D
ELSE
BR L4

L3:
MOV E,D
END ;OF IF A

L4:
IF A EQ B AND C NE D
CMP A,B
BNE L5
CMP C,D
BEQ L5
MOV F,G
END ;OF IF A

```



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 102

4307		000230			L5:	
4308	41	000230				IF A EQ B OR C NE D
4309		000230	023737	000000		CMP A,B
4310		000236	001404			BEQ L6
4311		000240	023737	000004		CMP C,D
4312		000246	001403			BEQ L7

.MAIN. MACRO M1111 01-APR-79 16:10 PAGE 3-1

4313						
4314						
4315						
4316						
4317						
4318						
4319	42	000250	013737	000012	000014	L6: MOV F,G
4320	43	000256				END ;OF IF A
4321		000256				L7: IFB A EQ B AND C EQ 1
4322	44	000256				CMPB A,B
4323		000256	123737	000000	000002	BNE L10
4324		000264	001010			CMP C, 1
4325		000266	023727	000004	000001	BNE L10
4326		000274	001004			MOV H,J
4327	45	000276	013737	000016	000022	ELSE
4328	46	000304				BR L11
4329		000304	000403			
4330		000306				L10: MOV E,J
4331	47	000306	013737	000010	000022	END ;OF IFB A
4332	48	000314				L11: IFB A EQ B AND B C EQ 1
4333		000314				CMPB A,B
4334	49	000314				BNE L12
4335		000314	123737	000000	000002	CMPB C, 1
4336		000322	001010			BNE L12
4337		000324	123727	000004	000001	MOV H,J
4338		000332	001004			ELSE
4339	50	000334	013737	000016	000022	BR L13
4340	51	000342				
4341		000342	000403			L12: MOV E,J
4342		000344				END ;OF IFB A
4343	52	000344	013737	000010	000022	L13: IF RESULT IS EQ
4344	53	000352				BNE L14
4345		000352				MOV A,B
4346	54	000352				END ;OF IF RESULT
4347		000352	001003			L14: IF BITS SET.IN A
4348	55	000354	013737	000000	000002	BIT BITS,A
4349	56	000362				BEQ L15
4350		000362				MOV B,C
4351	57	000362				END ;OF IF BITS
4352		000362	032737	000040	000000	L15: IF BITS OFF.IN A
4353		000370	001403			BIT BITS,A
4354	58	000372	013737	000002	000004	
4355	59	000400				
4356		000400				
4357	60	000400				
4358						
4359		000400	032737	000040	000000	

CZMSPA0 MS11-L/M/P MEMORY DIAG. M/CRO M1113 26-APR-82 09:41 PAGE 103

4361	61	000410	013737	000004	000006	MOV C,D
4362						END ;OF IF BITS
4363	62	000416				L16:
4364		000416				;ON.ERROR IS LIKE AN IF STATEMENT ON THE C-BIT
4365	63					;ON.ERROR EXAMPLES
4366	64					ON.ERROR
4367	65	000416				BCC L17
4368		000416	103004			MOV A,B
4369	66	000420	013737	000000	000002	ELSE
4370	67	000426				BR L20
4371		000426	000403			L17:
4372		000430				MOV C,B
4373	68	000430	013737	000004	000002	END ;OF ON.ERROR
4374						L20:
4375	69	000436				ON.NOERROR
4376		000436				
4377	70	000436				
4378						
4379						
4380						

.MAIN. MACRO M1111 01-APR-79 16:10 PAGE 3-2

4381						BCS L21
4382						MOV C,B
4383						ELSE
4384						BR L22
4385	71	000440	013737	000004	000002	L21:
4386	72	000446				MOV A,B
4387		000446	000403			END ;OF ON.NOERROR
4388		000450				L22:
4389	73	000450	013737	000000	000002	ON.ERROR THEN LET A :B= B
4390	74	000456				BCC L23
4391		000456				MOVB B,A
4392	75	000456				L23:
4393		000456	103003			;FOR EXAMPLES
4394		000460	113737	000002	000000	FOR I := -5 TO 23
4395		000466				MOV -5,I
4396	76					B0:
4397	77	000466	012737	177773	000020	INC A
4398		000466				END ;OF FOR I
4399		000474				INC I
4400	78	000474	005237	000000		CMP I, 23
4401	79	000500				BLE B0
4402		000500	005237	000020		E0:
4403		000504	023727	000020	000023	FOR R0 := 0 TO 140 BY 4
4404		000512	003770			CLR R0
4405		000514				B1:
4406	80	000514	005000			DEC A(R0)
4407		000514				END ;OF FOR R0
4408	81	000516	005360	000000		ADD 4,R0
4409	82	000522				CMP R0, 140
4410		000522	062700	000004		
4411		000526	020027	000140		
4412						

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 104

4414		000532	003771						
4415		000534							
4416	83	000534							
4417		000534	012737	000133	000020				
4418		000542							
4419	84	000542	063737	000000	000002				
4420	85	000550							
4421		000550	162737	000002	000020				
4422		000556	023727	000020	000003				
4423		000564	002366						
4424		000566							
4425	86								
4426	87	000566							
4427		000566							
4428	88	000566							
4429		000566	005000						
4430		000570							
4431	89	000570	116037	000000	000002				
4432	90	000576							
4433		000576	005737	000002					
4434		000602	002415						
4435	91	000604							
4436		000604	005200						
4437		000606	020027	000167					
4438		000612	003766						
4439		000614							
4440	92	000614							
4441		000614	012700	000400					

```

BLE B1
E1:  FOR I := 133 DOWNT0 3 BY 2
      MOV 133,I
B2:  ADD A,B
      END ;OF FOR I
      SUB 2,I
      CMP I,3
      BGE B2
E2:  ;BEGIN EXAMPLES
      BEGIN ALPHA
B3:  FOR R1 := 0 TO 167
      CLR R0
B4:  MOV B A(R0),B
      IF B LT 0 THEN LEAVE ALPHA
      TST B
      BLT E3
      END ;OF FOR R0
      INC R0
      CMP R0, 167
      BLE B4
E4:  FOR R0 := 400 TO 567
      MOV 400,R0

```

.MAIN. MACRO M1111 01-APR-79 16:10 PAGE 3-3

4442		000620							
4443	93	000620							
4444		000620	005737	000002					
4445		000624	002004						
4446	94	000626							
4447		000626	005200						
4448		000630	020027	000567					
4449		000634	003771						
4450		000636							
4451	95	000636							
4452		000636							
4453	96								
4454	97	000636							
4455		000636	000207						
4456	98	000640							
4457		000640	000261						
4458		000642	000207						
4459	99	000644							
4460		000644	000241						
4461		000646	000207						

```

B5:  IF B GE 0 THEN LEAVE ALPHA
      TST B
      BGE E3
      END ;OF FOR R0
      INC R0
      CMP R0, 567
      BLE B5
E5:  END ALPHA
E3:  ;$RETURN EXAMPLES
      $RETURN
      RTS PC
      $RETURN ERROR
      SEC
      RTS PC
      $RETURN NOERROR
      CLC
      RTS PC

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 105

```

4468      100
4469      101 000650 013700 000000
4470      102 000654
4471          000654 010046
4472          000656 006316
4473          000660 004737 000700
4474      103 000664 000000
4475      104 000666 000002
4476      105 000670 000004
4477      106 000672 000006
4478      107 000674 000010
4479      108 000676 000012
4480      109 000700
4481          000700
4482          000700 062616
4483          000702 013646
4484          000704 004736
4485      110
4486      111          000001

```

```

;CASE EXAMPLES
MOV     A,R0
CASE R0
MOV R0,-(SP)
ASL @SP
JSR PC,L24
A
B
C
D
E
F
END ;OF CASE R0
L24:
ADD (SP)+,@SP
MOV @SP+,-(SP)
JSR PC,@SP+
.END

```

## 7.5 MEMORY MANAGEMENT MAPPING

## 7.5.1 MEMORY MANAGEMENT MAPPING FOR THE 11/44 -

PAR	SUPERVISOR	KERNEL	USER
---	-----	-----	----
0	PROGRAM	PROGRAM	DST BK/FST MEM
1	PROGRAM	PROGRAM	SRC BK/FST MEM
2	PROGRAM	PROGRAM	SRC BK/FST MEM
3	TEST AREA	PROGRAM	SRC BK/FST MEM
4	TEST AREA	PROGRAM	DST BK/FST MEM
5	TEST AREA	PROGRAM	DST BK/FST MEM
6	TEST AREA	MAP TO CSR'S	DST BK/FST MEM
7	PERIF PAGE	PERIF PAGE	DST BK/FST MEM

## 7.5.2 MEMORY MANAGEMENT MAPPING FOR UNIBUS-11'S WITH SUPERVISOR MODE (EG 11/45) -

PAR	SUPERVISOR	KERNEL	USER
---	-----	-----	----
0	PROGRAM	PROGRAM	DST BK
1	PROGRAM	PROGRAM	SRC BK
2	PROGRAM	PROGRAM	SRC BK
3	TEST AREA	PROGRAM	SRC BK
4	TES AREA	PROGRAM	DST BK
5	TES AREA	PROGRAM	DST BK
6	TEST AREA	MAP TO CSR'S	DST BK
7	PERIF PAGE	PERIF PAGE	DST BK

4519  
4520  
4521  
4522  
4523  
4524  
4525  
4526  
4527  
4528  
4529  
4530  
4531

## 7.5.3 MEMORY MANAGEMENT MAPPING FOR UNIBUS-11'S W/O SUPERVISOR MODE (EG 11/34) -

PAR	KERNEL	USER
---	-----	----
0	PROGRAM	PROGRAM/DST BK
1	PROGRAM	PROGRAM/SRC BK
2	PROGRAM	PROGRAM/SRC BK
3	PROGRAM	TEST AREA/SRC BK
4	PROGRAM	TEST AREA/DST BK
5	PROGRAM	TEST AREA/DST BK
6	MAP TO CSR'S	TEST AREA/DST BK
7	PERIF PAGE	PERIF PAGE/DST BK~

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 108

```

4535
4536 000000
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554      163000
4555      000001
4556 000000

      .LIST      TOC
      .ENABL     ABS
      .ENABL     AMA
      .DSABL     GBL
      :NOTE:     CZMSDC.SML IS THE SUPER.MAC SOURCE AND IS RELEASED WITH
      :THIS PROGRAM. ALL THESE .MCALL STATEMENTS REFERENCE THAT FILE.
      .MCALL     SMACIT,..PUSH,..POP,..TAG,..BRAN,..EMIT,..EMITN,..EMITL,..EMITR
      .MCALL     .IFOPR,..IS,..GENBR,..OPADD,..OPSUB,CLEAR,SET,CLEARB,SETB
      .MCALL     RNE,REQ,RLT,RGE,RGT,RLE,RPL,RMI,RHI,RLOS,RHIS,RLO,RCS,RCC
      .MCALL     IF,..OR,..IFARI,..LEAVE,..GOTO,OR,AND,THEN,ELSE,WHILE,CASE
      .MCALL     FOR,TO,DOWNT0,REPEAT,UNTIL,THRU,END,BEGIN
      .MCALL     $$END,LEAVE,JUMPT0,GOTO,PUSH,POP,LET
      .MCALL     .SIMPLE,..ARITH,ORB,ANDB,IFB,UNTILB,WHILEB,ON.ERROR,ON.NOERROR
      .MCALL     $CALL,$RETURN

      .NLIST     TTM
      .LIST      MC,SYM
      .NLIST     MD,CND,ME
      LST$$=     0
      $SWR=      163000
      $TN=       1
      SMACIT

      ; I WANT FAT PAPER!
      ; LIST MACRO CALLS, SYMBOL TABLE
      ; DON'T LIST MACRO DEFS & CONDITIONALS & EXPANSIONS
      ; DEFINED TO LIST SUPERMAC EXPANSIONS
      ; USE THESE SYSMAC SWITCHES
      ; FIRST TEST NUMBER TO ONE(1)

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 110  
 DEFINE TRAPS

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

104401  
104402  
104403  
  
104405  
  
104407  
104410  
  
104411  
104412  
104413  
104414  
  
104415  
104416  
  
104417  
  
104420  
104421  
104422  
  
104423  
104424

```

      .SBTTL DEFINE TRAPS
;ALL ENTRIES HERE MUST HAVE A CORRESPONDING ENTRY IN THE
;TRAP TABLE '$TRPAD' (NEAR END OF PROGRAM).
;*TRAP DEFINITIONS
;
;HERE IS HOW TRAPS WORK IN THIS PROGRAM
;
;ALL TRAPS EXECUTE A 'TRAP' INSTRUCTION WHICH TAKES THE PROGRAM
;TO SYMBOLIC LOCATION '$TRAP'
;
;AT $TRAP THE PROGRAM PICKS UP THE RIGHT BYTE OF THE TRAP INSTRUCTION
;AND INDEXES INTO A TABLE AT LOCATION '$TRPAD' WHICH SENDS THE PROGRAM TO
;THE SPECIFIC ROUTINE TO HANDLE THAT SPECIFIC TRAPS TASK.
;
;THE ULTIMATE DESTINATION OF A TRAP INSTRUCTION CAN BE GUESSED AT AS FOLLOWS
;
;EXAMPLE:      NOP
                NOP
                NOP
                KERNEL          ;ENTER KERNEL MODE
                NOP
;
;ADD A DOLLAR SIGN TO THE SYMBOLIC NAME AND CHECK THE CRF FOR SOMETHING CLOSE
;IN THIS CASE THE CRF HAS $KERNEL LISTED AS 032546
;AT LOCATION 32546 YOU FIND THE ROUTINE $KERNEL
;
;NOTE THAT CRF SYMBOLS ARE TRUNCATED TO 6 CHARACTERS
;SYMBOLIC NAMES GREATER THAN 6 CHARACTERS ARE USED SO I CAN
;REMEMBER WHAT THEY MEAN!
;
;ALL GOOD TRAP ROUTINES RETURN VIA AN 'RTI' INSTRUCTION
TYPEIT= 104401      ;;TTY TYPEOUT ROUTINE
TYPOC= 104402      ;;TYPE OCTAL NUMBER (WITH LEADING ZEROS)
TYPOI= 104403      ;;TYPE OCTAL NUMBER (NO LEADING ZEROS)
TYPON= 104404      ;;TYPE OCTAL NUMBER (AS PER LAST CALL)
TYPDS= 104405      ;;TYPE DECIMAL NUMBER (WITH SIGN)
TYPBN= 104406      ;;TYPE BINARY (ASCII) NUMBER
;
GTSWR= 104407      ;;GET SOFT-SWR SETTING
CKSWR= 104410      ;;TEST FOR CHANGE IN SOFT-SWR
;
RDCHR= 104411      ;;TTY TYPEIN CHARACTER ROUTINE
RDLIN= 104412      ;;TTY TYPEIN STRING ROUTINE
RDOCT= 104413      ;;READ AN OCTAL NUMBER FROM TTY
RDDEC= 104414      ;;READ A DECIMAL NUMBER FROM TTY
;
SAVREG= 104415      ;;SAVE R0-R5 ROUTINE
RESREG= 104416      ;;RESTORE R0-R5 ROUTINE
;
KERNEL= 104417      ;ENTER KERNEL MODE
;
ENERGIZE=104420     ;TURN ON MEMORY MANAGEMENT & TRAPS
DEENERGIZE=104421  ;TURN OFF MEMORY MANAGEMENT & TRAPS
KMAP= 104422       ;MAP KERNEL 1 TO 1
;
CACHON= 104423      ;TURN ON CACHE
CACHOFF=104424     ;TURN OFF CACHE

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 110-1

DEFINE TRAPS

4616			
4617	104425	LOADCSR=104425	;LOAD CORRECT CSR
4618	104426	READCSR=104426	;READ CORRECT CSR
4619			
4620	104427	PERR01= 104427	;PROGRAM DETECTED ERROR
4621	104430	PERR02= 104430	;PROGRAM DETECTED ERROR
4622	104431	PERR03= 104431	;PROGRAM DETECTED ERROR
4623	104432	PERR04= 104432	;PROGRAM DETECTED ERROR
4624	104433	PERR07= 104433	;PROGRAM DETECTED ERROR
4625	104434	PERR10= 104434	;PROGRAM DETECTED ERROR
4626	104435	PERR11= 104435	;PROGRAM DETECTED ERROR
4627	104436	PERR12= 104436	;PROGRAM DETECTED ERROR
4628	104437	PERR13= 104437	;PROGRAM DETECTED ERROR
4629	104440	PERR14= 104440	;PROGRAM DETECTED ERROR
4630	104441	PERR15= 104441	;PROGRAM DETECTED ERROR
4631	104442	PERR16= 104442	;PROGRAM DETECTED ERROR
4632	104443	PERR17= 104443	;PROGRAM DETECTED ERROR
4633	104444	PERR20= 104444	;PROGRAM DETECTED ERROR
4634	104445	PERR21= 104445	;PROGRAM DETECTED ERROR
4635	104446	PERR22= 104446	;PROGRAM DETECTED ERROR
4636	104447	PERR23= 104447	;PROGRAM DETECTED ERROR
4637	104450	PERR24= 104450	;PROGRAM DETECTED ERROR
4638	104451	PERR25= 104451	;PROGRAM DETECTED ERROR
4639	104452	PERR26= 104452	;PROGRAM DETECTED ERROR
4640	104453	PERR27= 104453	;PROGRAM DETECTED ERROR
4641	104454	PERR30= 104454	;PROGRAM DETECTED ERROR
4642	104455	PERR31= 104455	;PROGRAM DETECTED ERROR
4643	104456	PERR32= 104456	;PROGRAM DETECTED ERROR
4644	104457	PERR33= 104457	;PROGRAM DETECTED ERROR
4645	104460	PEPR34= 104460	;PROGRAM DETECTED ERROR
4646	104461	PERR35= 104461	;PROGRAM DETECTED ERROR
4647	104462	PERR36= 104462	;PROGRAM DETECTED ERROR
4648	104463	PERR37= 104463	;PROGRAM DETECTED ERROR
4649	104464	PERR40= 104464	;PROGRAM DETECTED ERROR
4650	104465	PERR41= 104465	;PROGRAM DETECTED ERROR
4651	104466	PERR42= 104466	;PROGRAM DETECTED ERROR
4652	104467	PERR43= 104467	;PROGRAM DETECTED ERROR
4653			
4654	104470	ECCDIS= 104470	;DISABLE ECC ON ALL CSR'S
4655	104471	ECC1DIS=104471	;DISABLE ECC ON 1 SELECTED CSR
4656	104472	ECCINIT=104472	;INITIALIZE ALL ECC CSR'S
4657	104473	ECC1INIT=104473	;INITIALIZE 1 SELECTED ECC CSR
4658	104474	9CSR= 104474	;WRITE GENERATED CHECKBITS IN ALL CSR'S
4659	104475	CB1CSR= 104475	;WRITE GENERATED CHECKBITS IN 1 SELECTED CSR
4660	104476	WASSBE= 104476	;WAS THERE A SBE ON ANY CSR?
4661	104477	WAS1SBE=104477	;WAS THERE A SBE ON 1 SELECTED CSR?
4662	104500	WASDBE= 104500	;WAS THERE A DBE ON ANY CSR?
4663	104501	WAS1DBE=104501	;WAS THERE A DBE ON 1 SELECTED CSR?
4664	104502	CLRCSR= 104502	;CLEAR ALL CSR'S
4665	104503	CLR1CSR=104503	;CLEAR 1 SELECTED CSR
4666	104504	CHKDIS= 104504	;DISABLE ECC & WRITE CHECKBITS FROM ALL CSR'S
4667	104505	CHK1DIS=104505	;DISABLE ECC & WRITE CHECKBITS FROM 1 SELECTED CSR
4668	104506	ENASBE= 104506	;ENABLE TRAPS ON SBE'S FROM ALL CSR'S
4669	104507	ENA1SBE=104507	;ENABLE TRAPS ON SBE'S FROM 1 SELECTED CSR
4670	104510	TSTREAD=104510	;TEST LOC (R1) & TST FOR SBE (WITHOUT FETCHES)
4671	104511	INVALID=104511	;INVALIDATE BACKGROUND PATTERN ON 'BANK'
4672	104512	ERRGEN =104512	;CHECK ERROR ADDRESS



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 110-2  
DEFINE TRAPS

4673

104513

CBREG =104513

;ENABLES CHECK/SYNDROME BIT REGISTER

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 111  
 DEFINE BASIC PDP11 STUFF

```

4675                      .SBTTL DEFINE BASIC PDP11 STUFF
4676
4677      ;*INITIAL ADDRESS OF THE STACK POINTER
4678      002000      STACK= 2000      ;;FIRST ADDRESS OF THE STACK
4679      002000      KERSTK= STACK    ;;KERNEL STACK
4680      000740      SUPSTK= 740     ;;SUPERVISOR STACK
4681      000700      USESTK= 700     ;;USER STACK
4682      104000      ERROR=EMT       ;;BASIC DEFINITION OF ERROR CALL
4683      000004      SCOPE=IOT       ;;BASIC DEFINITION OF SCOPE CALL
4684      177776      PSW= 177776     ;;PROCESSOR STATUS WORD
4685      ;STKLMT=177774              ;;STACK LIMIT REGISTER
4686      ;PIRQ= 177772              ;;PROGRAM INTERRUPT REQUEST REGISTER
4687      177570      DSWR= 177570    ;;HARDWARE SWITCH REGISTER
4688      177570      DDISP= 177570   ;;HARDWARE DISPLAY REGISTER
4689      177546      LKS= 177546     ;;LINE CLOCK (KW11-L) STATUS REGISTER
4690
4691      ;*MISCELLANEOUS DEFINITIONS
4692      000011      HT= 11           ;;CODE FOR HORIZONTAL TAB
4693      000012      LF= 12           ;;CODE LINE FEED
4694      000015      CR= 15           ;;CODE CARRIAGE RETURN
4695      000200      CRLF= 200        ;;CODE FOR CARRIAGE RETURN-LINE FEED
4696      000007      MFPT= 7          ;;CODE FOR PROCESSOR TYPE INSTRUCTION
4697
4698      ;*GENERAL PURPOSE REGISTER DEFINITIONS
4699      ;SP=R6          ;;STACK POINTER
4700      ;KSP=SP        ;;KERNEL STACK POINTER
4701      000006      SSP=SP          ;;SUPERVISOR STACK POINTER
4702      000006      USP=SP          ;;USER STACK POINTER
4703      ;PC=R7          ;;PROGRAM COUNTER
4704
4705      ;*'SWITCH REGISTER' SWITCH DEFINITIONS
4706      100000      SW15= 100000
4707      040000      SW14= 40000
4708      020000      SW13= 20000
4709      010000      SW12= 10000
4710      004000      SW11= 4000
4711      002000      SW10= 2000
4712      001000      SW9= 1000
4713      000400      SW8= 400
4714      000200      SW7= 200
4715      000100      SW6= 100
4716      000040      SW5= 40
4717      000020      SW4= 20
4718      000010      SW3= 10
4719      000004      SW2= 4
4720      000002      SW1= 2
4721      000001      SW0= 1
4722
4723      ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
4724      100000      BIT15= 100000
4725      040000      BIT14= 40000
4726      020000      BIT13= 20000
4727      010000      BIT12= 10000
4728      004000      BIT11= 4000
4729      002000      BIT10= 2000
4730      001000      BIT9= 1000
4731      000400      BIT8= 400
  
```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 111-1  
 DEFINE BASIC PDP11 STUFF

```

4732      000200      BIT7= 200
4733      000100      BIT6= 100
4734      000040      BIT5= 40
4735      000020      BIT4= 20
4736      000010      BIT3= 10
4737      000004      BIT2= 4
4738      000002      BIT1= 2
4739      000001      BIT0= 1
4740
4741      *BASIC 'CPU' TRAP VECTOR ADDRESSES
4742      000004      ERRVEC= 4      ;;TIME OUT AND OTHER ERRORS
4743      000010      RESVEC= 10     ;;RESERVED AND ILLEGAL INSTRUCTIONS
4744      ;TBITVEC=14      ;;'T' BIT
4745      ;TRTVEC=      14      ;;TRACE TRAP
4746      ;BPTVEC=      14      ;;BREAKPOINT TRAP (BPT)
4747      000020      IOTVEC= 20      ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
4748      000024      PWRVEC= 24      ;;POWER FAIL
4749      000030      EMTVEC= 30      ;;EMULATOR TRAP (EMT) **ERROR**
4750      000034      TRAPVEC=34      ;;'TRAP' TRAP
4751      000060      TKVEC= 60      ;;TTY KEYBOARD VECTOR
4752      ;TPVEC= 64      ;;TTY PRINTER VECTOR
4753      ;LKVEC= 100      ;;LINE CLOCK (KW11-L) VECTOR
4754      000114      CACHVEC=114     ;;CACHE ERROR INTERRUPT VECTOR
4755      000114      PARVEC=CACHVEC
4756      ;PIRQVEC=240      ;;PROGRAM INTERRUPT REQUEST VECTOR
4757      000250      MMVEC= 250      ;;MEMORY MANAGEMENT VECTOR
4758      ;SBTTL DEFINE CACHE REGISTERS
4759      ;MEMERR = 177744      ;;CACHE ERROR REGISTER
4760      177746      CONTRL = 177746 ;;MEMORY CONTROL REGISTER
4761      177750      MAINT = 177750  ;;MEMORY MAINTENANCE REGISTER
4762      ;HITMIS = 177752      ;;HIT MISS REGISTER '1' IMPLIES HIT IN CACHE
4763      177754      DATARG = 177754 ;;DATA REGISTER
4764
4765      ;SBTTL DEFINE CPU REGISTERS
4766      177766      CPUERR = 177766 ;;CPU ERROR REGISTER HOLDS CONDITION THAT CAUSED
4767
4768      ;SBTTL DEFINE MEMORY MANAGEMENT REGISTERS
4769      *MEMORY MANAGEMENT STATUS REGISTER ADDRESSES
4770      177572      MMRG= 177572
4771      177574      MMR1= 177574
4772      177576      MMR2= 177576
4773      172516      MMR3= 172516
4774
4775      *USER 'I' PAGE DESCRIPTOR REGISTERS
4776      177600      UIPDR0= 177600
4777      ;UIPDR1=      177602
4778      ;UIPDR2=      177604
4779      ;UIPDR3=      177606
4780      ;UIPDR4=      177610
4781      ;UIPDR5=      177612
4782      ;UIPDR6=      177614
4783      ;UIPDR7=      177616
4784
4785      *USER 'D' PAGE DESCRIPTOR REGISTORS
4786      ;UDPDR0=      177620
4787      ;UDPDR1=      177622
4788      ;UDPDR2=      177624

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 111-2  
 DEFINE MEMORY MANAGEMENT REGISTERS

```

4789      ;UDPDR3=      177626
4790      ;UDPDR4=      177630
4791      ;UDPDR5=      177632
4792      ;UDPDR6=      177634
4793      ;UDPDR7=      177636
4794
4795      ;*USER 'I' PAGE ADDRESS REGISTERS
4796      FASTCITY=UIPAR0
4797      177640      UIPAR0= 177640      ;PATTERN PROGRAM SPACE
4798      177642      UIPAR1= 177642      ;PATTERN PROGRAM SPACE
4799      177644      UIPAR2= 177644      ;PATTERN PROGRAM SPACE
4800      177646      UIPAR3= 177646      ;PATTERN PROGRAM SPACE
4801      177650      UIPAR4= 177650      ;PATTERN PROGRAM SPACE
4802      177652      UIPAR5= 177652      ;PATTERN PROGRAM SPACE
4803      177654      UIPAR6= 177654      ;PATTERN PROGRAM SPACE
4804      ;UIPAR7=      177656      ;PATTERN PROGRAM SPACE
4805
4806      ;*USER 'D' PAGE ADDRESS REGISTERS
4807      177660      UDPAR0= 177660      ;PATTERN PROGRAM SPACE
4808      ;UDPAR1=      177662      ;PATTERN PROGRAM SPACE
4809      ;UDPAR2=      177664      ;PATTERN PROGRAM SPACE
4810      ;UDPAR3=      177666      ;PATTERN PROGRAM SPACE
4811      ;UDPAR4=      177670      ;PATTERN PROGRAM SPACE
4812      ;UDPAR5=      177672      ;PATTERN PROGRAM SPACE
4813      ;UDPAR6=      177674      ;PATTERN PROGRAM SPACE
4814      177676      UDPAR7= 177676      ;PATTERN PROGRAM SPACE
4815
4816      ;*SUPERVISOR 'I' PAGE DESCRIPTOR REGISTERS
4817      172200      SIPDR0= 172200
4818      ;SIPDR1=      172202
4819      ;SIPDR2=      172204
4820      ;SIPDR3=      172206
4821      ;SIPDR4=      172210
4822      ;SIPDR5=      172212
4823      ;SIPDR6=      172214
4824      ;SIPDR7=      172216
4825
4826      ;*SUPERVISOR 'D' PAGE DESCRIPTOR REGISTERS
4827      ;SDPDR0=      172220
4828      ;SDPDR1=      172222
4829      ;SDPDR2=      172224
4830      ;SDPDR3=      172226
4831      ;SDPDR4=      172230
4832      ;SDPDR5=      172232
4833      ;SDPDR6=      172234
4834      ;SDPDR7=      172236
4835
4836      ;*SUPERVISOR 'I' PAGE ADDRESS REGISTERS
4837      172240      SIPAR0= 172240
4838      ;SIPAR1=      172242
4839      ;SIPAR2=      172244
4840      172246      SIPAR3= 172246      ;TEST AREA
4841      ;SIPAR4=      172250      ;TEST AREA
4842      172252      SIPAR5= 172252      ;TEST AREA
4843      172254      SIPAR6= 172254      ;TEST AREA
4844      ;SIPAR7=      172256
4845

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 111-3

DEFINE MEMORY MANAGEMENT REGISTERS

```

4846                                     ;*SUPERVISOR 'D' PAGE ADDRESS REGISTERS
4847         172260      SDPAR0= 172260
4848                                     ;SDPAR1=      172262
4849                                     ;SDPAR2=      172264
4850                                     ;SDPAR3=      172266
4851                                     ;SDPAR4=      172270
4852         172272      SDPAR5= 172272
4853         172274      SDPAR6= 172274
4854         172276      SDPAR7= 172276
4855
4856                                     ;*KERNEL 'I' PAGE DESCRIPTOR REGISTERS
4857         172300      KIPDR0= 172300
4858                                     ;KIPDR1=      172302
4859                                     ;KIPDR2=      172304
4860                                     ;KIPDR3=      172306
4861                                     ;KIPDR4=      172310
4862                                     ;KIPDR5=      172312
4863                                     ;KIPDR6=      172314
4864                                     ;KIPDR7=      172316
4865
4866                                     ;*KERNEL 'D' PAGE DESCRIPTOR REGISTER;
4867                                     ;KDPDR0=      172320
4868                                     ;KDPDR1=      172322
4869                                     ;KDPDR2=      172324
4870                                     ;KDPDR3=      172326
4871                                     ;KDPDR4=      172330
4872                                     ;KDPDR5=      172332
4873                                     ;KDPDR6=      172334
4874                                     ;KDPDR7=      172336
4875
4876                                     ;*KERNEL 'I' PAGE ADDRESS REGISTERS
4877         172340      KIPAR0= 172340
4878                                     ;KIPAR1=      172342
4879                                     ;KIPAR2=      172344
4880                                     ;KIPAR3=      172346
4881         172350      KIPAR4= 172350
4882         172352      KIPAR5= 172352
4883         172354      KIPAR6= 172354
4884                                     ;KIPAR7=      172356
4885
4886                                     ;*KERNEL 'D' PAGE ADDRESS REGISTERS
4887         172360      KDPAR0= 172360
4888                                     ;KDPAR1=      172362
4889                                     ;KDPAR2=      172364
4890                                     ;KDPAR3=      172366
4891                                     ;KDPAR4=      172370
4892                                     ;KDPAR5=      172372
4893         172374      KDPAR6= 172374
4894         172376      KDPAR7= 172376
4895

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 113  
 DEFINE UNIBUS MAP REGISTERS

```

4898                                     .SBTTL DEFINE UNIBUS MAP REGISTERS
4899                                     :*THE LOWER 16 BITS OF THE MAP REGISTERS ARE LABELED 'MAPLXX'
4900                                     :*THE UPPER 6 BITS OF THE MAP REGISTERS ARE LABELED 'MAPHXX'
4901         170200      MAPL0 = 170200
4902         170202      MAPH0 = 170202
4903         170204      MAPL1 = 170204
4904      :MAPH1 = 170206
4905      :MAPL2 = 170210
4906      :MAPH2 = 170212
      :MAPL3 = 170214
      :MAPH3 = 170216
4908      :MAPL4 = 170220
4909      :MAPH4 = 170222
      :MAPL5 = 170224
      :MAPH5 = 170226
4911      :MAPL6 = 170230
4912      :MAPH6 = 170232
4913      :MAPL7 = 170234
4914      :MAPH7 = 170236
4915      :MAPL10 = 170240
4916      :MAPH10 = 170242
4917      :MAPL11 = 170244
4918      :MAPH11 = 170246
4919      :MAPL12 = 170250
4920      :MAPH12 = 170252
4921      :MAPL13 = 170254
4922      :MAPH13 = 170256
4923      :MAPL14 = 170260
4924      :MAPH14 = 170262
4925      :MAPL15 = 170264
4926      :MAPH15 = 170266
4927      :MAPL16 = 170270
4928      :MAPH16 = 170272
4929      :MAPL17 = 170274
4930      :MAPH17 = 170276
4931      :MAPL20 = 170300
4932      :MAPH20 = 170302
4933      :MAPL21 = 170304
4934      :MAPH21 = 170306
4935      :MAPL22 = 170310
4936      :MAPH22 = 170312
4937      :MAPL23 = 170314
4938      :MAPH23 = 170316
4939      :MAPL24 = 170320
4940      :MAPH24 = 170320
4941      :MAPL25 = 170324
4942      :MAPH25 = 170326
4943      :MAPL26 = 170330
4944      :MAPH26 = 170332
4945      :MAPL27 = 170334
4946      :MAPH27 = 170336
4947      :MAPL30 = 170340
4948      :MAPH30 = 170342
4949      :MAPL31 = 170344
4950      :MAPH31 = 170346
4951      :MAPL32 = 170350
4952      :MAPH32 = 170352
4953
4954

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 113-1  
 DEFINE UNIBUS MAP REGISTERS

4955		:MAPL33 = 170354
4956		:MAPH33 = 170356
4957		:MAPL34 = 170360
4958		:MAPH34 = 170362
4959		:MAPL35 = 170364
4960		:MAPH35 = 170366
4961		:MAPL36 = 170370
4962		:MAPH36 = 170372
4963		:MAPL37 = 170374
4964		:MAPH37 = 170376

4965			.SBTTL	DEFINE	SOFTWARE SWITCH & DISPLAY REGISTERS
------	--	--	--------	--------	-------------------------------------

4966					
4967	000174		DISPREG=	174	
4968	000176		SWREG=	176	

4969			.SBTTL	DEFINE	CONTROL STATUS REGISTERS
4970					
4971	172100		CSRADD=	172100	

4972			.SBTTL	DEFINE	PARAMETERS
4973					
4974	060000		FIRST=	60000	:START OF THE 16K TEST PATTERN AREA
4975	157776		LAST=	157776	:END OF THE 16K TEST PATTERN AREA
4976	040000		SIZE=	40000	:SIZE OF THE 16K TEST PATTERN AREA (FOR SOB INSTRUCTIONS)

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 115  
 DEFINE PARAMETERS

```

4979          .LIST MD                      ;BE NICE TO SEE MY DEFINITIONS
4980          .SBTTL MACRO FATAL
4981          :***** FATAL *****
4982          :FATAL IS USED TO REPORT FATAL ERRORS (ERRORS THAT PREVENT
4983          :THE PROGRAM FROM CONTINUING).
4984          :*****
4985          :*****MACRO***MACRO***MACRO***
4986          .MACRO FATAL ARG
4987          .NLIST
4988          .DSABL CRF
4989          .IIF DF LST$$ .LIST ME
4990          .ENABL CRF
4991          .LIST
4992          INC FATAL$                      ;SET FATAL INDICATOR
4993          ERROR +ARG
4994          .DSABL CRF
4995          .IIF DF LST$$ .NLIST ME
4996          .ENABL CRF
4997          .ENDM FATAL
4998
4999          .SBTTL MACRO TYPE
5000          .MACRO TYPE ARG
5001          .NLIST
5002          .DSABL CRF
5003          .IIF DF LST$$ .LIST ME
5004          .ENABL CRF
5005          .LIST
5006          .IF B ARG
5007          TYPEIT
5008          .IFF
5009          TYPEIT ,ARG
5010          .ENDC
5011          .DSABL CRF
5012          .IIF DF LST$$ .NLIST ME
5013          .ENABL CRF
5014          .ENDM TYPE
5015

```



```

5018 .SBTTL MACRO NEWTST
5019 :*****~** NEWTST *****~**
5020 :NEWTST IS USED AS THE FIRST INSTRUCTION OF A TEST.
5021 :IF WILL:
5022 :1) GENERATE A TEST NUMBER FOR THE LABEL OF THIS TEST
5023 :2) PUT STARS BEFORE AND AFTER A MESSAGE
5024 :ARGUMENTS
5025 :1) ASCII -- THIS IS THE MESSAGE THAT WILL APPEAR
5026 :ON THE LISTING
5027 :2) ICOUNT -- IF NON-BLANK AND BIT 11 OF $SWR = 1 IT WILL BE
5028 :THE NUMBER OF ITERATIONS TO MAKE ON THIS TEST
5029 :3) RETURN -- IF NON-BLANK WILL BE THE ADDRESS TO
5030 :WHICH THE NEXT SCOPE STATEMENT WILL
5031 :LOOP BACK TO.
5032 :4) COMAND -- IF NON-BLANK WILL BE THE FIRST
5033 :INSTRUCTION OF THE TEST
5034 :IF BLANK SCOPE WILL BE THE
5035 :FIRST INSTRUCTION
5036 :*****
5037 .MACRO NEWTST ASCII,ICOUNT,RETURN,COMAND
5038 $STN=1
5039 $NWTST=0
5040 .NLIST MC
5041 .IF B <COMAND>
5042 $$NEWTST \STN,<ASCII>,SCOPE
5043 .IFF
5044 $$NEWTST \STN,<ASCII>,<COMAND>
5045 .ENDC
5046 .NLIST
5047 .LIST ME
5048 .LIST
5049 .IF NE 4000&$SWR
5050 .IF NB ICOUNT
5051 .IF LE <ICOUNT-1>
5052 MOV #1,$TIMES ;;DO 1 ITERATION
5053 .IFF
5054 MOV #ICOUNT,$TIMES ;;DO ICOUNT ITERATIONS
5055 .ENDC
5056 .ENDC
5057 .IF NB RETURN
5058 MOV #RETURN,$LPADR ;;SET SCOPE LOOP ADDRESS
5059 .ENDC
5060 .ENDC
5061 .NLIST
5062 .LIST MC
5063 .LIST
5064 .NLIST ME
5065 .ENDM NEWTST

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 119  
 MACRO \$\$NEWTEST

```

5068      .SBTTL MACRO  $$NEWTEST
5069      .MACRO  $$NEWTEST      A,ASC,COMND
5070      .IRP    ASCI,<ASC>
5071      .IF EQ $NWTST
5072      $NWTST=1
5073      .SBTTL  T'A'      ASCI
5074      .NLIST
5075      .LIST   ME
5076      .LIST
5077      ;*****
5078      ;*TEST A      ASCI
5079      .IFF
5080      ASCI
5081      .ENDC
5082      .ENDM
5083      ;*****
5084      TST'A: COMND
5085      .NLIST ME
5086      $TN=$TN+1
5087      .ENDM  $$NEWTEST
5088
5089      .SBTTL MACRO  SUBTST
5090      ;***** SUBTST *****
5091      ;
5092      ;THIS MACRO WILL FORMAT A SUBTEST HEADING WITH STARS
5093      ;A .SBTTL WILL BE FORCED & .NLISTED FOR THE TABLE OF CONTENTS.
5094      ;
5095      ;ARGUMENT:
5096      ;1) TXT --      THIS IS THE MESSAGE THAT WILL APPEAR IN THE TABLE OF CONTENTS & LISTING.
5097      ;
5098      ;EXAMPLE:      SUBTST <<THIS IS A FUN SUBTST>>
5099      ;
5100      ;*****
5101
5102      .MACRO  SUBTST  ASCII
5103      .NLIST  MC
5104      $SUBTST <ASCII>
5105      .LIST  MC
5106      .ENDM  SUBTST
5107
5108      .SBTTL MACRO  $SUBTST
5109      .MACRO  $SUBTST ASC
5110      .IRP    ASCI,<ASC>
5111      .SBTTL  ASCI
5112      .NLIST
5113      .LIST   ME
5114      .LIST
5115      ;*****
5116      ;*SUBTEST      ASCI
5117      .ENDM
5118      ;*****
5119      .NLIST ME
5120      .ENDM  $SUBTST

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 121  
 MACRO TYPOCT

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

```

.SBTTL MACRO TYPOCT
***** TYPOCT *****
:
:TYPOCT IS USED TO CHANGE A BINARY NUMBER
:  TO A 6 DIGIT OCTAL NUMBER AND TYPE IT
:
:ARGUMENTS:
:1)  NUM      THE NUMBER TO BE TYPED
:2)  REMARK   ALLOWS A COMMENT TO BE MADE
:
:ROUTINES REQUIRED
:1)  CONVERT BINARY TO OCTAL AND TYPE (.$TYPOCT)
:2)  TYPE AN ASCII STRING (.$TYPE)
:
:EXAMPLES:
:1)  TYPOCT HILMT,<TYPES THE CONTENTS OF HILMT>
:2)  TYPOCT #5,<TYPES ' 000005'>
:
*****
.MACRO TYPOCT NUM,REMARK
.NLIST
.DSABL CRF
.IIF DF LST$$ .LIST ME
.ENABL CRF
.LIST
MOV     NUM, -(SP)      ;;SAVE NUM FOR TYPEOUT
.IIF NB <REMARK>,      ;;REMARK
TYPOC   NUM              ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
.DSABL CRF
.IIF DF LST$$ .NLIST ME
.ENABL CRF
.ENDM TYPOCT

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 123

MACRO TYPOCS

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

```
.SBTTL MACRO TYPOCS
***** TYPOCS *****
:TYPOCS IS USED TO CHANGE A BINARY NUMBER TO AN OCTAL
:NUMBER AND TYPE 1 TO 6 DIGITS
:WITH OR WITHOUT LEADING ZEROS.
:
:ARGUMENTS:
:1) NUM NUMBER TO BE TYPED
:2) REMARK ALLOWS A COMMENT TO BE MADE
:3) N NUMBER OF DIGITS (1 TO 6) TO BE TYPED
:4) Z BLANK=SUPPRESS LEADING ZEROS (TYPES SPACES)
:NON-BLANK=TYPE LEADING ZEROS
:
:ROUTINES REQUIRED
:1) CONVERT BINARY TO OCTAL AND TYPE (.$TYPOCT)
:2) TYPE AN ASCIZ STRING (.$TYPE)
:
:EXAMPLES:
:1) TYPOCS #12345,<TYPES '5'>,1
:2) TYPOCS #004,<TYPES '04'>,2,X
:3) TYPOCS #004,<TYPES '4'>,2
:*****
:
:MACRO TYPOCS NUM,REMARK,N,Z
:NLIST
:DSABL CRF
:IF DF LST$$ .LIST ME
:ENABL CRF
:.LIST
:MOV NUM,-(SP) ;;SAVE NUM FOR TYPEOUT
:IF NB <REMARK>, ;;REMARK
:TYPOS ;;GO TYPE--OCTAL ASCII
:IF NB N
:.BYTE N ;;TYPE N DIGIT(S)
:.IFF
:.BYTE 6 ;;TYPE 6 DIGITS
:.ENDC
:IF NB Z
:.BYTE 1 ;;TYPE LEADING ZEROS
:.IFF
:.BYTE 0 ;;SUPPRESS LEADING ZEROS
:.ENDC
:DSABL CRF
:IF DF LST$$ .NLIST ME
:ENABL CRF
:.ENDM TYPOCS
```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 125  
 MACRO TYPDEC

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

```

.SBTTL MACRO TYPDEC
***** TYPDEC *****
TYPDEC IS USE TO CHANGE A BINARY NUMBER TO A SIGNED
DECIMAL NUMBER AND TYPE IT REPLACING LEADING ZERO
WITH SPACES.
NOTE: IF THE NUMBER IS NEGATIVE A
MINUS SIGN WILL BE TYPED.

ARGUMENTS:
1) NUM NUMBER TO BE TYPED
2) REMARK ALLOWS A COMMENT TO BE MADE

ROUTINES REQUIRED
1) CONVERT BINARY TO DECIMAL AND TYPE (.$TYPDEC)
2) TYPE AN ASCII STRING (.$TYPE)

EXAMPLES
1) TYPDEC SIZE,<TYPE THE CONTENTS OF SIZE>
2) TYPDEC #-10.,<TYPE A MINUS TEN>
*****

.MACRO TYPDEC NUM,REMARK
.NLIST
.DSABL CRF
.IIF DF LST$$ .LIST ME
.ENABL CRF
.LIST
MOV NUM,-(SP) ;;SAVE NUM FOR TYPEOUT
.IIF NB <REMARK>, ;;REMARK
TYPDS ;;GO TYPE--DECIMAL ASCII WITH SIGN
.DSABL CRF
.IIF DF LST$$ .NLIST ME
.ENABL CRF
.ENDM TYPDEC
  
```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 126

MACRO BMOV

```

5261      .SBTTL MACRO BMOV
5262      :***** BMOV *****
5263      :
5264      : THIS MACRO MOVES A BLOCK OF DATA.
5265      :
5266      : ARGUMENTS:
5267      :
5268      : 1) FROMHERE      THE FIRST ADDRESS OF THE SOURCE BLOCK.
5269      :
5270      : 2) TOHERE        THE FIRST ADDRESS OF THE DESTINATION BLOCK.
5271      :                   IF BLANK THE 1ST ADDRESS OF THE USER INSTRUCTION
5272      :                   PAR'S IS USED (FASTCITY).
5273      :
5274      : 3) SIZE           THE SIZE OF THE SOURCE BLOCK.
5275      :                   IF BLANK A 16 WORD TRANSFER IS ASSUMED.
5276      :                   'WHY DEFAULT TO 16 WORDS?' YOU ASK!
5277      :                   'BECAUSE THAT'S HOW MANY WORDS TO THE USER PAR
5278      :                   REGISTERS & THAT'S WHERE I INTEND TO MOVE LOTS
5279      :                   OF STUFF.' I REPLY!
5280      :
5281      :*****
5282
5283      .MACRO BMOV FROMHERE,TOHERE,SIZE
5284      .IF B TOHERE
5285      .NLIST
5286      .DSABL CRF
5287      .IIF DF LST$$ .LIST ME
5288      .ENABL CRF
5289      .LIST
5290      JSR R5,BLOCK1
5291      FROMHERE
5292      .DSABL CRF
5293      .IIF DF LST$$ .NLIST ME
5294      .ENABL CRF
5295      .MEXIT
5296      .ENDC
5297      .IF B SIZE
5298      .NLIST
5299      .DSABL CRF
5300      .IIF DF LST$$ .LIST ME
5301      .ENABL CRF
5302      .LIST
5303      JSR R5,BLOCK2
5304      TOHERE
5305      FROMHERE
5306      .DSABL CRF
5307      .IIF DF LST$$ .NLIST ME
5308      .ENABL CRF
5309      .MEXIT
5310      .IFF
5311      .NLIST
5312      .DSABL CRF
5313      .IIF DF LST$$ .LIST ME
5314      .ENABL CRF
5315      .LIST
5316      JSR R5,BLOCK3
5317      SIZE

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 126-1  
MACRO BMOV

5318  
5319  
5320  
5321  
5322  
5323  
5324

TOHERE  
FROMHERE  
.DSABL CRF  
.IIF DF LST\$\$ .NLIST ME  
.ENABL CRF  
.ENDC  
.ENDM BMOV

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 128  
 MACRO MAP

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

```

.SBTTL MACRO MAP
***** MAP *****
: THIS MACRO MAPS A MEMORY BANK (16K) INTO THE
: TEST PATTERN AREA (SUPERVISOR VIRTUAL (60000-157777)).
: ARGUMENTS:
: 1) BANK THE BANK OF 16K WORDS TO BE MAPPED.
: THERE ARE 120 BANKS OF 16K WORDS
: EXAMPLES
: MAP LOC ;LOCATION 'LOC' CONTAINS THE # OF THE BANK TO MAP
: MAP #28. ;BANK 34 (OCTAL) WILL BE MAPPED
*****
.MACRO MAP BANK
PUSH R3
.NLIST
.DSABL CRF
.IIF DF LST$$ .LIST ME
.ENABL CRF
.LIST
.IF B BANK
MOV #120.,R3
.IFF
MOV BANK,R3
.ENDC
CALL MAPPER
.DSABL CRF
.IIF DF LST$$ .NLIST ME
.ENABL CRF
POP R3
.ENDM MAP

```



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 130  
 MACRO SUPERVISOR

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

```
.SBTTL MACRO SUPERVISOR
***** SUPERVISOR *****
:
: THIS MACRO SWITCHES TO SUPERVISOR MODE.
:
: ARGUMENTS: NONE.
:
: *****
```

```
.MACRO SUPERVISOR
.NLIST
.DSABL CRF
.IIF DF LST$$ .LIST ME
.ENABL CRF
.LIST
BIS #BIT14,PSW ;GO TO SUPERVISOR MODE
.DSABL CRF
.IIF DF LST$$ .NLIST ME
.ENABL CRF
.ENDM SUPERVISOR
```

```
.SBTTL MACRO USER
***** USER *****
:
: THIS MACRO SWITCHES TO USER MODE.
:
: ARGUMENTS: NONE.
:
: *****
```

```
.MACRO USER
.NLIST
.DSABL CRF
.IIF DF LST$$ .LIST ME
.ENABL CRF
.LIST
BIS #BIT15!BIT14,PSW ;GO TO USER MODE
.DSABL CRF
.IIF DF LST$$ .NLIST ME
.ENABL CRF
.ENDM USER
```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 131  
MACRO TESTAREA

5409  
5410  
5411  
5412  
5413  
5414  
5415  
5416  
5417  
5418  
5419  
5420  
5421  
5422  
5423  
5424  
5425  
5426  
5427  
5428

```
.SBTTL MACRO TESTAREA
***** TESTAREA *****
: THIS MACRO SWITCHES TO THE SPECIFIED TEST MODE.
: ARGUMENTS: NONE.
*****
```

```
.MACRO TESTAREA
.NLIST
.DSABL CRF
.IIF DF LST$$ .LIST ME
.ENABL CRF
.LIST
BIS TESTMODE,PSW
.DSABL CRF
.IIF DF LST$$ .NLIST ME
.ENABL CRF
.ENDM TESTAREA
```

;GO TO SYSTEM TEST MODE

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 133

MACRO SET4 &amp; RES4

```

5431      .SBTTL MACRO SET4 & RES4
5432      :***** SET4 & RES4 *****
5433      :
5434      : THESE MACROS SET & RESTORE VECTOR 4(TIMEOUT TRAP)
5435      :
5436      : IN IT'S RESTORED MODE TRAPS ARE REPORTED AS SUCH.
5437      :
5438      : ARGUMENTS:  LOC      ;THE LOCATION TO VECTOR TO (ONLY USED IN 'SET4' NOT 'RES4')
5439      :
5440      : I USE THE SET4 AND RES4 MACROS AROUND CODE THAT I EXPECT TO TRAP TO 4
5441      : LIKE LOOKING FOR ALL POSSIBLE CSR'S AND ETC.  WHENEVER CODE IS NOT
5442      : SURROUNDED BY SET4 AND RES4 THEN ANY TRAPS TO 4 WILL CAUSE AN ERROR
5443      : PRINTOUT THAT SAYS 'UNEXPECTED TRAP TO 4' AND ALL THE ASSOCIATED REGISTER JUNK
5444      :*****
5445
5446      .MACRO SET4 ARG
5447      .NLIST
5448      .DSABL CRF
5449      .IIF DF LST$$ .LIST ME
5450      .ENABL CRF
5451      .LIST
5452      MOV ARG,4
5453      .DSABL CRF
5454      .IIF DF LST$$ .NLIST ME
5455      .ENABL CRF
5456      .ENDM SET4
5457
5458      .MACRO RES4
5459      .NLIST
5460      .DSABL CRF
5461      .IIF DF LST$$ .LIST ME
5462      .ENABL CRF
5463      .LIST
5464      MOV #TIMEOUT,4
5465      CMP #1,PROTYP      ;IS THIS AN 11/44?
5466      BNE 101$           ;BRANCH IF NOT
5467      CLR CPUERR         ;CLEAR OUT THE CPU ERROR REGISTER BITS
5468
5469      101$:              ;THAT A EXPECTED TRAP COULD HAVE SET
5470      .DSABL CRF
5471      .IIF DF LST$$ .NLIST ME
5472      .ENABL CRF
5473      .ENDM RES4

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 135  
 MACRO DLEFT

5476  
 5477  
 5478  
 5479  
 5480  
 5481  
 5482  
 5483  
 5484  
 5485  
 5486  
 5487  
 5488  
 5489  
 5490  
 5491  
 5492  
 5493  
 5494  
 5495  
 5496  
 5497

```

.SBTTL MACRO DLEFT
:***** DLEFT *****
: THIS MACRO DOES A DOUBLE WORD LEFT SHIFT
: ARGUMENTS:  LOC      ;THE LOCATION TO BE SHIFTED LEFT (CARRY TO LOC+2)
:*****
      .MACRO DLEFT ARG
      .NLIST
      .DSABL CRF
      .IIF DF LST$$ .LIST ME
      .ENABL CRF
      .LIST
      ROL ARG
      ROL ARG+2
      .DSABL CRF
      .IIF DF LST$$ .NLIST ME
      .ENABL CRF
      .ENDM DLEFT
      .NLIST MD
;DON'T NEED TO SEE THEM ANY MORE

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 137  
TRAP CATCHER

```

5500          .SBTTL  TRAP CATCHER
5501          .=0
5502 000000 000000 000000      .WORD  0,0
5503          000177          .REPT  177          ;.WORD  .+2,HALT
5507
5508          .SBTTL  ACT11 HOOKS
5509          ;*THE HOOKS REQUIRED BY ACT11 ARE DEFINED AND SETUP BELOW:
5510          ;*
5511          ;*
5512          ;*
5513          ;*
5514          ;*
5515          ;*
5516          ;*
5517          ;*
5518          ;*
5519          ;*
5520          ;*
5521          ;*
5522          ;*
5523          ;*
5524 000046 000046      .=46
5525          015232      $ENDAD
5526 000052 000052      .=52
5527          000020      .WORD  BIT4
5528          000024      .SBTTL  APT11 HOOKS
5529 000024 000200      .=24      ;;SET POWER FAIL TO POINT TO START OF PROGRAM
5530          000042      200      ;;FOR APT START UP
5531 000042 002000      .=42
5532          000044      STACK      ;SO RT11 CAN START WITH RUN COMMAND
5533 000044 065740      .=44      ;;POINT TO APT INDIRECT ADDRESS PNTR.
5534          000200      $APTHDR ;;POINT TO APT HEADER BLOCK
5535 000200 000437      .=200
5536 000202 000442      START3: BR      START1      ;'NORMAL' START
5537          000300      BR      START2      ;RESTART (SAVE ERROR ACCOUNTING)
5538 000300 005037 002612      .=300
5539 000304 000137 003654      START1: CLR      RESTART
5540 000310          JMP      START
5541 000316 000137 003654      START2: SET      RESTART
5542          002000      JMP      START
          .=STACK

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 139  
 VARIABLES INITIALIZED TO ZERO

5545			.SBTTL	VARIABLES	INITIALIZED TO ZERO
5546			;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS		
5547			;*USED IN THE PROGRAM.		
5548	002000		\$CMTAG:		::START OF COMMON TAGS
5549	002000	000000	SELONLY:0		::SELECT ONLY BANKS MARKED BY FIELD SERVICE MODE FLAG
5550	002002	000000	DIAGFLAG:0		::SET FOR SHIFTING DIAGONAL TEST
5551	002004	000000	KAMIKAZE:0		::SET FOR KAMIKAZE MODE TESTING
5552	002006	000000	SKIPKAMI:0		::USED TO SKIP RESTORING KAMIKAZE MODE WHEN MODIFIED
5553			:NEXT TWO BYTES ARE DISPLAYED IN THE DISPLAY REGISTER		
5554	002010	000	\$PATMAR:.BYTE	0	::PATTERN NUMBER
5555	002011	000	\$BANK:.BYTE	0	::BANK & SIGN
5556	002012	000	\$ERFLG:.BYTE	0	::CONTAINS ERROR FLAG
5557	002013	000	\$ITEMB:.BYTE	0	::CONTAINS ITEM CONTROL BYTE
5558	002014	000000	LASTERROR:.WORD	0	::NUMBER OF ERRORS ON LAST PASS
5559	002016	000000	ERRPC:.WORD	0	::CONTAINS PC OF ERROR FOR TYPEOUT
5560	002020	000000	BADPC:.WORD	0	::CONTAINS PC OF ERROR
5561	002022	000000	ERRSP:.WORD	0	::CONTAINS SP OF ERROR FOR TYPEOUT
5562	002024	000000	BADSP:.WORD	0	::CONTAINS SP OF ERROR
5563	002026	000000	ERRPSW:.WORD	0	::CONTAINS PSW OF ERROR FOR TYPEOUT
5564	002030	000000	BADPSW:.WORD	0	::CONTAINS PSW OF ERROR
5565	002032	000000	ADDRESS:.WORD	0	::CONTAINS ADDRESS OF 'BAD' DATA
5566	002034	000000	PADDRESS:.WORD	0	::ADDRESS OF PARITY ERROR
5567	002036	000000 000000	PHYADD:.WORD	0,0	::22 BIT PHYSICAL ADDRESS
5568	002042	000000	GOOD:.WORD	0	::CONTAINS 'GOOD' DATA
5569	002044	000000	GOOD2:.WORD	0	::CONTAINS 'GOOD2' DATA
5570	002046	000000	GOOD3:.WORD	0	::CONTAINS 'GOOD3' DATA
5571	002050	000000	BAD:.WORD	0	::CONTAINS 'BAD' DATA
5572	002052	000000	BAD2:.WORD	0	::CONTAINS 'BAD2' DATA
5573	002054	000000	BAD3:.WORD	0	::CONTAINS 'BAD3' DATA
5574	002056	000000	BADXOR:.WORD	0	::XOR OF GOOD & BAD = BAD BITS!
5575	002060	000000	\$AUTO:.WORD	0	::AUTOMATIC MODE INDICATOR FOR APT,ACT, & XXDP
5576	002062	000000	FATAL\$: .WORD	0	::FATAL ERROR INDICATOR
5577	002064	000000	SKPERR:.WORD	0	::USED TO SKIP ERROR MESSAGE IN '\$ERRGEN'
5578	002066	000000	NEMCNT: 0		::NON-EXISTANT MEMORY COUNTER (HOLES)
5579	002070	000000	PARCNT: 0		::PARITY ERROR COUNTER
5580	002072	000000	PATERR: 0		::PATTERN ERROR COUNTER
5581	002074	000000	NOPAR: 0		::NO PARITY ERROR MODE INDICATOR
5582	002076	000000	NONEM: 0		::NO NON-EXISTANT MEMORY (HOLES) MODE INDICATOR
5583	002100	000000	BANK: 0		::MEMORY BANK UNDER TEST
5584	002102	000000	BANKINDEX:0		::USED TO INDEX INTO CONFIG TABLE
5585	002104	000000	CPUBIT: 0		::CONTAINS 1 BIT TO IDENTIFY CPU TO CONFIGURATION TABLE
5586	002106	000000	MUT: 0		::MEMORY UNDER TEST FLAG
5587	002110	000000	PATTERN:0		::PATTERN NUMBER UNDER TEST
5588	002112	000000	KPFLAG:.WORD	0	::BANK IS PROTECTED REGION OF ECC
5589	002114	000000	ACFLAG:.WORD	0	::BANK CAN BE ACCESSED BY THIS CPU
5590	002116	000000	MKFLAG:.WORD	0	::IF SET INDICATES MS11-M OR MF11S-K UNDER TEST
5591	002120	000000	PFLAG:.WORD	0	::BANK IS IN PROGRAM SPACE
5592	002122	000000	RRFLAG:.WORD	0	::BANK IS WHERE PROGRAM RELOCATION IS REQUIRED TO TEST
5593	002124	000000	RLFLAG:.WORD	0	::PROGRAM IS RELOCATED FLAG
5594	002126	000000	BMFLAG:.WORD	0	::'BANK IS IDENTIFIED AS BAD MEMORY' FLAG
5595	002130	000000	EUFLAG:.WORD	0	::'BANK HAS EUB MEMORY' FLAG
5596	002132	000000	TMFLAG:.WORD	0	::'TYPE OF MEMORY TO TEST' FLAG; 0 = PARITY, 1 = ECC
5597	002134	000000	INTFLAG:.WORD	0	::'BANK IS INTERLEAVED' FLAG
5598	002136	000000	INT64K:.WORD	0	::'BANK IS 64K INTERLEAVED' FLAG
5599	002140	000000	PMEMFLG:.WORD	0	::'MEMORY UNDER TEST IS A MS11-P' FLAG
5600	002142	000000	ABORTFLAG:.WORD	0	::'ABORT OCCURED' FLAG
5601	002144	000000	CTLKVEC:.WORD	0	::HOLDS OLD KERNAL STACK POINTER IN CASE OF CNTL/K

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 139-1

VARIABLES INITIALIZED TO ZERO

5602	002146	000000	CSR:	.WORD	0		;DATA TO OR FROM CSR
5603	002150	000000	CSRNO:	0			;CSR ADDRESS NUMBER (4 LSB'S)
5604	002152	000000	SAVCSR:	.WORD	0		;LOCATION TO SAVE CSRNO DURING FS COMMAND
5605	002154	000000	OLDCSR:	.WORD	0		;OLD CSR NUMBER(USED IN INH PTR TEST)
5606							;THESE LOCATIONS STORE GPR'S DURING SUPERVISOR TESTS
5607	002156	000000	SUPDR0:	0			
5608	002160	000000	SUPDR1:	0			
5609	002162	000000	SUPDR2:	0			
5610	002164	000000	SUPDR3:	0			
5611	002166	000000	SUPDR4:	0			
5612	002170	000000	SUPDR5:	0			
5613	002172	000000	SUPDR6:	0			
5614	002174	000000	DUMMY:	0			;DUMMY LOCATION FOR ADDRESS PASSING
5615							;THESE LOCATIONS STORE GPR'S & PSW DURING DETAILED ERROR PRINTOUTS
5616	002176	000000	DETR0:	0			
5617	002200	000000	DETR1:	0			
5618	002202	000000	DETR2:	0			
5619	002204	000000	DETR3:	0			
5620	002206	000000	DETR4:	0			
5621	002210	000000	DETR5:	0			
5622	002212	000000	DETR6:	0			
5623	002214	000000	DETR7:	0			
5624	002216	000000	DETR8:	0			
5625	002220	000000	DETR9:	0			
5626	002222	000000	DETR10:	0			
5627			DETR11:	0			
5628	002224	000000	DETR12:	0			
5629	002226	000000	DETR13:	0			
5630	002230	000000	DETR14:	0			
5631	002232	000000	DETR15:	0			
5632	002234	000000	DETR16:	0			
5633	002236	000000	DETR17:	0			
5634	002240	000000	DETR18:	0			
5635	002244	000000	DETR19:	0			
5636	002250	000000	DETR20:	0			
5637	002254	000000	DETR21:	0			
5638	002260	000000	DETR22:	0			
5639	002262	000	DETR23:	0			
5640	002264	000	DETR24:	0			
5641	002266	000000	DETR25:	0			
5642	002270	000000	DETR26:	0			
5643	002272	000000	DETR27:	0			
5644	002274	000000	DETR28:	0			
5645	002276	000000	DETR29:	0			
5646	002280	000000	DETR30:	0			
5647	002282	000000	DETR31:	0			
5648	002284	000000	DETR32:	0			
5649	002286	000000	DETR33:	0			
5650	002288	000000	DETR34:	0			
5651	002290	000000	DETR35:	0			
5652	002292	000000	DETR36:	0			
5653	002294	000000	DETR37:	0			
5654	002296	000000	DETR38:	0			
5655	002298	000000	DETR39:	0			
5656	002300	000000	DETR40:	0			
5657	002302	000000	DETR41:	0			
5658	002304	000000	DETR42:	0			

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 139-2  
 VARIABLES INITIALIZED TO ZERO

5659	002330	000000		SUCCESS: .WORD	0	: FLAG SET BY SUCCESSFUL TASK OR SUBROUTINE
5660	002332	000000		ZEROS: .WORD	0	: FOR AID IN 'MOV' INSTRUCTIONS
5661	002334	000000		TIME: .WORD	0	: SECONDS THAT BATTERIES SHOULD LAST
5662	002336	000000		SKIPMK: .WORD	0	: FLAG TO SKIP MKCONTROL SUBROUTINE
5663	002340	000000		NULLFLAG: .WORD	0	: SET WHEN RUNNING NULL PATTERNS
5664	002342	000000		QVFLAG: 0		: FLAGS QUICK VERIFY PASS UNDER APT, ACT, OR XXDP CHAIN MODE
5665	002344	000000		ACTFLAG: 0		: FLAGS ACT AUTOMATIC MODE PROGRAMMING RULES
5666	002346	000000		APTFLAG: 0		: FLAG APT AUTOMATIC MODE PROGRAMMING RULES
5667	002350	000000		XXDPCHAIN: 0		: FLAGS XXDP CHAIN MODE PROGRAMMING RULES
5668				: NOTE: THESE TWO BYTES MUST STAY TOGETHER		
5669	002352	000		\$NULL: .BYTE	0	: CONTAINS NULL CHARACTER FOR FILLS
5670	002353	000		\$FILLS: .BYTE	0	: CONTAINS # OF FILL CHARACTERS
5671	002354	000		\$TPFLG: .BYTE	0	: 'TERMINAL NOT AVAILABLE' FLAG
5672				.EVEN		
5673	002356	000000		\$ESCAPE: 0		: ESCAPE ON ERROR ADDRESS
5674	002360	000000		EVEN: 0		: USED FOR ALTERNATE DATA PATTERNS
5675	002362	000000		STRIPES: 0		: COUNTS DIAGONAL STRIPES
5676	002364	000000		COUNT: 0		: BACKED UP COPY OF STRIPES
5677	002366	000000		NOTAB: 0		: NO TABLE BEING PRINTED - NOW
5678	002370	000000		BSIZE: 0		: SIZE OF 11/45 MOS MEMORY IN K WORDS
5679	002372	000000		KSIZE: 0		: SIZE OF MF11S-K MEMORY IN K WORDS
5680	002374	000000		LSIZE: 0		: SIZE OF MS11-L MEMORY IN K WORDS
5681	002376	000000		MSIZE: 0		: SIZE OF MS11-M MEMORY IN K WORDS
5682	002400	000000		PSIZE: 0		: SIZE OF UNIBUS PARITY MEMORY IN K WORDS
5683	002402	000000		TOOMANY: 0		: FLAGS WHEN TOO MANY ERRORS HAVE BEEN PRINTED FOR A BANK
5684	002404	000000		READONLY: 0		: FLAG TO PATTERNS TO READ ONLY
5685	002406	000000	000000	TESTADD: 0,0		: THE ADDRESS TO RUN CSR TESTS ON
5686	002412	000000		UNITOP: 0		: HIGHEST ACCESSABLE BANK OF MEMORY THRU UNIBUS MAP
5687	002414	000000		STOPOK: 0		: FLAG TO ALLOW STOPPING WITH SWITCH REGISTER
5688	002416	000000		APTPAR: .WORD	0	: AMOUNT OF PARITY MEMORY ACCORDING TO APT
5689	002420	000000		APTECC: .WORD	0	: AMOUNT OF ECC MEMORY ACCORDING TO APT
5690	002422	000000		NOFSMODE: 0		: FLAG TO DISABLE FIELD SERVICE MODE
5691	002424	000000		NOERROR: 0		: 'THIS IS NOT AN ERROR' FLAG
5692	002426	000000		LOADBANK: 0		: BANK LOADERS ARE RELOCATED TO
5693	002430	000000		TEMP: 0		: USED FOR JUNK
5694	002432	000000		QUICK: 0		: QUICK STOP FLAG FOR APT POWER FAIL
5695	002434	000000		NOSCOPE: 0		: 'NO SCOPE LOOP ALLOWED' FLAG
5696	002436	000000		FSINFLAG: 0		: 'FIELD SERVICE - NO INTERNAL INTERLEAVE' FLAG
5697	002440	000000		APTSIZE: 0		: APT SIZING INFO FLAG
5698	002442	000000		FS7FLAG: 0		: TRUE WHEN IN FIELD SERVICE COMMAND 7
5699	002444	000000		CONFERROR: 0		: CONFIGURATION ERROR FLAG
5700	002446	000000		I: 0		: USED FOR GENERAL PURPOSE INDEXING
5701	002450	000000		NO22BIT: 0		: NO 22-BIT MODE FLAG
5702	002452	000000		NOSUPER: 0		: NO SUPERVISOR MODE FLAG
5703	002454	000000		ERRADD: .WORD	0	: HOLDS THE CSR'S ERROR ADDRESS
5704	002456	000000	000000 000000	CSRINFO: 0,0,0,0,0,0,0,0		: USED TO STORE INFORMATION ABOUT THE 16
	002464	000000	000000 000000			
	002472	000000	000000			
5705	002476	000000	000000 000000		0,0,0,0,0,0,0,0	: POSSIBLE CSR'S
	002504	000000	000000 000000			
	002512	000000	000000			
5706	002516	000000		LINK1: 0		: USED TO HOLD LINKS TO PATTERNS WHICH
5707	002520	000000		LINK2: 0		: CAN EXECUTE IN THE PAR/PDR'S OR NOT
5708	002522	000000		CSRHOLD: 0		: USED TO STORE CSR VALUES FOR CSR TESTS
5709	002524	000000		KFLAG: 0		: USED TO FLAG MF11S-K MEMORY TO TESTS
5710	002526	000000	000000	PGMCSR: .WORD	0,0	: POINTS TO PROGRAM CSR
5711	002532	000000		INHECC: .WORD	0	: FLAGS INHIBIT ECC TESTS ON RELOCATION



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 139-3  
VARIABLES INITIALIZED TO ZERO

5712 002534 000000  
5713 002536 000000  
5714 002540

INHBANK:.WORD 0  
FULLREL:.WORD 0  
\$CMTGE: ;\*END OF COMMON TAGS



```

5760          .SBTTL  CONFIGURATION TABLE
5761          :CONFIG:FIRST 16K CONFIGURATION WORDS (2 EACH)
5762          2ND      16K CONFIGURATION WORDS (2 EACH)
5763          :
5764          200TH    16K CONFIGURATION WORDS (2 EACH)
5765          :
5766          :CONFIGURATION WORDS:
5767          LOW:      BIT 0      ERRORS PRESENT
5768                   BIT 1      MEMORY SUCCESSFULLY ACCESSED
5769                   BIT 2-4     RESERVED
5770                   BIT 5      SKIP ECC LOGIC TESTS FLAG (1=SKIP)
5771                   BIT 6      PROTECTED REGION OF ECC MEMORY
5772                   BIT 7      PROTECTED (PROGRAM SPACE)
5773                   BIT 8-11    CSR CODE
5774                   BIT 12-15   INTERLEAVED CSR CODE
5775          HIGH:     BIT 0-7     NUMBER OF ERRORS
5776                   BIT 8-10    MEMORY TYPE
5777                   BIT 11      INTERLEAVED BOARD TYPE (0=128K, 1=64K)
5778                   BIT 12      INTERLEAVE ENABLED
5779                   BIT 13      'BACKGROUND PATTERN VALID' FLAG
5780                   BIT 14      BANK SELECTED FOR TEST BY FIELD SERVICE MODE
5781                   BIT 15      LOADERS HOME BANK
5782 002650 000201  :CONFIG: .REPT 201
5785 003654  :CONFIEND:

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 144

\*\*\*\*\* MAIN \*\*\*\*\*

5787  
5788 003654

```

.SBTTL ***** MAIN *****
START: SUBTST <<INITIALIZE VARIABLES TO ZERO>>
:*****
:*SUBTEST INITIALIZE VARIABLES TO ZERO
:*****
TSTB $ENV
BNE NORES
RESET
NORES: CLEAR MONFLG ;;CLEAR RETURN TO MONITOR FLAG
MOV SP,SAVMON ;;SAVE XXDP MONITOR RESTART ADDRESS
MOV KSTACK,SP ;;SETUP THE STACK POINTER
MOV #SCMTAG,R0 ;;FIRST LOCATION TO BE CLEARED
1$: CLR (R0)+ ;;CLEAR MEMORY LOCATION
CMP #SCMTGE,R0 ;;DONE?
BNE 1$ ;;LOOP BACK IF NO
MOV #167, LASTBANK ;;RESTORE L'S/BANK (THIS MUST BE DONE PRIOR TO SYSTEM SIZING)
SUBTST <<CLEAR NON-PROGRAM SPACE>>

```

```

5789 003654 105737 065660
5790 003660 001001
5791 003662 000005
5792 003664
5793 003670 010637 002270
5794 003674 013706 002560
5795 003700 012700 002000
5796 003704 005020
5797 003706 022700 002540
5798 0 3712 001374
5799 003714 012737 000167 002552
5800 003722

```

```

:*****
:*SUBTEST CLEAR NON-PROGRAM SPACE
:*****

```

```

5801
5802
5803
5804 003722 012737 000001 002074
5805 003730 005000
5806 003732 000241
5807 003734 005520
5808 003736 020027 160000
5809 003742 103773
5810 003744 005037 002074
5811

```

```

;THIS ATTEMPS TO GET RID OF ANY PARITY ERRORS BY WRITING INTO
;EVERY LOCATION THAT IS NOT LOADED INTO BY THE PROGRAM OR ALLOCATED
;TO THE XXDP LOADERS
MOV #1,NOPAR ;;PARITY ACTION = COUNT & IGNORE
CLR R0
2$: CLC
ADC (R0)+
CMP R0,#160000
BLO 2$
CLR NOPAR ;;RESTORE DEFAULT PARITY ACTION

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-32 09:41 PAGE 145  
CLEAR NON-PROGRAM SPACE

5813 003750

SUBTST &lt;&lt;TYPE OF SYSTEM SIZER&gt;&gt;

\*\*\*\*\*  
:SUBTEST TYPE OF SYSTEM SIZER  
\*\*\*\*\*

```

5814 003750 000401          BR      SYSSIZ          ;SKIP OVER VARIABLE LOCATION
5815 003752 000000          PROTYP: .WORD 0
5816 003754          SYSSIZ: SET4 #4$
5817 003762 005737 177746          TST  CONTRL          ;SEE IF CACHE REGISTER RESPONDS
5818 003766          SET4 #9$          ;YES - DO WE HAVE 11/44 TYPE CACHE
5819 003774 005737 177750          TST  MAINT          ;OR 11/60 TYPE CACHE?
5820 004000 000411          BR      5$          ;BRANCH IF 11/44 TYPE CACHE
5821 004002 012737 000014 002544 9$: MOV #14,CACHKF          ;TURN OFF CONSTANT FOR 11/60 CACHE
5822 004010 000405          BR      5$
5823 004012 005037 002540          CLR  CACHKN          ;NO CACHE ON SYSTEM
5824 004016 012737 002332 067330 4$: MOV #ZEROS,DT14          ;DO NOT PRINT CONTRL ERROR MESSAGES
5825 004024          SET4 #6$
5826 004032 005737 172516          TST  MMR3          ;DO WE HAVE AN MMR3?
5827 004036 005037 172516          CLR  MMR3          ;YES WE DO
5828 004042 052737 000020 172516 BIS #BIT4,MMR3          ;SEE IF THERE IS 22-BIT MODE
5829 004050 032737 000020 172516 BIT #BIT4,MMR3
5830 004056 001026          BNE  10$          ;BRANCH IF 22-BIT RELOCATION
5831 004060 000413          BR      7$          ;BRANCH IF .MMR3 BUT NO 22-BIT RELOC.
5832          ;* 11/34 TYPE MACHINES ENTER HERE
5833 004062 012737 140000 002546 6$: MOV #140000,TESTMODE          ;MAKE TESTMODE USER
5834 004070 005237 002452          INC  NOSUPER
5835 004074 005037 067200          CLR  DT5+10
5836 004100 005037 067340          CLR  DT14+10
5837 004104 005237 002450          INC  NO22BIT
5838          ;* 11/45 TYPE MACHINES ENTER HERE
5839 004110 005237 002450          7$: INC  NO22BIT
5840 004114 012737 000007 002552 MOV #7, LASTBANK
5841 004122 005037 067202          CLR  DT5+12
5842 004126 005037 067342          CLR  DT14+12
5843 004132 000417          BR      8$
5844 004134          10$: SET4 #8$
5845 004142 000007          MFPT
5846          ;TYPE OF PROCESSOR TEST: THIS INSTRUCTION
5847          ; (AVAILABLE ON NEWER PROCESSORS ONLY) PLACES
5848          ; A CODE IN THE LOWER BYTE OF R0 THAT
5849          ; INDICATES THE PROCESSOR TYPE. 1=11/44
5850          ; 3=11/24
5850 004144 110037 003752          MOVB R0,PROTYP          ;MOV THE CODE TO PROTYP
5851 004150 022737 000003 003752 CMP #3,PROTYP          ;IS THIS AN 11/24?
5852 004156 001005          BNE  8$          ;BRANCH IF NOT - WE HAVE AN 11/44
5853 004160 005237 002452          INC  NOSUPER          ;NO SUPERVISOR MODE
5854 004164 012737 140000 002546 MOV #140000,TESTMODE          ;MAKE TEST MODE USER
5855          ;
5856 004172          8$: SET4 #11$          ;TRAPS GO TO 11$          ;R-C
5857 004200 005037 061322          CLR  CPERRF          ;CLEAR THE FLAG          ;R-C
5858 004204 005737 177766          TST  @#177766          ;IS THERE A CPU ERROR REGISTER? ;R-C
5859 004210 012737 177777 061322 MOV #-1,CPERRF          ;YES-TRAPPED
5860 004216          11$: RES4          ;R-C

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 147  
TYPE OF SYSTEM SIZER

5863 004240

```

SUBTST <<INITIALIZE VARIABLES TO NON ZERO>>
:*****
:*SUBTEST INITIALIZE VARIABLES TO NON ZERO
:*****

```

5864 004240  
5865 004246 012737 000003 002602  
5866 004254  
5867 004262 012737 176543 002572  
5868 004270 012737 123456 002574  
5869 004276 013737 002572 002566  
5870 004304 013737 002574 002570  
5871 004312 012737 000377 002616  
5872 004320 012737 177400 002620  
5873 004326

```

SET WORST
MOV #3,FLIPLOC
SET HEADER
MOV #176543,MSEEDH
MOV #123456,MSEEDL
MOV MSEEDH,SEEDHI ;PRIME THE RANDOM NUMBER GENERATOR
MOV MSEEDL,SEEDLO ;BOTH HIGH AND LOW WORDS
MOV #377,BAKPAT
MOV #177400,SWAPAT
SUBTST <<INITIALIZE VECTORS>>

```

```

:*****
:*SUBTEST INITIALIZE VECTORS
:*****

```

5874 004326 012737 060166 000020  
5875 004334 012737 000340 000022  
5876 004342 012737 060522 000030  
5877 004350 012737 000340 000032  
5878 004356 012737 065754 000034  
5879 004364 012737 000340 000036  
5880 004372 012737 054356 000024  
5881 004400 012737 000340 000026  
5882 004406 012737 042406 000114  
5883 004414 012737 000340 000116  
5884 004422 012737 042602 000010  
5885 004430 012737 000340 000012  
5886 004436 012737 042556 000004  
5887 004444 012737 000340 000006  
5888 004452 012737 042570 000250  
5889 004460 012737 000340 000252  
5890 004466 104423

```

MOV #$$SCOPE,IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
MOV #340,IOTVEC+2 ;;LEVEL 7
MOV #$$ERROR,EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
MOV #340,EMTVEC+2 ;;LEVEL 7
MOV #$$TRAP,TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
MOV #340,TRAPVEC+2;LEVEL 7
MOV #$$PWRDN,PWRVEC ;;POWER FAILURE VECTOR
MOV #340,PWRVEC+2 ;;LEVEL 7
MOV #$$PARITY,PARVEC;GET READY FOR PARITY ERRORS
MOV #340,PARVEC+2
MOV #PDP1105,RESVEC;RESERVED INSTRUCTION TRAP
MOV #340,RESVEC+2
MOV #$$TIMEOUT,ERRVEC;SETUP TIMEOUT ERRORS
MOV #340,ERRVEC+2 ;SET PRIORITY OF ERROR TRAPS
MOV #$$MMTRAP,MMVEC ;VECTOR FOR MEMORY MANAGEMENT
MOV #340,MMVEC+2
CACHON ;TURN CACHE ON

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 149  
INITIALIZE VECTORS

5893 004470

5894  
5895  
5896  
5897  
5898 004470 012700 065724  
5899 004474 012001  
5900 004476 012703 020204  
5901 004502 012702 000020  
5902 004506 004737 004606  
5903 004512 012001  
5904 004514 012702 000010  
5905 004520 004737 004606  
5906 004524 012001  
5907 004526 012703 020434  
5908 004532 012702 000020  
5909 004536 004737 004606  
5910 004542 012001  
5911 004544 012702 000010  
5912 004550 004737 004606  
5913 004554 012001  
5914 004556 012703 020620  
5915 004562 012702 000020  
5916 004566 004737 004606  
5917 004572 012001  
5918 004574 012702 000010  
5919 004600 004737 004606  
5920 004604 000417  
5921  
5922 004606

SUBTST <<INITIALIZE PATTERNS>>

\*\*\*\*\*  
:SUBTEST INITIALIZE PATTERNS  
\*\*\*\*\*

:THE APT E-TABLE DETERMINES WHICH PATTERNS ARE GOING TO BE RUN.  
:EACH BIT SET REPRESENTS A PATTERN TABLE ENTRY THAT IS TO BE LEFT  
:ALONE (TO BE RUN). EACH BIT CLEARED REPRESENTS A PATTERN TABLE ENTRY  
:THAT IS TO BE OVERLAYED WITH THE ADDRESS OF A NULL PATTERN.

MOV #SDDW0,R0  
MOV (R0)+,R1  
MOV #MKCSRT,R3  
MOV #16.,R2  
CALL PATPLUG  
MOV (R0)+,R1  
MOV #8.,R2  
CALL PATPLUG  
MOV (R0)+,R1  
MOV #MKPAT,R3  
MOV #16.,R2  
CALL PATPLUG  
MOV (R0)+,R1  
MOV #8.,R2  
CALL PATPLUG  
MOV (R0)+,R1  
MOV #MKJPAT,R3  
MOV #16.,R2  
CALL PATPLUG  
MOV (R0)+,R1  
MOV #8.,R2  
CALL PATPLUG  
BR SUBAAA

PATPLUG:SUBTST <<SUBR PLUG IN NULL PATTERNS>>

\*\*\*\*\*  
:SUBTEST SUBR PLUG IN NULL PATTERNS  
\*\*\*\*\*

FOR I := #1 TO R2  
ROR R1  
ON.NOERROR ;IF CARRY CLEAR  
MOV #MT0999,(R3)  
END ;OF ON.ERROR  
ADD #2,R3  
END ;OF FOR  
RETURN

5923 004606  
5924 004614 006001  
5925 004616  
5926 004620 012713 026760  
5927 004624  
5928 004624 062703 000002  
5929 004630  
5930 004642 000207

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 151  
 SUBR PLUG IN NULL PATTERNS

5933 004644

SUBAAA: SUBTST &lt;&lt;CLEAR THE CONFIGURATION TABLE&gt;&gt;

\*\*\*\*\*  
 :SUBTEST CLEAR THE CONFIGURATION TABLE  
 \*\*\*\*\*

5934

:THIS ZEROS (UNLESS WE STARTED AT ADDRESS 202) THE CONFIG TABLE

5935

:WHICH IS FULLY DISCRIBED AT LOCATION 'CONFIG'.

5936

:ENABLE LSB

5937 004644

IF RESTART IS FALSE

5938 004652

MOV #CONFIG,R0

5939 004656

CLR (R0)+

5940 004660

CMP #CONFIEND,R0

5941 004664

BNE 1\$

5942 004666

END :OF IF RESTART

5943

:DSABL LSB

5944 004666

MOV #BIT1,CPUBIT

:SET ID BIT

5945 004674

SUBTST &lt;&lt;SIZE FOR A HARDWARE SWITCH REGISTER&gt;&gt;

\*\*\*\*\*  
 :SUBTEST SIZE FOR A HARDWARE SWITCH REGISTER  
 \*\*\*\*\*

5946

::IF NOT FOUND OR IT IS

5947

::EQUAL TO A '-1', SETUP FOR A SOFTWARE SWITCH REGISTER.

5948

:ENABL LSB

5949 004674

SET4 #3\$

:TRAPS TO 4 GOTO 3\$

5950 004702

MOV #DSWR,SWR

::SETUP FOR A HARDWARE SWITCH REGISTER

5951 004710

MOV #DDISP,DISPLAY

::AND A HARDWARE DISPLAY REGISTER

5952 004716

IF #-1 EQ @SWR

:IF NO TRAP FROM REFERENCE TO @SWR AND @SWR = #-1

5953 004726

BR 2\$

::BRANCH IF NO TIMEOUT

5954 004730

MOV #2\$,(SP)

::SET UP FOR TRAP RETURN

5955 004734

RTI

5956 004736

RES4

:RESET TRAPS TO 4 TO DEFAULT

5957 004760

MOV #SWREG,SWR

::POINT TO SOFTWARE SWR

5958 004766

MOV #DISPREG,DISPLAY

5959 004774

END :OF IF #-1

5960

:DSABL LSB



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 153  
 SIZE FOR A HARDWARE SWITCH REGISTER

5963 004774

```

SUBAAB: SUBTST <<SETUP ACT, APT, & XXDP>>
*****
:SUBTEST      SETUP ACT, APT, & XXDP
*****
:THIS SETS UP A BUNCH OF FLAGS TO TELL THE PROGRAM EVERYTHING
:IT CARES TO KNOW ABOUT APT, ACT, & XXDP.
CLR          $PASS          ;CLEAR PASS COUNT
IFB #BIT5 SET.IN $ENVM
SET          STPFLG          ;INDICATE NO TERMINAL
END :OF IFB #BIT5
IFB #BIT7 SET.IN $ENVM
SET          APTSIZE
END :OF IFB #BIT7
IFB $ENV EQ #1
SET          APTFLAG,QVFLAG,$AUTO,QUICK
MOV          #APTDOWN,PWRVEC
MOV          #SSWREG,SWR      ;USE APT SWR
ELSE
IF 42 NE #STACK AND 42 NE #0
SET QVFLAG,$AUTO
IF 42 EQ #SENDAD
SET          ACTFLAG
ELSE
SET          XXDPCHAIN
END :OF IF 42
END :OF IF 42
END :OF IFB $ENV
  
```

5964

5965

5966 004774 005037 065646

5967 005000

5968 005010

5969 005016

5970 005016

5971 005026

5972 005034

5973 005034

5974 005044

5975 005074 012737 047746 000024

5976 005102 012737 065662 002622

5977 005110

5978 005112

5979 005130

5980 005144

5981 005154

5982 005162

5983 005164

5984 005172

5985 005172

5986 005172

CZMSPA0 MS1 -L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 154  
 SETUP ACT, APT, & XXDP

5988 005172

```

SUBTST <<PROTECT PROGRAM & LOADERS>>
*****
*SUBTEST    PROTECT PROGRAM & LOADERS
*****
BIS        #BIT7,CONFIG        ;PROTECT PROGRAM SPACE (BANK 0)
BIS        #BIT7,CONFIG+4      ;PROTECT LOADER SPACE (BANK 1)
IF #SENDAD NE 42                ;NOT ACT-11?
    IF NO22BIT NE #0
        SET    MONFLG          ;RETURN TO XXDP MONITOR
        ERROR +64              ;ILLEGAL PROCESSOR
    ELSE
        TYPE    MSG000          ;TYPE PROGRAM TITLE
    END
END ;OF IF #SENDAD

```

5989 005172 052737 000200 002650  
 5990 005200 052737 000200 002654  
 5991 005206  
 5992 005216  
 5993 005224  
 5994 005232 104064  
 5995 005234  
 5996 005236  
 5997 005242  
 5998 005242  
 5999  
 6000 005242

```

SUBTST <<CHECK SYSTEM FOR CACHE>>
*****
*SUBTEST    CHECK SYSTEM FOR CACHE
*****
; * THIS FIGURES OUT IF THERE IS A CACHE ON THE SYSTEM,
; * WHAT TYPE OF SYSTEM IT IS, AND WHETHER IT IS ENABLED
; * OR DISABLED.
SET4        #3$
TST         CONTRL              ;IS THERE A CONTROL REGISTER?
SET4        #2$
TST         MAINT               ;IS THERE A MAINTENANCE REGISTER?
SET4        #1$
TST         DATARG              ;IS THERE A DATA REGISTER?
TYPE        MSG117              ; 11/44
BR          4$
1$: SET      MONFLG              ; 11/34
    ERROR    +64
                                ;PROCESSOR NOT SUPPORTED BY THIS DIAGNOSTIC
                                ; 11/60
2$: SET      MONFLG
    ERROR    +64
                                ;PROCESSOR NOT SUPPORTED BY THIS DIAGNOSTIC
                                ;SET CACHE DISABLE BITS
                                ;CLEAR CACHE DISABLE BITS
                                ;IS THE BIT SET?
                                ;BRANCH IF THE BIT IS SET
                                ;IS THE BIT SET?
                                ;BRANCH IF THE BIT IS SET
                                ; CACHE BYPASSED
4$: BIS      #BIT2!BIT3,CONTRL
    BIC      #BIT2!BIT3,CONTRL
    BIT      #BIT2,CONTRL
    BNE      7$
    BIT      #BIT3,CONTRL
    BEQ      6$
7$: TYPE    MSG121
    CACHOFF
    MOV      CACHKN,CACHKN+2
    CLR      CACHKN
    BR       8$
3$: TYPE    MSG119
6$: TYPE    MSG120
                                ;SAVE INFO ABOUT CACHE
                                ;CACHE CANNOT BE USED - IT'S BYPASSED
                                ;
                                ; NO
                                ;CACHE AVAILABLE

```

6001  
 6002  
 6003  
 6004 005242  
 6005 005250 005737 177746  
 6006 005254  
 6007 005262 005737 177750  
 6008 005266  
 6009 005274 005737 177754  
 6010 005300  
 6011 005304 000410  
 6012 005306  
 6013 005314 104064  
 6014  
 6015 005316  
 6016 005324 104064  
 6017  
 6018 005326 052737 000014 177746  
 6019 005334 042737 000014 177746  
 6020 005342 032737 000004 177746  
 6021 005350 001004  
 6022 005352 032737 000010 177746  
 6023 005360 001413  
 6024 005362  
 6025 005366 104424  
 6026 005370 013737 002540 002542  
 6027 005376 005037 002540  
 6028 005402 000404  
 6029 005404  
 6030 005410

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 155  
CHECK SYSTEM FOR CACHE

6032 005414

SUBTST &lt;&lt;SETUP USER &amp; SUPERVISOR STACK&gt;&gt;

\*\*\*\*\*  
:SUBTEST SETUP USER & SUPERVISOR STACK  
\*\*\*\*\*

6033 005414 104421  
6034 005416 005737 002452  
6035 005422 001011

8\$: DEENERGIZE ;TURN OFF MEMORY MANAGEMENT  
TST NOSUPER ;IS THERE A SUPERVISOR MODE?  
BNE 5\$ ;NO-SKIP SUPERVISOR SETUP.

6036  
6037  
6038 005424 042737 030000 177776  
6039 005432 052737 010000 177776

;SET PREVIOUS MODE TO SUPERVISOR  
BIC #BIT13!BIT12,PSW  
BIS #BIT12,PSW

6040  
6041 005440  
6042 005444 006606

PUSH #SUPSTK  
MTP1 SSP

6043  
6044  
6045 005446 052737 030000 177776 5\$:

;SET PREVIOUS MODE TO USER  
BIS #BIT13!BIT12,PSW

6046  
6047 005454  
6048 005460 006606  
6049

PUSH #USESTK  
MTP1 USP

6050 005462

SUBTST &lt;&lt;GET SOFTWARE SWITCH REGISTER IF NECESSARY&gt;&gt;

\*\*\*\*\*  
:SUBTEST GET SOFTWARE SWITCH REGISTER IF NECESSARY  
\*\*\*\*\*

6051 005462  
6052 005470  
6053 005500 104407  
6054 005502  
6055 005502  
6056  
6057 005502

IF \$AUTO IS FALSE ;IF NOT (APT OR ACT)  
IF SWR EQ #SWREG ;IF SOFTWARE SWITCH REG SELECTED  
GTSWR ;GET SOFT-SWR SETTINGS  
END :OF IF SWR  
END :OF IF \$AUTO

SUBTST &lt;&lt;GET MEMORY MANAGEMENT READY&gt;&gt;

\*\*\*\*\*  
:SUBTEST GET MEMORY MANAGEMENT READY  
\*\*\*\*\*

6058 005502 104422  
6059 005504  
6060 005520 104420

KMAP ;MAP KERNEL SPACE 1 TO 1  
MAP ;MAP SUPERVISOR SPACE (TEST AREA) 1 TO 1  
ENERGIZE ;TURN ON MEMORY MANAGEMENT

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 157  
GET MEMORY MANAGEMENT READY

6063 005522

005522 000004

6064  
6065  
6066  
6067  
6068  
6069  
6070  
6071  
6072  
6073  
6074  
6075  
6076  
6077  
6078  
6079  
6080  
6081  
6082  
6083 005524 005005  
6084 005526 005000  
6085 005530 012703 172100  
6086 005534 012737 000001 002074  
6087 005542  
6088 005550  
6089 005550 005713  
6090 005552 052705 000001  
6091 005556 005004  
6092 005560 042760 000004 002456  
6093 005566 052760 000030 002456  
6094 005574 005013  
6095 005576  
6096 005602  
6097 005610 004737 005714  
6098 005614  
6099 005614 005013  
6100 005616 004737 006206  
6101 005622  
6102 005632 016037 002456 002050  
6103 005640 104021  
6104 005642  
6105 005642 062700 000002  
6106 005646 062703 000002  
6107 005652 006305  
6108 005654  
6109 005656 005204  
6110 005660  
6111 005660  
6112 005666 006005  
6113 005670 005704  
6114 005672  
6115 005674 052705 100000

# NEWTST <<BIT TEST OF ALL CSR'S>>

\*\*\*\*\*  
:TEST 1 BIT TEST OF ALL CSR'S  
\*\*\*\*\*

TST1: SCOPE

\* THE FIRST PART OF THE CONFIGURATION ANALYSIS DOES THE FOLLOWING:  
\* 1) FINDS WHICH CSR'S RESPOND, AND PUTS THEM INTO THE CSR INFORMATION  
\* TABLE, AND STORES ANOTHER BIT FOR 'TOTCSRS'.  
\* 2) TESTS THE CSR BITS COMMON TO ALL CSR'S.  
\* 3) FIGURES OUT IF THE MODULE IS A FCC OR PARITY MEMORY  
\* 4) TESTS THE BITS PARTICULAR TO THAT TYPE OF CSR.  
\* 5) IF ANY BITS TEST BAD IN THE CSR UNDER TEST, THE CSR OK BIT IN THE  
\* CSR INFORMATION TABLE IS CLEARED.

\* THE INFORMATION BITS ONE THROUGH THREE FORM A CODE WHICH GIVES THE TYPE  
\* OF CSR:

TYPE	NOT USED BIT2	ECC TYPE BIT1	ECC BIT0	CODE TOTALS
MS11-L	0	0	0	0
MS11-M	0	0	1	1
MS11-P	0	1	1	3

\* THIS MEMORY CODE WILL BE USED IN THE SECOND PART OF THIS ANALYSIS:

```

CLR R5 ;R5 IS THE TOTAL CSR NUMBER
CLR R0 ;R0 IS A TABLE INDEX
MOV #CSRADD,R3 ;R3 HAS THE CSR ADDRESS
MOV #1,NOPAR ;IGNORE PARITY ERRORS
SET4 #NXTCSR
REPEAT
  TST (R3) ;DOES THIS CSR RESPOND???
  BIS #1,R5 ;MARK IT IN CSR MAP
  CLR R4 ;CLEAR THE LAST CSR INDICATOR
  BIC #4,CSRINFO(R0) ;CLEAR UNUSED BITS
  BIS #BIT4!BIT3,CSRINFO(R0) ;YES-MARK IT IN CSR INFORMATION TABLE
  CLR (R3) ;CLEAR THE CSR UNDER TEST
  LET (R3) := #BIT13 ;IS THIS AN ECC MEMORY???
  IF #BIT13 SET IN (R3) ;IS BIT 13 SET FOR ECC MEMORY???
  CALL ECCTYPE ;FIGURE OUT WHAT KIND OF ECC MEMORY WE HAVE
END
CLR (R3) ;CLEAR CSR UNDER TEST
CALL RWCSR ;BIT TEST OF ALL BITS IN CSR'S
IF CSRINFO(R0) MI #30 ;DO WE HAVE A LEGAL CONFIGURATION?
MOV CSRINFO(R0),BAD ;MOVE IN BAD DATA
ERROR +21
END
NXTCSR:
ADD #2,R0 ;GO TO NEXT CSR
ADD #2,R3 ;GO TO NEXT CSR
ASL R5 ;SHIFT CSR MAP
ON.ERROR ;IS THERE A CSR 0
INC R4 ;YES-SET CSR PRESENT FLAG
END
UNTIL R0 EQ #40 ;UNTIL ALL CSR'S ARE DONE
ROR R5 ;RESYNC R5
TST R4 ;WAS THERE A CSR 0?
RNE 22$ ;BRANCH IF NOT EQUAL
BIS #BIT15,R5 ;YES SET IT IN CSR TABLE

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 157-1  
T1 BIT TEST OF ALL CSR'S

6116 005700  
6117 005704 004737 005774  
6118 005710

22\$: LET TOTCSRS := R5 ;STORE CSR MAP IN TOTCSRS  
CALL CSRMAP ;PRINT CSR MAP  
JUMPTO CTEST ;

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 158

T1 BIT TEST OF ALL CSR'S

6120 005714

SUBTST &lt;&lt;DETERM.NE TYPE OF ECC MEMORY&gt;&gt;

```

*****
*SUBTEST      DETERMINE TYPE OF ECC MEMORY
*****

```

```

THIS ROUTINE WILL DETERMINE IF THE ECC MEMORY UNDER TEST IS
A MS11-M OR A MS11-P

```

ECCTYPE:

```

6121
6122
6123
6124
6125 005714
6126 005714 052760 000001 002456
6127 005722
6128 005726
6129 005734 005013
6130 005736
6131 005742
6132 005746
6133 005754 052760 000002 002456
6134 005762
6135 005762
6136 005764 042760 000002 002456
6137 005772
6138 005772 000207

```

```

BIS #BIT0,CSRINFO(R0) ;MARK IT IN THE TABLE AS BEING A ECC MEMORY
LET (R3) := #60004 ;IS THIS A MS11-P???
IF #BIT11 SET.IN (R3) ;IS BIT 11 SET???
CLR (R3) ;CLEAR CSR
LET (R3) := #20004 ;ENABLE CHECK/SYNDROME BIT REGISTER
LET (R3) := #27744 ;IT IS BUT MAKE SURE AGAIN
IF (R3) EQ #23744 ;DO WE HAVE 6 OR 7 CHECK BITS IN CSR
BIS #BIT1,CSRINFO(R0) ;6 CHECK BITS-MARK IT A MS11-P
END ;
ELSE ;IT IS A MS11-M
BIC #BIT1,CSRINFO(R0) ;MARK IT IN THE TABLE
END ;
RETURN ;

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 159  
DETERMINE TYPE OF ECC MEMORY

6140 005774

6141 005774 005000  
6142 005776  
6143 006002  
6144 006006 005001  
6145 006010  
6146 006010 010102  
6147 006012 022702 000011  
6148 006016 100002  
6149 006020 062702 000007  
6150 006024 062702 000060  
6151 006030 110237 074704  
6152 006034  
6153 006040  
6154 006044 005201  
6155 006046  
6156 006054  
6157 006060  
6158 006060  
6159 006066  
6160 006076  
6161 006106 112737 000120 074704  
6162 006114  
6163 006116 112737 000115 074704  
6164 006124  
6165 006124  
6166 006124  
6167 006134 112737 000114 074704  
6168 006142  
6169 006142  
6170 006144 112737 000040 074704  
6171 006152  
6172 006152  
6173 006156  
6174 006162 000240  
6175 006164 062700 000002  
6176 006170  
6177 006176  
6178 006202 000207  
6179 006204 000000

```

CSRMAP: SUBTST <<PRINT CSR REGISTER MAP>>
*****
*SUBTEST PRINT CSR REGISTER MAP
*****
CLR R0 ;CLEAR CSR INFO POINTER
TYPE MSG008 ;PRINT TITLE
TYPE MSG016 ;PRINT CSR NUMBERS
CLR R1
REPEAT
MOV R1,R2
CMP #9,R2
BPL 1$ ;JUMP AROUND NEXT INSTRUCTION
ADD #7,R2
ADD #0,R2 ;MAKE IT ASCII
MOVB R2,MSG015
TYPE MSG015
TYPE MSG014 ;TYPE SINGLE SPACE
INC R1
UNTIL R1 EQ #16.
TYPE MSG009 ;TYPE MEMTYPE
REPEAT
IF CSRINFO(R0) NE #0 ;IS CSR NONEXSISTANT???
IF #BIT0 SET.IN CSRINFO(R0)
IF #BIT1 SET.IN CSRINFO(R0)
MOVB #'P,MSG015 ;IT IS A MS11-P
ELSE
MOVB #'M,MSG015 ;IT IS A MS11-M
END
END
IF #BIT1!BIT0 OFF.IN CSRINFO(R0)
MOVB #'L,MSG015 ;IT IS A MS11-L
END
ELSE
MOVB #' ,MSG015
END
TYPE MSG015 ;TYPE MEMORY TYPE
TYPE MSG014 ;TYPE SPACE
NOP
ADD #2,R0 ;POINT TO NEXT ENTRY
UNTIL R0 EQ #40
TYPE MSG129
RETURN
TRACE: .WORD 0

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 160  
PRINT CSR REGISTER MAP

6181 006206

6182  
6183  
6184  
6185  
6186  
6187  
6188  
6189  
6190  
6191 006206  
6192 006206  
6193 006216  
6194 006220 006205  
6195 006222  
6196 006226  
6197 006232  
6198 006242  
6199 006246  
6200 006250  
6201 006254  
6202 006254  
6203 006262 040537 002316  
6204 006266  
6205 006270  
6206 006272 040504  
6207 006274  
6208 006300  
6209 006304  
6210 006310 104035  
6211 006312 042760 000010 002456  
6212 006320  
6213 006320  
6214 006324  
6215 006326 005013  
6216 006330 040504  
6217 006332  
6218 006340  
6219 006346  
6220 006352 104010  
6221 006354 042760 000010 002456  
6222 006362  
6223 006362  
6224 006366  
6225  
6226 006366 005237 002262  
6227 006372  
6228 006400  
6229 006410  
6230 006414  
6231 006416  
6232 006422  
6233 006422  
6234 006422 005237 177640

```

SUBTST <<READ AND WRITE ALL CSR BITS>>
*****
*SUBTEST      READ AND WRITE ALL CSR BITS
*****
: THIS ROUTINE 'RWCSR' CHECK TO SEE THAT THE CSR CAN BEWRITTEN ON CORRECTLY
: BY WRITING AND CHECKING FOR THE FOLLOWING PATTERNS:
:
:     1-ZEROS
:     2-ONES
:     3-SHIFTING A ONE THROUGH A FIELD OF ZEROS
:     4-SHIFTING A ZEROS THROUGH A FIELD OF ONES
:
RWCSR:
PUSH R4,R5,UIPARO      ;SAVE R4,R5, AND UIPARO ON STACK
LET R5 := R0            ;GET CSR NUMBER FOR POSSIBLE ERROR
ASR R5
LET CSRNO := R5
LET ADDRESS := R3       ;GET ADDRESS FOR POSSIBLE ERROR
IF #BIT0 SET IN CSRINFO(R0) ;WHAT KIND OF MEMORY IS THIS???      ;GET BIT MASKS FOR D
    LET R5 := #17740    ;MASK FOR MS11-M/P
ELSE
    LET R5 := #70032    ;MASK FOR MS11-L
END
LET CSR1S := #177777    ;SET CSR1S TO ALL ONES
BIC R5,CSR1S            ;CLEAR BITS FOR GOOD DATA
LET (R3) := #0          ;0----->CSR
LET R4 := (R3)          ;MASK OUT UNWANTED BITS
BIC R5,R4
IF R4 NE #0             ;DO WE HAVE A CORRECT READ
    LET GOOD := #0      ;GOOD DATA=0'S
    LET CSR := R4       ;BAD DATA=CSR
    ERROR +35           ;BIT SET ERROR
    BIC #BIT3,CSRINFO(R0) ;CLEAR CSR OK BIT
END
LET (R3) := CSR1S       ;ONES--->(R3)
LET R4 := (R3)          ;MASK OUT CORRECT FIELD
CLR (R3)                ;CLEAR OUT CSR
BIC R5,R4
IF R4 NE CSR1S          ;WAS PATTERN WRITTEN CORRECTLY?
    LET GOOD := CSR1S   ;GOOD DATA = ALL LEGAL BITS SET IN CSR
    LET CSR := R4       ;BAD DATA=CSR
    ERROR +10           ;BIT CLEAR ERROR
    BIC #BIT3,CSRINFO(R0) ;CLEAR CSR OK BIT
END
LET PASFLG := #0        ;SET UP LOOP COUNTER
REPEAT
    INC PASFLG          ;INCREMENT LOOP COUNTER
    LET UIPARO := #-1   ;USE USER PAR FOR BIT COUNTER
    IF PASFLG EQ #1     ;PASS 1
        LET R2 := #1    ;1----->FIELD OF ZEROS
    ELSE
        LET R2 := #177776 ;0----->FIELD OF ONES
    END
    REPEAT              ;DO BITS 0-4 AND 13-15
        INC UIPARO      ;INCREMENT BIT POINTER

```



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 160-1  
READ AND WRITE ALL CSR BITS

6235	006426				IF PASFLG EQ #2 AND #BIT0 OFF.IN CSRINFO(R0) ;
6236	006446	042702	040004		BIC #BIT14,BIT2,R2 ;IF THIS IS PASS 2 ON A MS11-L, CLEAR EUB BIT AND WRITE
6237	006452				END
6238	006452				LET (R3) := R2 ;WRITE DATA
6239	006454				LET R1 := R2 ;GET GOOD DATA AND MASK IT OUT
6240	006456	040501			BIC R5,R1 ;GET GOOD DATA
6241	006460				LET R4 := (R3) ;GET DATA THAT IS READ
6242	006462	040504			BIC R5,R4 ;MASK OUT CSR BITS
6243	006464				IF R1 NE R4 ;IS DATA CORRECT???
6244	006470				LET BAD := R4 ;BAD DATA = CSR CONTENTS
6245	006474				LET CSR := R1 ;GET GOOD DATA
6246	006500				IF PASFLG EQ #1 ;SELECT ERROR DEPENDING ON PASS
6247	006510	104035			ERROR +35 ;BIT SET ERROR
6248	006512				ELSE ;PASS 2
6249	006514	104010			ERROR +10 ;BIT CLEAR ERROR
6250	006516				END
6251	006516	042760	000010	002456	BIC #BIT3,CSRINFO(R0) ;CLEAR CSR OK BIT
6252	006524				END
6253	006524				IF PASFLG EQ #1 ;GET DATA FOR NEXT LOOP
6254	006534	006302			ASL R2 ;SHIFT 1 ACROSS 0'S
6255	006536				ELSE ;
6256	006540	000261			SEC ;SET CARRY
6257	006542	006102			ROL R2 ;ROTATE A 0 ACROSS A FIELD OF ONES
6258	006544				END
6259	006544				UNTIL UIPARO EQ #15. ;UNTIL ALL BITS ARE DONE
6260	006554				UNTIL PASFLG EQ #2 ;DONE WITH 2 PASSES
6261	006564				IF #BIT0 SET.IN CSRINFO(R0) THEN JUMPTO DONE ;IF MS11-L DO ONE LAST WRITE
6262	006600				LET (R3) := #140005 ;WRITE ONES TO CSR WITH EUB BIT ENABLED
6263	006604				LET R2 := (R3) ;READ CSR FOR CORRECT BITS
6264	006606	042702	037772		BIC #37772,R2 ;CLEAR UNWANTED BITS
6265	006612				IF R2 NE #140005 ;WAS WRITE CORRECT
6266	006620				LET GOOD := #140005 ;GOOD DATA
6267	006626				LET CSR := R2 ;BAD DATA
6268	006632	104010			ERROR +10 ;BIT CLEAR ERROR
6269	006634	042760	000010	002456	BIC #BIT3,CSRINFO(R0) ;CLEAR CSR OK BIT!
6270	006642				END
6271	006642			DONE:	LET (R3) := #0 ;CLEAR OUT CSR
6272	006644				PJP UIPARO,R5,R4 ;RESTORE UIPARO,R4, AND R5
6273	006654	000207			RETURN
6274					

6277					:THE FOLLOWING ROUTINE DETERMINES WHICH CSR CONTROLS PROGRAM SPACE
6278					:
6279	006656	104424			CTEST: CACHOFF
6280	006660	012737	177777	002526	MOV #177777,PGMCSR
6281	006666	C12737	002000	172350	MOV #2000,KIPAR4 ;SET UP MAP REGISTER
6282	006674	012701	002406		MOV #TESTADD,R1
6283	006700	012737	100000	002406	MOV #100000,TESTADD
6284	006706	012737	100002	002410	MOV #100002,TESTADD+2
6285	006714	005000			CLR R0 ;CLEAR CSR COUNTER
6286	006716	005037	002150		CLR CSRNO
6287	006722	013703	002222		MOV TOTCSRS,R3 ;OBTAIN CSR MAP
6288	006726	000240			NOP ;DEBUG AID
6289	006730	006303		4\$:	ASL R3 ;PUT HIGH ORDER BIT INTO C BIT
6290	006732	103407			BCS 2\$ ;BRANCH IF CSR EXISTS
6291	006734	062700	000002	1\$:	ADD #2,R0 ;UPDATE CSR COUNTER
6292	006740	010037	002150		MOV R0,CSRNO
6293	006744	005703			TST R3 ;IS MAP EMPTY?
6294	006746	001474			BEQ 3\$ ;BRANCH IF SO
6295	006750	000767			BR 4\$
6296	006752	000240		2\$:	NOP ;DEBUG AID
6297	006754	000241			CLC ;CLEAR CARRY
6298	006756	032760	000003	002456	BIT #BIT1!BIT0,CSRINFO(R0) ;IS THIS PARITY MEMORY?
6299	006764	001014			BNE 5\$ ;BRACH IF NOT
6300	006766	052760	000004	172100	BIS #BIT2,CSRADD(R0) ;SET WRITE WRONG PARITY
6301	006774	012771	123456	000000	MOV #123456,a(R1) ;WRITE DATA
6302	007002	012771	123456	000002	MOV #123456,a2(R1)
6303	007010	005060	172100		CLR CSRADD(R0) ;RESTORE CSR
6304	007014	000414			BR 6\$
6305	007016	012760	000000	172100	5\$: MOV #0,CSRADD(R0) ;CLEAR THE CSR UNDER TEST
6306	007024	012771	123456	000000	MOV #123456,a(R1) ;WRITE DATA
6307	007032	012771	123456	000002	MOV #123456,a2(R1)
6308	007040	012760	020006	172100	MOV #20006,CSRADD(R0) ;SET DIAG CHECK MODE
6309	007046	005771	000000		6\$: TST a(R1) ;WRITE CHECKBITS TO CSR
6310	007052	016004	172100		MOV CSRADD(R0),R4 ;WRITE CSR TO R4
6311	007056	032760	000003	002456	BIT #BIT1!BIT0,CSRINFO(R0) ;PARITY MEMORY?
6312	007064	001003			BNE 7\$ ;BRANCH IF NOT
6313	007066	005704			TST R4 ;PARITY ERROR?
6314	007070	100421			BMI 8\$ ;BRACH IF SO
6315	007072	000720			BR 1\$ ;TRY NEXT CSR
6316	007074	000240		7\$:	NOP ;DEBUG AID
6317	007076	072427	177773		ASH #-5,R4
6318	007102	042704	177600		BIC #^C177,R4
6319	007106	032760	000002	002456	BIT #BIT1,CSRINFO(R0) ;WHAT KIND OF ECC MEMORY IS THIS
6320	007114	001003			BNE 10\$ ;BRANCH IF MS11-P
6321	007116	012702	000157		MOV #157,R2 ;LOAD IN CORRECT CHECK BITS FOR MS11-M
6322	007122	000402			BR 11\$
6323	007124	012702	000040		10\$: MOV #40,R2 ;CORRECT CHECK BITS FOR MS11-P
6324	007130	020204		11\$:	CMP R2,R4 ;CORRECT CHECKBITS?
6325	007132	001300			BNE 1\$ ;BRANCH IF NOT
6326	007134	010037	002526	8\$:	MOV R0,PGMCSR
6327	007140	000240		3\$:	NOP ;DEBUG AID
6328	007142	104502			CLRCSR ;CLEAR ALL CSR'S
6329	007144	012771	000000	000000	MOV #0,a(R1) ;RESTORE TEST LOCATIONS
6330	007152	012771	000000	000002	MOV #0,a2(R1)
6331	007160	023727	002526	177777	CMP PGMCSR,#177777
6332	007166	001402			BEQ FINT ;IF PROGRAM CSR NOT FOUND GO TO FINT
6333	007170	000137	007642		JMP CLRMEM ;GO TO SIZING ROUTINE IF FOUND

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 163  
READ AND WRITE ALL CSR BITS

```

6335
6336      : IF PGMCSR WAS NOT FOUND BY THE PRECEEDING ROUTINE, THIS ROUTINE TRIES
6337      : TO FIND IT FOR INTERLEAVED MEMORIES
6338
6339 007174      FINT:  SET4  #2$      :NE MEMORY TRAPS GO TO 2$
6340 007202      1$:  MOV   #123456,@(R1)  :WRITE DATA AT FIRST LOCATION OF BANK 2 IN BOARD
6341 007210      012771 123456 000002      MOV   #123456,@2(R1)  :WRITE DATA AT SECOND LOCATION OF BANK 2 IN BOARD
6342 007216      062737 010000 172350      ADD   #10000,KIPAR4  :UPDATE PAR4 TO POINT TO UPPER BOARDS
6343 007224      000766      BR      1$      :KEEP GOING TILL NO MORE MEMORY
6344 007226      012700 177776      2$:  MOV   #-2,R0
6345 007232      013703 002222      MOV   TOTCSRS,R3  :PUT CSR MAP IN R3
6346 007236      062700 000002      3$:  ADD   #2,R0      :UPDATE CSR COUNTER
6347 007242      010037 002150      MOV   R0,CSRNO      :UPDATE CSRNO
6348 007246      006303      ASL   R3
6349 007250      103403      BCS   4$      :BRANCH IF CSR EXISTS
6350 007252      005703      TST   R3      :ANY CSR'S LEFT?
6351 007254      001405      BEQ   5$      :BRANCH IF NOT
6352 007256      000767      BR      3$      :LOOK FOR NEXT CSR
6353 007260      012760 020006 172100 4$:  MOV   #20006,CSRADD(R0)  :SET DIAGNOSTIC CHECK MODE IN CSR
6354 007266      000763      BR      3$      :LOOK FOR NEXT CSR
6355 007270      5$:  SET4  #6$      :NE MEMORY TRAPS NOW GO TO 6$
6356 007276      012700 177776      MOV   #-2,R0      :RESET CSR POINTER
6357 007302      012737 002000 172350      MOV   #2000,KIPAR4  :REMAP PAR4 TO POINT TO BANK 2
6358 007310      005771 000000      TST   @(R1)      :TEST NONASSERTED LOCATIONS
6359 007314      062700 000002      6$:  ADD   #2,R0      :UPDTAE CSR POINTER
6360 007320      010037 002150      MOV   R0,CSRNO
6361 007324      022700 000040      CMP   #40,R0
6362 007330      001535      BEQ   10$      :NOT FOUND?
6363 007332      032760 000002 002456      BIT   #BIT1,CSRINFO(R0)  :BRANCH IF NOT
6364 007340      001003      BNE   55$      :GET TYPE OF ECC MEMORY
6365 007342      012702 000157      MOV   #157,R2      :BRANCH IF MS11-P
6366 007346      000402      BR      56$      :MS11-M CHECK BITS
6367 007350      012702 000040      55$:  MOV   #40,R2      :MS11-P CHECK BITS
6368 007354      016004 172100      56$:  MOV   CSRADD(R0),R4      :GET CSR CONTENTS
6369 007360      072427 177773      ASH   #-5,R4
6370 007364      042704 177600      BIC   #^C177,R4
6371 007370      020204      CMP   R2,R4
6372 007372      001401      BEQ   7$      :CLEAR ALL BUT CHECKBITS
6373 007374      000747      BR      6$      :PROPER CHECKBITS?
6374 007376      110037 002526      7$:  MOVB  R0,PGMCSR      :BRANCH IF SO
6375 007402      012700 177776      SET4  #8$      :TRY NEXT CSR IF NOT
6376 007410      013703 002222      MOV   #-2,R0      :WRITE NON-ASSERTED CSR # IN PGMCSR
6377 007414      062700 000002      23$:  MOV   TOTCSRS,R3      :NE TRAPS GO TO 8$
6378 007420      010037 002150      ADD   #2,R0
6379 007424      006303      MOV   R0,CSRNO
6380 007430      103403      ASL   R3
6381 007432      005703      BCS   24$      :PUT CSR MAP IN R3
6382 007434      001405      TST   R3      :UPDATE CSR COUN
6383 007436      000767      BEQ   25$      :UPDATE CSRNO
6384 007440      012760 020006 172100 24$:  MOV   #20006,CSRADD(R0)  :BRANCH IF CSR EXISTS
6385 007442      000763      BR      23$      :ANY CSR'S LEFT?
6386 007450      012700 177776      25$:  MOV   #-2,R0      :BRANCH IF NOT
6387 007452      005771 000002      TST   @2(R1)      :LOOK FOR NEXT CSR
6388 007456      062700 000002      8$:  ADD   #2,R0      :SET DIAGNOSTIC CHECK MODE IN CSR
6389 007462      010037 002150      MOV   R0,CSRNO      :LOOK FOR NEXT CSR
6390 007466      022700 000040      CMP   #40,R0
6391 007472

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 163-1  
 READ AND WRITE ALL CSR BITS

6392	007476	001452			BEQ	10\$		
6393	007500	032760	000002	002456	BIT	#BIT1,CSRINFO(R0)		:CHECK FOR TYPE OF ECC MEMORY
6394	007506	001003			BNE	76\$		:BRANCH IF MS11-P
6395	007510	012702	000157		MOV	#157,R2		:CHECK BITS FOR MS11-M
6396	007514	000402			BR	77\$		
6397	007516	012702	000040		MOV	#40,R2	76\$:	:CHECK BITS FOR MS11-P
6398	007522	016004	172100		MOV	CSRADD(R0),R4	77\$:	
6399	007526	072427	177773		ASH	#-5,R4		
6400	007532	042704	177600		BIC	#^C177,R4		
6401	007536	020204			CMP	R2,R4		:PROPER CHECKBITS?
6402	007540	001401			BEQ	9\$		:BRANCH IF SO
6403	007542	000747			BR	8\$		:TRY NEXT CSR IF NOT
6404	007544	110037	002527		MOVB	R0,PGMCSR+1	9\$:	:WRITE ASSERTED CSR # IN PGMCSR
6405	007550	052737	100000	002526	BIS	#BIT15,PGMCSR		:SET INTERLEAVED INDICATOR IN PGMCSR
6406	007556	104502			CLRCR			
6407	007560	012737	002000	172350	MOV	#2000,KIPAR4		
6408	007566				SET4	#12\$		:NE MEMORY TRAPS GO TO 12\$
6409	007574	012771	000000	000000	MOV	#0,@(R1)	11\$:	:WRITE DATA AT FIRST LOCATION OF BANK 2 IN BOARD
6410	007602	012771	000000	000002	MOV	#0,@2(R1)		:WRITE DATA AT SECOND LOCATION OF BANK 2 IN BOARD
6411	007610	062737	010000	172350	ADD	#10000,KIPAR4		:UPDATE PAR4 TO POINT TO UPPER BOARDS
6412	007616	000766			BR	11\$		
6413	007620	104423			CACHON		12\$:	
6414	007622	000407			BR	CLRMEM		
6415	007624	012737	001000	172350	MOV	#1000,KIPAR4	10\$:	
6416	007632				TYPE	MSG126		:ERROR - PROGRAM CSR NOT FOUND!
6417	007636	005037	002526		CLR	PGMCSR		:SET TO DEFAULT OF 0

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 164  
 READ AND WRITE ALL CSR BITS

6419 007642

SUBTST <<CLEAR ALL MEMORY SPACE FROM BANK 2 ON>>

\*\*\*\*\*  
 \*SUBTEST CLEAR ALL MEMORY SPACE FROM BANK 2 ON  
 \*\*\*\*\*

THIS ROUTINE CLEARS ALL MEMORY SPACE BEGINNING AT ADDRESS 200,000 AND  
 CONTINUES UNTIL THERE IS NO MEMORY LEFT. IT SHOULD CLEAR ANY PARITY ERRORS  
 CREATED BY THE LAST ROUTINE, AND CLEAN UP ANY JUNK LEFT HANGING AROUND IN  
 HIGHER MEMORY.

6420  
 6421  
 6422  
 6423  
 6424  
 6425

6426	007642			
6427	007650	005037	006204	
6428	007654	012737	000001	002074
6429	007662	012737	002000	172350
6430	007670	012701	100000	
6431	007674	020127	117776	
6432	007700	001003		
6433	007702	012737	177777	006204
6434	007710	005021		
6435	007712	005737	006204	
6436	007716	001001		
6437	007720	000765		
6438	007722	062737	000200	172350
6439	007730	022737	170000	172350
6440	007736	001405		
6441	007740	005037	006204	
6442	007744	012701	100000	
6443	007750	000751		
6444	007752	000240		
6445	007754	005037	006204	
6446	007760			

CLRMEM:	SET4	#CLREX	;NONEM TRAPS GO TO CLREX
	CLR	TRACE	
	MOV	#1,NOPAR	;IGNORE PARITY ERRORS
	MOV	#200,KIPAR4	;SET UP MAP TO START AT BANK 2
	MOV	#100000,R1	;R1 MAPS TO KIPAR4
1\$:	CMP	R1,#117776	;WHOLE 16K BANK DONE?
	BNE	2\$	;KEEP GOING IF NOT
	MOV	#-1,TRACE	;USE TRACE FLAG TO FLAG END OF BANK
2\$:	CLR	(R1)+	;CLEAR CONTENTS & INCREMENT
	TST	TRACE	;EOB FLAG SET?
	BNE	3\$	;GO TO NEXT BANK IF SO
	BR	1\$	
3\$:	ADD	#200,KIPAR4	;SET MAP FOR NEXT BANK
	CMP	#170000,KIPAR4	;ARE WE AT THE PERIPHERIAL PAGE
	BEQ	CLREX	;YES-GO ON
	CLR	TRACE	;RESET FLAG
	MOV	#100000,R1	;RESET R1
	BR	1\$	;CLEAR NEXT BANK
CLREX:	NOP		
	CLR	TRACE	
	RES4		

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 166  
CLEAR ALL MEMORY SPACE FROM BANK 2 ON

6449 010002

ANA2: SUBST &lt;&lt;MATCH ALL CSR'S WITH MEMORY&gt;&gt;

\*\*\*\*\*  
\*SUBTEST MATCH ALL CSR'S WITH MEMORY  
\*\*\*\*\*

\* THE SECOND PART OF THE ANALYSIS MATCHES UP THE CSR'S WITH THE MEMORY, AND  
\* DIALS ALL THE INFORMATION FOUND IN THE CONFIGURATION TABLE. FOR ECC,  
\* THIS IS DONE BY TAKING EACH CSR FOUND IN THE PREVIOUS SECTION SEQUENTIALLY  
\* AND CHECKING THROUGH ALL OF MEMORY, ONE BANK AT A TIME, TO SEE WHICH BANKS  
\* RESPOND TO THE CSR IN QUESTION. THE FIRST DOUBLE WORD PAIR IN EACH BANK ARE  
\* WRITTEN WITH DATA AND DIAGNOSTIC CHECK MODE SET IN THE CSR IN ORDER TO AC-  
\* COMPLISH THIS. ALL POSSIBLE CONFIGURATIONS OF DOUBLE WORD PAIRS (NON-INTER-  
\* LEAVED, 64K INTERLEAVED, OR 128K INTERLEAVED) ARE CHECKED FOR EACH BANK  
\* THROUGH USE OF TESTADD AND KERNEL INSTRUCTION PAGE ADDRESS REGISTERS 4 AND  
\* 5. IF WE GET THE PROPER CHECKBITS BACK, WE HAVE A MATCH. IF NOT, THE ROUT-  
\* INE CHECKS FOR SINGLE OR DOUBLE BIT ERRORS.

\* IF ONE OR THE OTHER IS FOUND, THE ERROR ADDRESS IS CHECKED  
\* TO SEE IF IT IS THAT BANK. IF IT IS, WE HAVE A MATCH. AT THE END OF EACH  
\* BANK PASS, FOR EACH CSR PASS, THE PROGRAM COMES UP WITH A NUMBER, STORED IN  
\* "I", WHICH DENOTES THE FOLLOWING:

I	MEMORY DESCRIPTION
-	-----
0	NON-EXISTANT MEMORY
1	NON-INTERLEAVED MEMORY MS11-L,MS11-P
2	64K INTERLEAVED, A1 NOT ASSERTED MEMORY
3	128K INTERLEAVED, A1 NOT ASSERTED MEMORY
4	64K INTERLEAVED, A1 ASSERTED MEMORY
5	128K INTERLEAVED, A1 ASSERTED MEMORY

\* NOTE - I=2 THROUGH I=5 CAN ONLY OCCUR WITH MS11-M MEMORY.

\* NOTE THAT PARITY MEMORY WRITES WRONG PARITY TO THE DOUBLE WORDS, THEN LOOKS  
\* FOR THE PARITY ERROR BIT TO BE SET. IF THE BIT IS SET, WE HAVE A MATCH.

6480 010002			SET4	#100\$	:NE MEMORY TRAPS GO TO 100\$
6481 010010	005037	002310	CLR	CHECK	:CLEAR CHECK
6482 010014	012701	002406	MOV	#TESTADD,R1	:SET UP THE VIRTUAL ADDR. POINTER
6483 010020	013703	002222	MOV	TOTCSRS,R3	:MOVE CSR MAP INTO R3
6484 010024	005000		CLR	R0	:CLEAR THE CSR POINTER
6485 010026	005005		CLR	R5	:CLEAR THE PROGRAM CSR STATUS POINTER
6486 010030	005737	002450	TST	NO22BIT	:IS THIS AN 11/44 OR 11/24?
6487 010034	001403		BEQ	7\$	:BRANCH IF IT IS
6488 010036	005037	002554	CLR	LASTBLOCK	:ADJUST LASTBLOCK INDICATOR FOR 124K MACHINE
6489 010042	000413		BR	1\$	:BRANCH OVER NEXT PIECE OF CODE
6490 010044	022737	000167	CMP	#167, LASTBANK	:IS THERE UNIBUS MEMORY ABOVE 17000000?
6491 010052	001407		BEQ	1\$	:BRANCH IF NOT
6492 010054	013702	002552	MOV	LASTBANK,R2	:SET UP A NEW LAST BLOCK INDICATOR
6493 010060	005202		INC	R2	
6494 010062	072227	000011	ASH	#9,R2	
6495 010066	010237	002554	MOV	R2, LASTBLOCK	
6496 010072	012702	000004	MOV	#4,R2	:R2 IS INDEX FOR CONFIG TABLE
6497 010076	012737	001000	MOV	#1000,KIPAR4	:SET KIPAR4 FOR BANK 1
6498 010104	012737	001000	MOV	#1000,KIPAR5	:SET KIPAR5 FOR BANK 1
6499 010112	006303		ASL	R3	:DOES THIS CSR EXIST?
6500 010114	103420		BCS	3\$	:BRANCH IF IT DOES EXIST
6501 010116	062700	000002	ADD	#2,R0	:INCREMENT THE CSR POINTER
6502 010122	010037	002150	MOV	R0, CSRNO	:STORE IT IN CSRNO ALSO

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 166-1  
MATCH ALL CSR'S WITH MEMORY

6503	010126	005703			TST	R3		:ARE THERE ANY MORE CSR'S TO DO?
6504	010130	001370			BNE	2\$		:BRANCH IF ALL CSRS NOT DONE
6505	010132	012737	001000	172350	MOV	#1000,KIPAR4		:RESTORE KIPAR4
6506	010140	012737	001200	172352	MOV	#1200,KIPAR5		:RESTORE KIPAR5
6507	010146	013706	002560		MOV	KSTACK,SP		:RESTORE STACK
6508	010152	000137	011400		JMP	SUBAAS		:JUMP TO SUBAAS IF ALL CSR'S ARE DONE
6509	010156	010037	002150		MOV	R0,CSRNO		:MAKE SURE CSRNO IS UPDATED
6510	010162	104424			3\$: CACHOFF			:TURN THE CACHE OFF
6511	010164	000240			13\$: NOP			
6512	010166	012737	100000	002406	45\$: MOV	#100000,TESTADD		:SET UP VIRTUAL ADDRESS TO KIPAR4
6513	010174	012737	120002	002410	MOV	#120002,TESTADD+2		:SET UP VIRTUAL ADDRESS TO KIPAR5
6514	010202	032762	000040	002650	BIT	#BIT5,CONFIG(R2)		:IS THIS A BANK TO SKIP?
6515	010210	001402			BEQ	43\$		:NO - BRANCH AROUND NEXT INSTRUCTION
6516	010212	000137	011314		JMP	6\$		:YES - GO TO END OF BANK
6517	010216	005037	002446		43\$: CLR	I		:CLEAR THE MEMORY CONFIGURATION COUNTER
6518	010222	005771	000000		4\$: TST	@(R1)		:TEST TO SEE THAT THERE IS MEMORY PRESENT
6519	010226	005237	002446		INC	I		
6520	010232				PUSH	@(R1),@2(R1)		:SAVE THE LOCATIONS UNDER TEST
6521	010242	032760	000003	002456	BIT	#BIT1!BIT0,CSRINFO(R0)		:IS THIS PARITY MEMORY?
6522	010250	001014			BNE	34\$		:NO - BRANCH
6523	010252	052760	000004	172100	BIS	#BIT2,CSRADD(R0)		:SET WRITE WRONG PARITY
6524	010260	012771	123456	000000	MOV	#123456,@(R1)		:SET THE FIRST LOCATION UNDER TEST
6525	010266	012771	123456	000002	MOV	#123456,@2(R1)		:SET THE SECOND LUT
6526	010274	005060	172100		CLR	CSRADD(R0)		:CLEAR THE CSR
6527	010300	000411			BR	41\$		:TEST LOCATIONS
6528	010302	012771	123456	000000	34\$: MOV	#123456,@(R1)		:SET THE FIRST LOCATION UNDER TEST
6529	010310	012771	123456	000002	MOV	#123456,@2(R1)		:SET THE SECOND LUT
6530	010316	104503			CLR1CSR			:RESET CSR
6531	010320	104475			CB1CSR			:SET DIAG. CHECK MODE IN CSR UNDER TEST
6532	010322	000240			NOP			:DEBUG AID
6533	010324	005771	000000		41\$: TST	@(R1)		:READ THE FIRST LUT TO WRITE CKBITS. INTO CSR
6534	010330	104426			READCSR			:READ THE CSR UNDER TEST
6535	010332	000240			NOP			:DEBUG AID
6536	010334	013704	002146		MOV	CSR,R4		:GET THE CHECKBITS FROM THE CSR
6537	010340	000240			NOP			:DEBUG AID
6538	010342	010437	002430		MOV	R4,TEMP		:SAVE IN TEMP FOR LATER
6539	010346	104503			CLR1CSR			:RESET CSR
6540	010350				POP	@2(R1),@(R1)		:RESTORE LOCATIONS UNDER TEST
6541	010360	032760	000003	002456	BIT	#BIT1!BIT0,CSRINFO(R0)		:IS THIS PARITY MEMORY?
6542	010366	001004			BNE	42\$		:NO - BRANCH
6543	010370	005704			TST	R4		:DID WE GET A PARITY ERROR?
6544	010372	100431			BMI	25\$		:YES - FILL IN CONFIG TABLE
6545	010374	000137	011314		JMP	6\$		:NO - JUMP TO END OF BANK
6546	010400	072427	177773		42\$: ASH	#-5,R4		:MANIPULATE THE CSR BITS
6547	010404	042704	177600		BIC	#^C177,R4		:INTO A USABLE FORM.
6548	010410	032760	000002	002456	BIT	#BIT1,CSRINFO(R0)		:WHAT KIND OF ECC MEMORY IS THIS??
6549	010416	001005			BNE	76\$		:BRANCH IF MS11-P
6550	010420	012737	000157	002312	MOV	#157,CBITS		:MS11-M CHECK BITS
6551	010426	000240			NOP			:DEBBUG AIDE
6552	010430	000404			BR	77\$		
6553	010432	012737	000040	002312	76\$: MOV	#40,CBITS		:MS11-P CHECK BITS
6554	010440	000240			NOP			:DEBBUGING AIDE
6555	010442	023704	002312		77\$: CMP	CBITS,R4		:DO THE CHECKBITS COMPARE TO WHAT WAS WRITTEN?
6556	010446	000240			NOP			:DEBBUG AIDE
6557	010450	001402			BEQ	25\$		:BRANCH IF THERE IS A MATCH
6558	010452	000137	011002		JMP	22\$		:ELSE BRANCH IF NOT THE SAME
6559					;*			

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 166-2  
MATCH ALL CSR'S WITH MEMORY

```

6560      ;* WE COME HERE IF THERE IS A MATCH
6561      ;*
6562 010456 010004      25$: MOV R0,R4      ;GET THE CSR NUMBER
6563 010460 000240      NOP
6564 010462 006204      ASR R4      ;SET IT UP FOR USE IN THE
6565 010464 000304      SWAB R4      ;CONFIGURATION TABLE.
6566 010466 042704 170377 BIC #170377,R4 ;CLEAR OFF EXTRANEIOUS BITS
6567 010472 032737 000004 002446 BIT #BIT2,I ;INTERLEAVED A1 ASSERTED MEMORY FOUND?
6568 010500 001402      BEQ 15$      ;BRANCH IF NOT
6569 010502 072427 000004      ASH #4,R4 ;PUT CSR NUMBER IN INTERLEAVED CSR SLOT
6570 010506 050462 002650      15$: BIS R4,CONFIG(R2) ;PUT CSR NUMBER IN CONFIG. TABLE
6571 010512 016004 002456      MOV CSRINFO(R0),R4 ;GET MEMORY TYPE
6572 010516 042704 177770      BIC #^C7,R4 ;CLEAR OFF THE EXTRANEIOUS BITS
6573 010522 000304      SWAB R4      ;MOVE INTO PROPER POSITION
6574 010524 050462 002652      BIS R4,CONFIG+2(R2) ;SET IT INTO THE CONFIG TABLE
6575 010530 022737 000001 002446 CMP #1,I ;WAS THIS NON-INTERLEAVED MEMORY?
6576 010536 001431      BEQ 24$      ;BRANCH IF IT WAS
6577 010540 052762 010000 002652 BIS #BIT12,CONFIG+2(R2) ;SET THE INTERLEAVED BIT
6578 010546 010204      MOV R2,R4 ;SAVE THE CURRENT BANK INDEX
6579 010550 032737 000001 002446 BIT #BIT0,I ;WAS THIS 128K INTERLEAVED?
6580 010556 001006      BNE 5$      ;BRANCH IF TRUE
6581 010560 052762 004000 002652 BIS #BIT11,CONFIG+2(R2) ;SET 64K INTERLEAVED FLAG IN CONFIG
6582 010566 062704 000020      ADD #20,R4 ;SET NEW BANK POINTER TO 4 BANKS AHEAD
6583 010572 000402      BR 16$      ;JUMP OVER NEXT INSTRUCTION
6584 010574 062704 000040      5$: ADD #40,R4 ;SET NEW BANK POINTER 8 BANKS AHEAD
6585 010600 052764 000040 002650 16$: BIS #BIT5,CONFIG(R4) ;SET SK'P ECC LOGIC TESTS FLAG
6586 010606 056264 002650 002650 BIS CONFIG(R2),CONFIG(R4) ;SET OTHER INFO INTO THAT BANK
6587 010614 056264 002652 002652 BIS CONFIG+2(R2),CONFIG+2(R4)
6588      ;*
6589      ;* THIS SECTION IS EXECUTED ONLY WHEN THE BANK=1
6590      ;*
6591 010622 022737 001000 172350 24$: CMP #1000,KIPAR4 ;IS THIS BANK 1 ?
6592 010630 001402      BEQ 30$      ;BRANCH IF TRUE
6593 010632 000137 011154      JMP 33$ ;ELSE JUMP TO END OF THIS BANK
6594 010636 032737 100020 002430 30$: BIT #BIT15!BIT4,TEMP ;WAS THERE A SBE OR DBE?
6595 010644 001417      BEQ 10$      ;BRANCH IF NOT
6596 010646 013704 002430      MOV TEMP,R4 ;GET CSR CONTENTS
6597 010652 072427 177767      ASH #-9,R4 ;MAKE ERROR ADDRESS INTO BANK #
6598 010656 022704 000001      CMP #1,R4 ;ERROR IN BANKS 0 OR 1?
6599 010662 003010      BGT 10$      ;BRANCH IF NOT
6600 010664 052762 000001 002650 BIS #BIT0,CONFIG(R2) ;SET ERROR FLAG IN CONFIG TABLE
6601 010672 105262 002652      INCB CONFIG+2(R2) ;ADD ONE TO BANK ERROR COUNT
6602 010676      SET CFGERROR ;PRINT CONFIG TABLE
6603 010704 053737 002654 002650 10$: BIS CONFIG+4,CONFIG ;SET UP INFORMATION IN BANK ZERO
6604 010712 053737 002656 002652 BIS CONFIG+6,CONFIG+2
6605 010720 000240      NOP
6606 010722 022737 000001 002446 CMP #1,I ;DEBUG AID
6607 010730 001002      BNE 46$      ;WAS THIS NON-INTERLEAVED MEMORY
6608 010732 000137 011314      JMP 6$ ;NO - BRANCH OVER NEXT STMT.
6609 010736 012704 000020      46$: MOV #20,R4 ;YES - JUMP TO END OF THIS BANK
6610 010742 032737 000001 002446 BIT #BIT0,I ;SET UP COUNTER FOR 64K INTERLEAVED
6611 010750 001402      BEQ 26$      ;WAS IT 128K INTERLEAVED?
6612 010752 062704 000020      ADD #20,R4 ;BRANCH IF NOT
6613 010756 053764 002650 002650 26$: BIS CONFIG,CONFIG(R4) ;SET UP COUNTER FOR 128K INTERLEAVED
6614 010764 053764 002652 002652 BIS CONFIG+2,CONFIG+2(R4) ;SET OTHER BANK WITH SAME INFORMATION
6615 010772 052764 000040 002650 BIS #BIT5,CONFIG(R4) ;AS IN BANK 0
6616 011000 000465      BR 33$      ;SET SKIP ECC LOGIC TESTS FLAG
;BRANCH

```



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 166-3  
MATCH ALL CSR'S WITH MEMORY

```

6617
6618
6619
6620 011002 032737 100020 002146 22$: BIT #BIT15:BIT4,CSR ;SBE OR DBE FLAGS SET?
6621 011010 001001 BNE 8$ ;BRANCH IF TRUE
6622 011012 000460 BR 33$ ;CHECK TO SEE IF IT IS MS11-M
6623 011014 013704 002150 8$: MOV CSRN0,R4 ;GET CSRN0
6624 011020 042764 000006 172100 BIC #6,CSRADD(R4) ;TURN OFF DIAG CHECK & ECC DISABLE
6625 011026 PUSH R0,R1 ;SAVE R0 & R1
6626 011032 016401 172100 MOV CSRADD(R4),R1 ;GET CSR INFORMATION
6627 011036 072127 177773 ASH #-5,R1 ;SET UP ERROR ADDRESS
6628 011042 042701 177600 BIC #^C177,R1
6629 011046 052764 040000 172100 BIS #BIT14,CSRADD(R4) ;GET EXTENDED ERROR ADDRESS BITS
6630 011054 016400 172100 MOV CSRADD(R4),R0 ;READ FROM CSR
6631 011060 042764 040000 172100 BIC #BIT14,CSRADD(R4) ;TURN OFF EUB BIT
6632 011066 042700 177037 BIC #^C740,R0 ;SET UP EXTENDED BITS
6633 011072 006300 ASL R0
6634 011074 006300 ASL R0
6635 011076 060001 ADD R0,R1 ;SET UP TOTAL ERROR ADDRESS
6636 011100 010104 27$: MOV R1,R4 ;SAVE IN R4
6637 011102 POP R1,R0 ;RESTORE R0 & R1
6638 011106 072427 000005 ASH #5,R4 ;SET ERROR ADDRESS UP IN PAR NOTATION
6639 011112 020437 172350 CMP R4,KIPAR4 ;DOES IT EQUAL KIPAR4?
6640 011116 001001 BNE 28$ ;BRANCH IF FALSE
6641 011120 000403 BR 35$ ;YES - MARK INFO IN CONFIG TABLE
6642 011122 020437 172352 28$: CMP R4,KIPAR5 ;DOES IT EQUAL KIPAR5?
6643 011126 001012 BNE 33$ ;BRANCH IF FALSE
6644 011130 052762 000001 002650 35$: BIS #^D0,CONFIG(R2) ;SET BANK ERROR FLAG
6645 011136 105262 002652 INCB CONFIG+2(R2) ;INCREMENT BANK ERROR COUNTER
6646 011142 SET CONFGERR0R ;PRINT CONFIG TABLE
6647 011150 000137 010456 JMP 25$ ;YES - MARK INFO IN CONFIG TABLE
6648
6649
6650
6651
6652 011154 032760 000001 002456 33$: BIT #BIT0,CSRINFO(R0) ;IS THIS MS11-M MEMORY?
6653 011162 001454 BEQ 6$ ;NO - GO TO END OF BANK
6654 011164 032760 000002 002456 BIT #BIT1,CSRINFO(R0),
6655 011172 001050 BNE 6$
6656 011174 022737 000001 002446 CMP #1,I ;IS THIS 1ST TIME THROUGH?
6657 011202 103410 BLO 18$ ;BRANCH IF NOT
6658 011204 162737 000002 002410 SUB #2,TESTADD+2 ;TRY AS 64K INTERLEAVED
6659 011212 062737 004000 172352 ADD #4000,KIPAR5 ;A1 NON-ASSERTED MEMORY
6660 011220 000137 010222 JMP 4$ ;TRY TO MATCH AGAIN
6661 011224 022737 000004 002446 18$: CMP #4,I ;4TH TIME THROUGH?
6662 011232 001404 BEQ 20$ ;YES - BRANCH
6663 011234 022737 000002 002446 CMP #2,I ;2ND TIME THROUGH
6664 011242 103405 BLO 12$ ;NO - BRANCH
6665 011244 062737 004000 172352 20$: AND #4000,KIPAR5 ;TRY AS 128K INTERLEAVED
6666 011252 000137 010222 JMP 4$ ;TRY TO MATCH AGAIN
6667 011256 022737 000003 002446 12$: CMP #3,I ;THIRD TIME THROUGH?
6668 011264 103413 BLO 6$ ;NO - BRANCH
6669 011266 062737 000002 002406 ADD #2,TESTADD ;TRY TESTING THE BANK
6670 011274 062737 000001 002410 ADD #2,TESTADD+2 ;AS A1 ASSERTED
6671 011302 162737 004000 172352 SUB #4000,KIPAR5 ;64K INTERLEAVED MEMORY
6672 011310 000137 010222 JMP 4$ ;TRY TO MATCH AGAIN
6673

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 166-4  
MATCH ALL CSR'S WITH MEMORY

6674						;*END OF BANK ROUTINE		
6675						;		
6676	011	4	104503		6\$:	CLR1CSR		;CLEAR THE CSR UNDER TEST
6677	0113		062702	000004		ADD #4,R2		;UPDATE CONFIGURATION POINTER
6678	011322		042737	001000	172350	ADD #1000,KIPAR4		;UPDATE KIPAR4 TO NEXT BANK
6679	011330		013737	172350	172352	MOV KIPAR4,KIPAR5		;AND UPDATE KIPAR5
6680	011336		000240			NOP		;DEBUG AID
6681	011340		023737	002554	172350	CMP LASTBLOCK,KIPAR4		;HAVE WE DONE THE WHOLE MEMORY SPACE?
6682	011346		10140			BLOS 19\$		;BRANCH IF DONE ;R-C
6683	011350		000137	010166		JMP 45\$		;JUMP IF NOT DONE
6684	011354		062700	000002	19\$:	ADD #2,R0		;INCREMENT CSR POINTER
6685	011360		000240			NUP		;DEBUG AID
6686	011362		104423			CACHON		;TURN ON THE CACHE
6687	011364		000137	010072		JMP 1\$		;JUMP TO TRY NEXT CSR
6688								
6689	011370		062706	000004	100\$:	ADD #4,SP		;RESTORE STACK ;R-C
6690	011374		000137	011314		JMP 6\$		;GO TO END OF BANK ROUTINE ;R-C

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 167  
MATCH ALL CSR'S WITH MEMORY

6692 011400 104423  
6693 011402 104472  
6694 011404

SUBAAS: CACHON ;MAKE SURE THE CACHE IS ON  
ECCINIT ;TRAP ON DOUBLE BIT ERRORS (NORMAL)  
NEWST <<TEST BANK 0 ACCESSES>>

\*\*\*\*\*  
: \*TEST 2 TEST BANK 0 ACCESSES  
\*\*\*\*\*

011404 000004

6695  
6696  
6697  
6698  
6699  
6700  
6701 011406 005037 002070  
6702 011412 012737 000001 002074  
6703 011420 005037 002066  
6704 011424 012737 000001 002076  
6705 011432  
6706 011440 005000  
6707 011442 012701 040000  
6708 011446 104424  
6709 011450 005720  
6710 011452 077102  
6711 011454 104423  
6712  
6713 011456 005737 002070  
6714 011462 001403  
6715 011464  
6716 011472 005737 002066  
6717 011476 001406  
6718 011500 162737 000002 002032  
6719 011506  
6720 011514 053737 002104 002650  
6721 011522  
6722  
6723 011544

TST2: SCOPE  
;THIS DOES A 'TST' INSTRUCTION ON EVERY LOCATION IN BANK #0 TO SEE  
;IF IT GETS ANY PARITY TRAPS.  
;SINCE EVERY LOCATION IS EITHER LOADED OR WRITTEN INTO BY THE PROGRAM  
;PRIOR TO THIS POINT - THEN A PARITY ERROR IMPLIES THAT THERE IS A  
;HARDWARE FAILURE IN THE MEMORY.  
;THESE ERRORS ARE COUNTED AND A FATAL ACTION IS TAKEN  
CLR PARCNT ;CLEAR PARITY ERROR COUNTER  
MOV #1,NOPAR ;SET THE NO PARITY ERROR FLAG  
CLR NEMCNT ;CLEAR NON-EXISTANT MEMORY ERROR COUNTER  
MOV #1,NONEM ;SET THE NON-EXISTANT MEMORY ERROR MODE TO COUNT  
SET4 #NONEXIST ;TRAPS TO 4 GOTO NONEXIST  
CLR R0  
MOV #SIZE,R1  
CACHOFF ;TURN CACHE OFF  
1\$: TST (R0)+ ;SEE IF I CAN DO A READ ACCESS WITHOUT A PARITY TRAP  
SOB R1,1\$  
CACHON ;TURN CACHE ON  
;SEE IF ANY FAILURES  
TST PARCNT ;ANY PARITY ERRORS?  
BEQ 2\$ ;NO - SKIP  
FATAL 3  
2\$: TST NEMCNT ;ANY NON-EXISTANT MEMORY (HOLES)?  
BEQ 3\$ ;SKIP IF EQUAL  
SUB #2,ADDRESS ;UPDATE 1ST ADDRESS FAILURE FROM AUTO INCREMENT #  
FATAL 4  
3\$: BIS CPUBIT,CONFIG ;SET CORRECT ACCESSED BIT ON BANK 0  
RES4 ;RESET TRAPS TO 4 TO DEFAULT

SUBTST <<ENABLE ECC FOR CORRECT TRAPS>>

\*\*\*\*\*  
: \*SUBTEST ENABLE ECC FOR CORRECT TRAPS  
\*\*\*\*\*

6724 011544  
6725 011562 104506  
6726 011564  
6727 011566 104472  
6728 011570

IF #SWO SET.IN @SWR OR ACTFLAG IS TRUE  
ENASBE ;TRAP ON SINGLE BIT ERRORS  
ELSE  
ECCINIT ;TRAP ON DOUBLE BIT ERRORS (NORMAL)  
END ;OF IF #SWO

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 169  
 ENABLE ECC FOR CORRECT TRAPS

6731 011570

NEWSTST &lt;&lt;TEST BANKS 1-200 (OCTAL) FOR ZEROS &amp; ONES&gt;&gt;

\*\*\*\*\*  
 :\*TEST 3 TEST BANKS 1-200 (OCTAL) FOR ZEROS & ONES  
 \*\*\*\*\*

011570 000004

```

TST3:  SCOPE
        :EACH BANK IS TESTED FOR EXISTANCE AND IF IT EXISTS
        :THEN IS IS TESTED FOR ZEROS & ONES.
        :EXCEPT -
        :       PROTECTED BANKS (WHERE THE PROGRAM IS) ARE ONLY TESTED BY
        :       'TST' INSTRUCTIONS LIKE BANK #0
        :ANY BAD BANKS ARE LOGGED IN THE CONFIGURATION TABLE.
        :THIS ROUTINE IS ONLY DOING A SMART SIZE - NOT ACTUAL TESTING!

        CLR      BANK
        MOV      #1,NOPAR           ;SET NO PARITY ERROR FLAG
        MOV      #2,NONEM          ;SET NON-EXISTANT MEMORY MODE TO EXIT TEST LOOP
        SET4     #NONEXIST         ;TRAPS TO 4 GOTO NONEXIST
        CMP      #1,PROTYP         ;IS THIS AN 11/44?
        BEQ      1$                ;BRANCH IF TRUE
        MOV      #MTST3+4,LINK1     ;SET UP LINKS
        MOV      #MTST3+6,LINK2
        BR       TAG9$

1$:      BMOV     MTST3              ;PUT IN FAST MEMORY
        MOV      #UIPAR2,LINK1     ;SET UP LINKS
        MOV      #UIPAR3,LINK2
        TAG9$:   INC      BANK
        CMP      LASTBANK,BANK     ;DONE?
        BLO      TAG2$             ;YES - SKIP TO NEXT TEST
        MOV      BANK,R1
        ASL      R1
        ASL      R1                ;BANK * 4
        MOV      R1,BANKINDEX
        CLR      PATERR            ;CLEAR PATTERN ERROR COUNTER
        CLR      PARCNT            ;CLEAR PARITY ERROR COUNTER
        CLR      NEMCNT            ;CLEAR NON-EXISTANT MEMORY COUNTER (HOLES)
        MAP      BANK              ;MAP SUPERVISOR SPACE (TEST AREA) TO BANK
        TSTB     CONFIG(R1)        ;IS THIS BANK PROTECTED?
        BMI      TSTBANK           ;YES - GO TEST BANK SPECIAL
        MOV      #207,@LINK1       ;PUT 'RETURN' INSTRUCTION AFTER WRITE ROUTINE
        MOV      #FIKST,R0
        MOV      R0,R4
        MOV      #SIZE,R1
        MOV      R1,R3
        CLR      R2
        CACHOFF
        TESTAREA
        CMP      #1,PROTYP         ;DATA IS ZEROS
        BEQ      1$                ;TURN CACHE OFF
        CALL     MTST3             ;ENTER SUPERVISOR MODE
        BR       2$                ;IS THIS AN 11/44?
        BR       2$                ;BRANCH IF TRUE

1$:      CALL     FASTCITY          ;CALL TO THE USER INSTRUCTION PAR'S
2$:      KERNEL
        CACHON
        BR       TAG3$            ;ENTER KERNEL MODE
        BR       TAG3$            ;TURN CACHE ON
        CLR      BANK              ;SKIP NEXT INSTRUCTION
        RES4
        CLR      NOPAR             ;RESET TRAPS TO 4 TO DEFAULT
        BR       SUBAAI           ;INDICATE DEFAULT PARITY ACTION
  
```

```

6732
6733
6734
6735
6736
6737
6738
6739 011572 005037 002100
6740 011576 012737 000001 002074
6741 011604 012737 000002 002076
6742 011612
6743 011620 022737 000001 003752
6744 011626 001407
6745 011630 012737 012426 002516
6746 011636 012737 012430 002520
6747 011644 000411
6748 011646
6749 011654 012737 177644 002516
6750 011662 012737 177646 002520
6751 011670 005237 002100
6752 011674 023737 002552 002100
6753 011702 103457
6754 011704 013701 002100
6755 011710 006301
6756 011712 006301
6757 011714 010137 002102
6758 011720 005037 002072
6759 011724 005037 002070
6760 011730 005037 002066
6761 011734
6762 011750 105761 002650
6763 011754 100555
6764 011756 012777 000207 170532 WARN1:
6765 011764 012700 060000
6766 011770 010004
6767 011772 012701 040000
6768 011776 010103
6769 012000 005002
6770 012002 104424
6771 012004
6772 012012 022737 000001 003752
6773 012020 001403
6774 012022 004737 012422
6775 012026 000402
6776 012030 004737 177640
6777 012034 104417
6778 012036 104423
6779 012040 000416
6780 012042 005037 002100
6781 012046
6782 012070 005037 002074
6783 012074 000564
  
```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 169-1  
T3 TEST BANKS 1-200 (OCTAL) FOR ZEROS & ONES

6784	012076	005737	002066		TAG3\$:	TST	NEMCNT		;ANY TRAPS?
6785	012102	001401				BEQ	1\$		;NO - SKIP
6786	012104	000671				BR	TAG9\$		;NOW - TRY NEXT BANK
6787	012106	104424			1\$:	CACHOFF			;TURN CACHE OFF
6788	012110					TESTAREA			;ENTER SUPERVISOR MODE
6789	012116	004777	170376			CALL	@LINK2		;FINISH PATTERN
6790	012122	104417				KERNEL			;ENTER KERNEL MODE
6791	012124	104423				CACHON			;TURN CACHE ON
6792	012126	005737	002072			TST	PATERR		;ANY PATTERN ERRORS
6793	012132	001040				BNE	2\$		;YES - SKIP
6794	012134	005737	002070			TST	PARCNT		;ANY PARITY ERRORS
6795	012140	001035				BNE	2\$		;YES - SKIP
6796	012142	005737	002066			TST	NEMCNT		;ANY NON EXISTANT MEMORY
6797	012146	001032				BNE	2\$		;YES - SKIP
6798	012150	012700	060000			MOV	#FIRST,R0		
6799	012154	010004				MOV	R0,R4		
6800	012156	012701	040000			MOV	#SIZE,R1		
6801	012162	010103				MOV	R1,R3		
6802	012164	013702	002600			MOV	ONES,R2		;DATA IS ONES
6803	012170	012777	000240	170320		MOV	#000240,@LINK1		;PUT 'NOP' INSTRUCTION BACK IN SUBROUTINE
6804	012176	104424				CACHOFF			;TURN CACHE OFF
6805	012200					TESTAREA			;ENTER TEST MODE
6806	012206	022737	000001	003752		CMP	#1,PROTYP		;IS THIS AN 11/44?
6807	012214	001403				BEQ	5\$		;BRANCH IF IT IS
6808	012216	004737	012422			CALL	MTST3		;DO IN MEMORY IF NOT
6809	012222	000402				BR	6\$		;JUMP OVER NEXT INSTRUCTION
6810	012224	004737	177640		5\$:	CALL	FASTCITY		;CALL TO THE USER INSTRUCTION PAR'S
6811	012230	104417			6\$:	KERNEL			;ENTER KERNEL MODE
6812	012232	104423				CACHON			;TURN CACHE ON
6813	012234	013700	002102		2\$:	MOV	BANKINDEX,R0		
6814	012240	005737	002072			TST	PATERR		;ANY PATTERN ERRORS?
6815	012244	001006				BNE	3\$		;YES - SKIP
6816	012246	005737	002070			TST	PARCNT		;ANY PARITY ERRORS?
6817	012252	001003				BNE	3\$		;YES - SKIP
6818	012254	005737	002066			TST	NEMCNT		;ANY HOLES?
6819	012260	001406				BEQ	4\$		;NONE - SKIP
6820	012262	052760	000001	002650	3\$:	BIS	#BIT0,CONFIG(R0)		;SET ERROR BIT IN THIS BANK
6821	012270					SET	CONFERROR		;FORCE PRINTING OF CONFIGURATION TABLE
6822	012276	053760	002104	002650	4\$:	BIS	CPUBIT,CONFIG(R0)		;SET ACCESSED BIT
6823	012304	000137	011670			JMP	TAG9\$		
6824									
6825									
6826	012310								
6827	012312	012737	000001	002076	TSTBANK:	PUSH	R1		;SET NON-EXISTANT MEMORY TO COUNT
6828	012320	012700	060000			MOV	#1,NONEM		
6829	012324	012701	020000			MOV	#FIRST,R0		
6830	012330	104424				MOV	#20000,R1		
6831	012332					CACHOFF			;TURN CACHE OFF
6832	012340	005720				TESTAREA			;ENTER TEST MODE
6833	012342	077102			4\$:	TST	(R0)+		
6834	012344	104417				SUB	R1,4\$		
6835	012346	104423				KERNEL			;ENTER KERNEL MODE
6836	012350	012737	000002	002076		CACHON			;TURN CACHE ON
6837	012356					MOV	#2,NONEM		;RESET NON-EXISTANT MEMORY TO EXIT TEST LOOP
6838	012360					POP	R1		
6839	012366	052761	000001	002650		IF PARCNT NE #0			
6840	012374					BIS	#BIT0,CONFIG(R1)		;ERROR BANK
						SET	CONFERROR		

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR J2 09:41 PAGE 169-2  
 T3 TEST BANKS 1-200 (OCTAL) FOR ZEROS & ONES

6841	012402				END ;OF IF PARCNT	
6842	012402				IF NEMCNT EQ #0	
6843	012410	053761	002104	002650	BIS CPUBIT,CONFIG(R1)	;ACCESSED BANK
6844	012416				END ;OF IF NEMCNT	
6845	012416	000137	011670		JMP TAG9\$	
6846	012422	010220			MOV R2,(R0)+	:V177640
6847	012424	077102			SOB R1,MTST3	:V177642
6848	012426	000240			NOP	:V177644
6849	012430	012401			MOV (R4)+,R1	:V177646
6850	012432	020102			CMP R1,R2	:V177650
6851	012434	001402			BEQ 3\$	:V177652
6852	012436	005237	002072		INC PATERR	:V177654
6853	012442	077306			SOB R3,2\$	:V177660
6854	012444	000207			RETURN	:V177662

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 170  
 T3 TEST BANKS 1-200 (OCTAL) FOR ZEROS & ONES

6856 012446

SUBAAI: SUBST &lt;&lt;FIND SHADOW INHIBIT MODE POINTERS&gt;&gt;

;\*\*\*\*\*

;\*SUBTEST FIND SHADOW INHIBIT MODE POINTERS

;\*\*\*\*\*

;\* THIS SECTION LOOKS FOR INTERLEAVED MS11-M MEMORIES AND FIGURES OUT

;\* WHERE THE SHADOW INHIBIT MODE POINTERS ARE LOCATED. THESE AREAS

;\* ARE THEN MARKED AS PROGRAM SPACE.

6857

6858

6859

6860 012446 005037 002100

6861 012452 004737 047020

6862 012456 013700 002102

6863 012462

6864 012476

6865 012504 062700 000020

6866 012510 062737 000010 002100

6867 012516

6868 012520 062702 000040

6869 012524 062737 000020 002100

6870 012532

6871 012532 052760 000200 002650

6872 012540

6873 012542 005237 002100

6874 012546

6875 012546 023737 002552 002100

6876 012554 002336

SHADL1: CLR BANK ;RESET BANK TO ZERO

CALL EXBANK ;SET BANK PARAMETERS

MOV BANKINDEX,R0

IF ACFLAG IS TRUE AND INTFLAG IS TRUE

IF INT64K IS TRUE

ADD #20,R0 ;POINT TO BANKINDEX + 4

ADD #10,BANK ;POINT TO BANK + 8

ELSE

ADD #40,R2 ;POINT TO BANKINDEX + 8

ADD #20,BANK ;POINT TO BANK + 16

END; OF IF INT64K

BIS #BIT7,CONFIG(R0) ;MAKE NEW BANK PROGRAM SPACE

ELSE

INC BANK ;GO TO NEXT BANK

END; OF IF ACFLAG

CMP LASTBANK,BANK ;HAVE WE DONE ALL THE BANKS?

BGE SHADL1 ;BRANCH IF NOT

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 172  
 FIND SHADOW INHIBIT MODE POINTERS

6879 012556

012556 000004

6880  
 6881  
 6882  
 6883  
 6884  
 6885  
 6886  
 6887  
 6888  
 6889  
 6890  
 6891  
 6892  
 6893  
 6394  
 6895  
 6896  
 6897 012560 104424  
 6898 012562 012737 177777 002154  
 6899 012570  
 6900 012574 012701 060000  
 6901 012600 004737 047020  
 6902 012604 013700 002102  
 6903 012610  
 6904 012616  
 6905 012624  
 6906 012632  
 6907 012640 012703 040000  
 6908 012644 012737 000001 002236  
 6909 012652  
 6910 012654 012703 000002  
 6911 012660  
 6912 012660 116002 002651  
 6913 012664 006302  
 6914 012666 042702 177741  
 6915 012672 010237 002150  
 6916 012676  
 6917 012706 013737 002150 002154  
 6918 012714  
 6919 012722 052760 000100 002650  
 6920 012730  
 6921 012730 004737 013064

```

NEWST <<ECC INHIBIT MODE POINTER TEST>>
*****
*TEST 4      ECC INHIBIT MODE POINTER TEST
*****
TST4:  SCOPE
;THE MS11-M OR MF11S-K INHIBIT ECC DISABLE AND DIAGNOSTIC CHECK MODE
;ON THE BOTTOM FIRST OR SECOND 16K WORDS CONTROLLED BY A CSR. THIS
;IS CONSIDERED TO BE A PROTECTED BANK BY THE PROGRAM. IT MAY BE
;QUITE COMPLEX TO DETERMINE ON A GIVEN SYSTEM CONFIGURATION WHICH
;BANKS CAN BE PROTECTED;
;SO
;THIS ROUTINE ATTEMPS TO CREATE A DOUBLE BIT ERROR IN ADDRESS 0 & 2
;OF EVERY ECC BANK. ECC HARDWARE WILL PREVENT THIS FROM HAPPENING
;IN PROTECTED BANKS WHICH SHOULD ALWAYS INCLUDE BANK ZERO - WHERE
;THE PROGRAM IS.

;WARNING:!!!!!!!!!!
;IN CASE OF HARDWARE FAILURE IT IS COMMON THAT A DOUBLE BIT ERROR
;WILL BE CREATED ON THE KERNEL STACK & 'CRASH' THE DIAGNOSTIC
;DURING THIS ROUTINE. YOUR ONLY CLUE IS THAT YOU CAN GET AS FAR AS
;THIS ROUTINE BUT NOT PAST IT!
CACHOFF                                ;TURN CACHE OFF
MOV  #-1,OLDCSR
FOR BANK := #0 TO LASTBANK
  MOV  #FIRST,R1                                ;SET UP VIRT ADDR POINTER
  CALL EXBANK
  MOV  BANKINDEX,R0
  IF ACFLAG IS TRUE
    IF MKFLAG IS TRUE
      IF SKIPMK IS FALSE
        IF INTFLAG IS TRUE
          MOV  #40000,R3                                ;SET INDEX COUNTER
          MOV  #1,SPLTCSR                                ;MAP AS INTERLEAVED BANK
        ELSE
          MOV  #2,R3                                ;SET INDEX COUNTER
        END; OF IF INTFLAG
        MOVB  CONFIG+1(R0),R2
        ASL   R2
        BIC   #^C36,R2
        MOV   R2,CSRNO
        IF CSRNO NE OLDCSR
          MOV  CSRNO,OLDCSR
          IF PFLAG IS FALSE
            BIS #BIT6,CONFIG(R0)
          END; OF IF PFLAG
          CALL IMPTEST

```



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 6-APR-82 09:41 PAGE 173  
 T4 ECC INHIBIT MODE POINTER TEST

6923	012734			IF INTFLAG IS TRUE	
6924	012742	116002	002651	MOVB CONFIG+1(R0),R2	
6925	012746	072227	177775	ASH #-3,R2	
6926	012752	042702	177741	BIC #^C36,R2	
6927	012756	010237	002150	MOV R2,CSRNO	
6928	012762	062701	000002	ADD #2,R1	;FIX POINTER FOR A1 ASSERTED HALF
6929	012766	004737	013064	CALL IMPTEST	
6930	012772	005037	002236	CLR SPLTCSR	
6931	012776			END; OF IF INTFLAG	
6932	012776			END; OF IF CSRNO	
6933	012776			END; OF IF SKIPMK	
6934	012776			END; OF IF MKFLAG	
6935	012776			END; OF IF ACFLAG	
6936	012776			END; OF FOR BANK	
6937	013012			MAP	;MAP TEST SPACE TO BANK 0
6938	013026	005037	002100	CLR BANK	
6939	013032			IF #SWO SET.IN @SWR OR ACTFLAG IS TRUE	
6940	013050	104506		ENASBE	;TRAP ON SINGLE BIT ERRORS
6941	013052			ELSE	
6942	013054	104472		ECCINIT	;TRAP ON DOUBLE BIT ERRORS (NORMAL)
6943	013056			END; OF IF #SWO	
6944	013056	104423		CACHON	;TURN THE CACHE BACK ON
6945	013060	000137	013372	JMP SUBAAR	;JUMP OVER THE SUBROUTINE

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 174  
T4 ECC INHIBIT MODE POINTER TEST

6947	013064	005004			IMPTTEST:CLR	R4	
6948	013066				MAP BANK		;MAP SUPERVISOR SPACE (TEST AREA) TO BANK
6949	013102	005005			CLR	R5	
6950	013104	012737	020000	002146	MOV	#BIT13,CSR	
6951	013112				TESTAREA		;ENTER TEST MODE
6952	013120				PUSH	(R1)	;SAVE TEST LOCATION
6953	013122	060301			ADD	R3,R1	;INDEX TO NEXT LOCATION
6954	013124				PUSH	(R1)	;SAVE TEST LOCATION
6955	013126	104505			CHK1DIS		;DISABLE ECC & WRITE CHECKBITS FOR 1 CSR
6956	013130	010411			MOV	R4,(R1)	;WRITE CHECKBITS (ALL ZEROS)
6957	013132	160301			SUB	R3,R1	
6958	013134	010411			MOV	R4,(R1)	
6959	013136	104503			CLR1CSR		;CLEAR CSR
6960	013140	005711			TST	(R1)	;READ CHECKBITS INTO REAL CSR
6961	013142	104501			WAS1DBE		;WAS THERE A DOUBLE BIT ERROR
6962							
6963							
6964							
6965	013144				ON.NOERROR :1		
6966	013146	012737	020000	002146	MOV	#BIT13,CSR	
6967	013154	104505			CHK1DIS		;DISABLE ECC & WRITE CHECKBITS FOR 1 CSR
6968	013156	013711	002600		MOV	ONES,(R1)	
6969	013162	060301			ADD	R3,R1	
6970	013164	013711	002600		MOV	ONES,(R1)	
6971	013170	160301			SUB	R3,R1	
6972	013172	104503			CLR1CSR		;CLEAR CSR
6973	013174	005711			TST	(R1)	
6974	013176	104501			WAS1DBE		;WAS THERE A DOUBLE BIT ERROR
6975	013200				ON.NOERROR :2		
6976	013202				IF #BIT9 SET.IN CONFIG+2(R0)		;IS THIS A MS11-P
6977	013212	104513			CBREG		;ENABLE CHECK/SYNDROME BIT REGISTER
6978	013214	012737	023140	002146	MOV	#23140,CSR	;WRITE DBE'S IN CSR
6979	013222				ELSE		;OR A MS11-M
6980	013224	012737	027400	002146	MOV	#27400,CSR	;WRITE DBE'S IN CSR
6981	013232				END		
6982	013232	104505			CHK1DIS		;DISABLE ECC & WRITE CHECKBITS FOR 1 CSR
6983	013234	010411			MOV	R4,(R1)	
6984	013236	060301			ADD	R3,R1	;ADD INDEX TO GET TO SECOND WORD
6985	013240	010411			MOV	R4,(R1)	
6986	013242	160301			SUB	R3,R1	;SUBTRACT INDEX TO FIRST WORD
6987	013244	104503			CLR1CSR		;CLEAR CSR
6988	013246	005711			TST	(R1)	
6989	013250	104501			WAS1DBE		;WAS THERE A DOUBLE BIT ERROR
6990	013252				ON.NOERROR :3		
6991	013254				IF #BIT9 SET.IN CONFIG+2(R0)		;IS THIS A MS11-P
6992	013264	104513			CBREG		;ENABLE CHECK/SYNDROME BIT REGISTER
6993	013266	012737	023604	002146	MOV	#23604,CSR	;WRITE DBE'S IN CSR
6994	013274				ELSE		;IS IT A MS11-P
6995	013276	012737	074000	002146	MOV	#74000,CSR	;WRITE DBE'S IN CSR
6996	013304				END		
6997	013304	104505			CHK1DIS		;DISABLE ECC & WRITE CHECKBITS FOR 1 CSR
6998	013306	010411			MOV	R4,(R1)	
6999	013310	060301			ADD	R3,R1	;INDEX TO SECOND WORD
7000	013312	010411			MOV	R4,(R1)	
7001	013314	104503			CLR1CSR		;CLEAR CSR
7002	013316	160301			SUB	R3,R1	;GO BACK TO FIRST WORD
7003	013320	005711			TST	(R1)	

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 174-1  
T4 ECC INHIBIT MODE POINTER TEST

7004 013322	104501		WAS1DBE		;WAS THERE A DOUBLE BIT ERROR
7005 013324			END ;OF ON.NOERROR ;3		
7006 013324			END ;OF ON.NOERROR ;2		
7007 013324			END ;OF ON.NOERROR ;1		
7008 013324			ON.ERROR		
7009 013326	005205		INC R5		;IDENTIFY AS BAD BANK
7010 013330			END ;OF ON.ERROR		
7011 013330	104471		ECC1DIS		;DISABLE ERROR CORRECTION
7012 013332	010411		MOV R4,(R1)		;CLEAR OUT DOUBLE BIT ERROR!
7013 013334	060301		ADD R3,R1		;INDEX TO SECOND WORD
7014 013336	010411		MOV R4,(R1)		;CLEAR OUT DOUBLE BIT ERROR!
7015 013340	104503		CLR1CSR		
7016 013342	005705		TST R5		
7017 013344	001405		BEQ 1\$		
7018 013346	050560	002650	BIS R5,CONFIG(R0)		
7019 013352	105260	002652	INCB CONFIG+2(R0)		
7020 013356	104036		ERROR +36		
7021 013360		1\$:	POP (R1)		;RESTORE TEST LOCATION (2ND WORD)
7022 013362	160301		SUB R3,R1		;GO BACK TO FIRST WORD
7023 013364			POP (R1)		;RESTORE TEST LOCATION (1ST WORD)
7024 013366	104417		KERNEL		
7025 013370	000207		RETURN		
7026					
7027 013372			SUBAAR: SET	STOPOK	;PROGRAM CAN NOW BE HALTED

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 182  
T4 ECC INHIBIT MODE POINTER TEST

7036 013400

7037 013400 012700 000020  
7038 013404 012701 002456  
7039 013410 005021  
7040 013412 077002  
7041 013414  
7042 013420 004737 047020  
7043 013424 013700 002102  
7044  
7045 013430  
7046 013436 116003 002651  
7047 013442 042703 177760  
7048 013446 006303  
7049 013450 005263 002456  
7050 013454  
7051  
7052 013462  
7053 013462  
7054 013470 116003 002651  
7055 013474 010304  
7056 013476 042703 177760  
7057 013502 072427 177774  
7058 013506 042704 177760  
7059 013512  
7060 013516 042760 014000 002652  
7061 013524 042760 170000 002650  
7062 013532  
7063 013534  
7064 013534  
7065 013536  
7066 013540  
7067 013540  
7068 013546  
7069 013546  
7070 013546  
7071 013546  
7072 013562  
7073 013566 005000  
7074 013570 005001  
7075 013572 005005  
7076 013574 005037 014002  
7077 013600 022761 000040 002456 2\$:  
7078 013606 002043  
7079 013610 022761 000010 002456  
7080 013616 002003  
7081 013620 004737 014124  
7082 013624 000434  
7083 013626 0160C5 002650 3\$:  
7084 013632 032705 000002  
7085 013636 001415  
7086 013640 042705 170377  
7087 013644 072527 177771  
7088 013650 020501  
7089 013652 001007

```

SUBTST <<LEGAL CONFIGURATION CHECK>>
*****
*SUBTEST      LEGAL CONFIGURATION CHECK
*****
1$:  MOV      #16,R0
    MOV      #CSRINFO,R1
    CLR      (R1)+
    SOB      R0,1$
        FOR BANK := #0 TO LASTBANK
            CALL      EXBANK
            MOV BANKINDEX,R0

            IF ACFLAG IS TRUE
                MOVB    CONFIG+1(R0),R3
                BIC     #^C17,R3
                ASL     R3
                INC     CSRINFO(R3)
                IF MKFLAG IS TRUE
                    ;MAKE SURE THAT EACH BANK HAS NO MORE THAN 2 CSRS
                    BEGIN LEGALCSR
                        IF INTFLAG IS TRUE
                            MOVB    CONFIG+1(R0),R3
                            MOV R3,R4
                            BIC     #^C17,R3
                            ASH     #-4,R4
                            BIC     #^C17,R4
                            IF R3 EQ R4
                                BIC     #BIT11!BIT12,CONFIG+2(R0)
                                BIC     #170000,CONFIG(R0)
                                LEAVE LEGALCSR
                            END; OF IF R3
                        ELSE
                            LEAVE LEGALCSR
                        END; OF IF INTFLAG
                    SET     CONFGERROR
                    END     LEGALCSR
                END; OF IF MKFLAG
            END; OF IF ACFLAG
        END; OF FOR BANK

    PUSH     R5,R0
    CLR      R0
    CLR      R1
    CLR      R5
    CLR      MBERR
    CMP      #40,CSRINFO(R1)
    BGE      5$
    CMP      #10,CSRINFO(R1)
    BGE      3$
    CALL     :LLCSR
    BR       5$
    MOV      CONFIG(R0),R5
    BIT      #BIT1,R5
    BEQ      4$
    BIC      #^C7400,R5
    ASH      #-7,R5
    CMP      R5,R1
    BNE      4$

;SAVE CONTENTS OF R5, R0
;CLEAR REGISTERS

;CLEAR ERROR INDICATOR
;IS CURRENT CSR <= 40
;BRANCH IF SO
;IS CURRENT CSR < 10
;BRANCH IF SO
;CALL ERROR ROUTINE
;TRY NEXT CSR
;MOVE LOW WORD TO R5
;DOES MEMORY EXIST HERE?
;BRANCH IF NOT
;ISOLATE CSR NUMBER IN
;REGISTER 5
;IS IT THE CURRENT CSR?
;TRY NEXT WORD OF CONFIG IF NOT

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 182-1  
LEGAL CONFIGURATION CHECK

7090	013654	032760	010000	002652	BIT	#BIT12,CONFIG+2(R0)	:IS IT INTERLEAVED?	
7091	013662	001003			BNE	4\$	:BRANCH IF '0	
7092	013664	012737	000001	014002	MOV	#1,MBERR	:SET ERROR INDICATOR	
7093	013672	062700	000004		4\$: ADD	#4,R0	:UPDATE CONFIG COUNTER	
7094	013676	022700	000340		CMP	#340,R0	:CONFIG TABLE ALL DONE?	
7095	013702	001351			BNE	3\$	:BRANCH IF NOT	
7096	013704	005737	014002		TST	MBERR	:ERRORS FOUND?	
7097	013710	001402			BEQ	5\$	:TRY NEXT CSR IF NOT	
7098	013712	004737	014124		CALL	ILLCSR	:CALL ERROR ROUTINE	
7099	013716	005000			5\$: CLR	R0	:REINITIALIZE CONFIG COUNTER	
7100	013720	005037	014002		CLR	MBERR	:CLEAR ERROR INDICATOR	
7101	013724	062701	000002		ADD	#2,R1	:UPDATE CSR COUNTER	
7102	013730	022701	000040		CMP	#40,R1	:ALL CSR'S DONE?	
7103	013734	001321			BNE	2\$	:BRANCH IF NOT	
7104	013736				POP	R0,R5	:RESTORE REGISTERS	
7105	013742	005037	014002		CLR	MBERR	:RESET ERROR INDICATOR	
7106	013746	012700	000734		MOV	#734,R0	:INDEX TO TOP OF CONFIG TABLE	:R-C
7107	013752	032760	000002	002650	6\$: BIT	#BIT1,CONFIG(R0)	:MEMORY PRESENT?	:R-C
7108	013760	001003			BNE	7\$	:BRANCH IF SO	:R-C
7109	013762	162700	000004		SUB	#4,R0	:TRY NEXT LOWER ENTRY IN CONFIG TABLE	:R-C
7110	013766	000771			BR	6\$		:R-C
7111	013770	006200			7\$: ASR	R0		:R-C
7112	013772	006200			ASR	R0	:DIVIDE INDEX BY 4 TO GET BANK #	:R-C
7113	013774	010037	002552		MOV	R0,LASTBANK	:STORE IN LASTBANK	:R-C
7114	014000	000402			BR	SKUJ		
7115	014002	000000			MBERR: .WORD 0		:SAVE SPACE FOR ERROR INDICATOR	
7116	014004	000000			PHEBE: .WORD 0		:SAVE SPACE FOR ODD BOUNDARY INTERLEAVED INDICATOR	
7117	014006	005000			SKUJ: CLR	R0	:CLEAR CONFIG COUNTER	
7118	014010	005037	014004		CLR	PHEBE	:CLEAR COUNTER	
7119	014014	032760	000002	002650	1\$: BIT	#BIT1,CONFIG(R0)	:IS THERE MEMORY PRESENT?	
7120	014022	001431			BEQ	3\$	:BRANCH IF NOT	
7121	014024	032760	010000	002652	BIT	#BIT12,CONFIG+2(R0)	:IS IT INTERLEAVED?	
7122	014032	001005			BNE	2\$	:BRANCH IF SO	
7123	014034	005237	014004		INC	PHEBE	:INCREMENT COUNTER	
7124	014040	062700	000004		ADD	#4,R0	:INCREMENT CONFIG COUNTER	
7125	014044	000763			BR	1\$	:TRY NEXT BANK	
7126	014046	023727	014004	000010	2\$: CMP	PHEBE,#10	:IS THE COUNTER EQUAL TO...	
7127	014054	001417			BEQ	4\$	:ONE OF THE SPECIAL VALUES.	
7128	014056	023727	014004	000030	CMP	PHEBE,#30	:IF IT IS...	
7129	014064	001413			BEQ	4\$	:BRANCH TO 4\$	
7130	014066	023727	014004	000050	CMP	PHEBE,#50		
7131	014074	001407			BEQ	4\$		
7132	014076	023727	014004	000070	CMP	PHEBE,#70		
7133	014104	001403			BEQ	4\$		
7134	014106	005037	014004		3\$: CLR	PHEBE	:CLEAR INDICATOR	
7135	014112	000403			BR	5\$		
7136	014114	012737	000001	014004	4\$: MOV	#1,PHEBE	:SET INDICATOR	
7137	014122	000421			5\$: BR	SUBAAP	:BRANCH TO NEXT SUBTEST	
7138	014124	010102			ILLCSR: MOV	R1,R2	:R2 HAS CSR NUMBER	
7139	014126	006202			ASR	R2	:MAKE ACCEPTABLE FOR PRINTING	
7140	014130	022702	000012		CMP	#10.,R2		
7141	014134	100002			BPL	1\$		
7142	014136	062702	000007		ADD	#7,R2		
7143	014142	062702	000060		1\$: ADD	#60,R2		
7144	014146	110237	100014		MOVH	R2,MSG122	:PUT NUMBER INTO ERROR MESSAGE	
7145	014152				TYPE	MSG122		
7146	014156				SET	CONFERROR		

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 182-2  
LEGAL CONFIGURATION CHECK

7147 014164 000207

RETURN

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 184  
LEGAL CONFIGURATION CHECK

7150 014166

SUBAAP: SUBTST &lt;&lt;PRINT CONFIGURATION DETAILS&gt;&gt;

\*\*\*\*\*  
:SUBTEST PRINT CONFIGURATION DETAILS  
\*\*\*\*\*

7151 014166  
7152 014202 013702 002552  
7153 014206 006302  
7154 014210 006302

CLEAR LSIZE,MSIZE,PSIZE  
MOV LASTBANK,R2  
ASL R2  
ASL R2  
FOR R1 := #0 TO R2 BY #4  
IF CPUBIT SET.IN CONFIG(R1)  
IF #BIT8 SET.IN CONFIG+2(R1)  
IF #BIT9 SET.IN CONFIG+2(R1)  
LET PSIZE := PSIZE + #1  
ELSE  
LET MSIZE := MSIZE + #1  
END;IF BIT9  
ELSE  
LET LSIZE := LSIZE + #1  
END;IF BIT8  
END; OF IF CPUBIT  
END ;OF FOR ALL BANKS IN TABLE

7168  
7169 014274 005037 002446  
7170 014300  
7171 014302 006361 002370  
7172 014306 006361 002370  
7173 014312 006361 002370  
7174 014316 006361 002370  
7175 014322 066137 002370 002446

CLR I  
FOR R1 := #0 TO #10 BY #2  
ASL BSIZE(R1)  
ASL BSIZE(R1)  
ASL BSIZE(R1)  
ASL BSIZE(R1) ;BSIZE(R1) := BSIZE(R1) \* 16.  
ADD BSIZE(R1),I ;I <- I + BSIZE(R1)  
END; FOR R1

7176 014330  
7177 014342  
7178 014344  
7179 014354  
7180 014360  
7181 014360  
7182 014372 006337 002412  
7183 014376 006337 002412  
7184 014402 006337 002412  
7185 014406 006337 002412  
7186 014412

FOR R1 := #0 TO #200 BY #4  
IF CPUBIT SET.IN CONFIG(R1)  
LET UNITOP := UNITOP + #1  
END; OF IF CPUBIT  
END; OF FOR R1  
ASL UNITOP  
ASL UNITOP  
ASL UNITOP  
ASL UNITOP ;UNITOP := UNITOP \* 16.  
IF I LT UNITOP THEN LET I := UNITOP

7187 014430  
7188 014434 005737 002374  
7189 014440 001405

2\$:

TYPE \$CRLF  
TST LSIZE  
BEQ 3\$  
TYPDEC LSIZE  
TYPE MSG112

7190 014442  
7191 014450  
7192 014454 005737 002376  
7193 014460 001405

3\$:

TST MSIZE  
BEQ 4\$  
TYPDEC MSIZE  
TYPE MSG113

7194 014462  
7195 014470  
7196 014474 005737 002400  
7197 014500 001405

4\$:

TST PSIZE  
BEQ 5\$  
TYPDEC PSIZE  
TYPE MSG114

7198 014502  
7199 014510  
7200 014514  
7201 014522

5\$:

TYPDEC I  
TYPE MSG070  
IF #SW6 OFF.IN @SWR  
CALL PCONFIG

7202 014526  
7203 014536 004737 041352

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 184-1  
PRINT CONFIGURATION DETAILS

7204 014542

END; OF IF #SW6



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 186  
PRINT CONFIGURATION DETAILS

7207 014542

7208 014542  
7209 014556 005037 002430  
7210 014562 012700 065670  
7211 014566  
7212 014570  
7213 014576 111001  
7214 014600 042701 177400  
7215 014604  
7216 014612 000261  
7217 014614  
7218 014616 000241  
7219 014620  
7220 014620 006101  
7221 014622 005201  
7222 014624 006301  
7223 014626 006301  
7224 014630 006301  
7225 014632 006301  
7226 014634 163701 002430  
7227 014640 010137 002430  
7228 014644  
7229 014654 060137 002416  
7230 014660  
7231 014660  
7232 014670 060137 002420  
7233 014674  
7234 014674 062700 000004  
7235 014700  
7236 014700  
7237 014710  
7238 014740 104046  
7239 014742  
7240 014742

SUBTST &lt;&lt;CHECK APT SIZING&gt;&gt;

\*\*\*\*\*  
:SUBTEST CHECK APT SIZING  
\*\*\*\*\*

IF APTFLAG IS TRUE AND APTSIZE IS TRUE

CLR TEMP  
MOV #SMAMS1,R0  
FOR R2 := #0 TO #4  
IFB 1(R0) NE #0  
MOVB (R0),R1  
BIC #177400,R1  
IF 2(R0) LT #0  
SEC  
ELSE  
CLC  
END ;OF IF 2(R0)  
ROL R1  
INC R1  
ASL R1  
ASL R1  
ASL R1  
ASL R1  
SUB TEMP,R1  
MOV R1,TEMP  
IFB 1(R0) EQ #3  
ADD R1,APTPAR  
END ;OF IFB 1(R0)  
IFB 1(R0) EQ #4  
ADD R1,APTECC  
END ;OF IFB 1(R0)  
ADD #4,R0  
END ;OF IFB 1(R0)  
END ;OF FOR R2  
IF APTPAR NE LSIZE OR APTECC NE MSIZE OR APTECC NE PSIZE  
ERROR +46  
END ;OF IF APTPAR  
END ;OF IF APTFLAG

;TO COMPENSATE FOR 4 BANKS BEING (0-3)

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 187  
CHECK APT SIZING

7242 014742

7243 014742 000004  
7244 014744 005037 002220  
7245 014750 017700 165646  
7246 014754 042700 177761  
7247 014760 004770 014770  
7248 014764 000137 015010  
7249 014770 015452  
7250 014772 015560  
7251 014774 015666  
7252 014776 016016  
7253 015000 016146  
7254 015002 016276  
7255 015004 016450  
7256 015006 016600  
7257 015010 004737 015352  
7258  
7259 015014

015014 000004  
7260  
7261  
7262 015016  
7263 015024  
7264 015040 004737 024102  
7265 015044  
7266 015052 005037 002106  
7267 015056  
7268 015056 004737 015352  
7272  
7273 015062

7274 015062 004737 024566

LOOP: NEWTST <<DIAGNOSTIC MODE DISPATCH ROUTINE>>

\*\*\*\*\*  
: \*TEST 5 DIAGNOSTIC MODE DISPATCH ROUTINE  
\*\*\*\*\*

TST5: SCOPE  
CLR CONTFLAG  
MOV @SWR,R0 ;GET SWITCHES  
BIC #\*C16,R0 ;MASK TO ONLY MODE BITS  
CALL @DISPTBL(R0) ;DISPATCH TO ROUTINE THROUGH NEXT TABLE  
JMP MEMDONE ;GO TO NEXT TEST  
DISPTBL:BA\*PAF ;MODE 0:BANKS FORWARD, PATTERNS FORWARD  
BA\*PAR ;MODE 1:BANKS FORWARD, PATTERNS REVERSE  
BAWPAF ;MODE 2:BANKS WORST FIRST, PATTERNS FORWARD  
BAWPAR ;MODE 3:BANKS WORST FIRST, PATTERNS REVERSE  
PAFBAF ;MODE 4:PATTERNS FORWARD, BANKS FORWARD  
PAFBAW ;MODE 5:PATTERNS FORWARD, BANKS WORST FIRST  
PARBAF ;MODE 6:PATTERNS REVERSE, BANKS FORWARD  
PARBAW ;MODE 7:PATTERNS REVERSE, BANKS WORST FIRST

MEMDONE:CALL DOBACK ;CHECK BACKGROUND PATTERN

NEWTST<<UNIQUE BANK TEST>>

\*\*\*\*\*  
: \*TEST 6 UNIQUE BANK TEST  
\*\*\*\*\*

TST6: SCOPE  
;MAKE SURE THAT EACH BANK CAN HAVE UNIQUE DATA  
;WRITE AND READ THE BANK NUMBER IN EACH BANK (EXCEPT WHERE THE PROGRAM IS)  
IF SELONLY IS FALSE  
SET HEADER,MUT  
CALL MT0027  
SET HEADER  
CLR MUT  
END ;OF IF SELONLY  
CALL DOBACK ;RESTORE BACKGROUND PATTERN

FLUSH: SUBTST <<FLUSH OUT DBE'S>>

\*\*\*\*\*  
: \*SUBTEST FLUSH OUT DBE'S  
\*\*\*\*\*

CALL MT0030

C7MSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 189  
END OF PASS ROUTINE

7277					.SBTTL END OF PASS ROUTINE
7278					*****
7279					;*INCREMENT THE PASS NUMBER (\$PASS)
7280					;*INDICATE END-OF-PROGRAM AFTER EACH PASSES THRU THE PROGRAM
7281					;*TYPE 'END PASS #XXXXX' (WHERE XXXXX IS A DECIMAL NUMBER)
7282					;*IF THERES A MONITOR GO TO IT
7283					;*IF THERE ISN'T JUMP TO LOOP
7284	015066	005037	002436		\$EOP: CLR FSINFLAG
7285	015072	012700	002652		MOV #CONFIG+2,R0 ;MOVE 2ND WORD OF CONFIG TO R0
7286	015076	042710	020000		1\$: BIC #BIT13,(R0) ;CLEAR BACKGROUND VALID BIT
7287	015102	062700	000004		ADD #4,R0 ;INCREMENT TO NEXT BANK
7288	015106	020027	003620		CMP R0,#3620 ;DONE?
7289	015112	003771			BLE 1\$ ;NO - BRANCH
7290	015114	013737	002614	002014	MOV \$ERTTL,LASTERROR
7291	015122	005237	065646		INC \$PASS ;:INCREMENT THE PASS NUMBER
7292	015126	042737	100000	065646	BIC #100000,\$PASS ;:DON'T ALLOW A NEG. NUMBER
7293	015134				TYPE MSG077 ;:TYPE 'END PASS #'
7294	015140				IF #SW11 SET IN @SWR OR QVFLAG IS TRUE OR \$PASS EQ #1
7295	015166				TYPE MSG035 ;:QV
7296	015172	005037	002342		CLR QVFLAG
7297	015176				END OF IF SW11
7298	015176				TYPDEC \$PASS
7299	015204	013700	000042		MOV 42,R0 ;:GET MONITOR ADDRESS
7300	015210	001456			BEQ \$DOAGAIN ;:BRANCH IF NO MONITOR
7301	015212	022700	002000		\$ZAP42: CMP #STACK,R0 ;ARE WE UNDER RT11
7302	015216	001453			BEQ \$DOAGAIN ;YES - BRANCH
7303					;WE ARE UNDER (HEAVEN HELP US) XXDP!
7304	015220				PUSH R0
7305	015222	004737	047664		CALL SHUTUP
7306	015226				POP R0
7307	015230	000005			RESET ;:CLEAR THE WORLD
7308	015232	004710			\$ENDAD: CALL (R0) ;:GO TO MONITOR
7309	015234	000240			NOP ;:SAVE ROOM
7310	015236	000240			NOP ;:FOR
7311	015240	000240			NOP ;:ACT11
7312	015242				\$DOAGN: ;UNDO SHUTUP STUFF
7313					;RESTORE STACK
7314					;ENERGIZE UNIBUS MAP & 22 BIT ADDRESSING
7315					;ENERGIZE MEMORY MANAGEMENT
7316					;PUT LOADERS BACK HOME
7317	015242	013706	002560		MOV KSTACK,SP
7318	015246	005737	002450		TST NO22BIT ;IS THIS AN 11/44 OR 11/24?
7319	015252	001003			BNE 1\$
7320	015254	052737	000060	172516	1\$: BIS #BIT5!BIT4,MMR3
7321	015262	104420			ENERGIZE ;:TURN ON MEMORY MANAGEMENT
7322	015264	013700	002562		MOV LOADHOME,R0 ;:DESTINATION BANK
7323	015270	012701	000001		MOV #1,R1 ;:SOURCE BANK
7324	015274	004737	046410		CALL BANKMOV
7325	015300				IF APTFLAG IS TRUE
7326	015306				IF \$USWR EQ \$PASS
7327	015316	012701	000050		APTHANG: MOV #50,R1
7328	015322	077001			2\$: SOB R0,2\$
7329	015324	062737	000001	065650	ADD #1,\$DEVCT
7330	015332	005537	065652		ADC \$UNIT
7331	015336	077107			SOB R1,2\$
7332	015340	005237	065646		INC \$PASS
7333	015344	000764			BR APTHANG

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 21-APR-82 09:41 PAGE 189-1  
END OF PASS ROUTINE

7334 015346

7335 015346

7336 015346 000137 014742

END :OF IF \$USWR

END :OF IF APTFLAG

\$DOAGAIN: JMP LOOP

;RETURN

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 191  
 END OF PASS ROUTINE

7339 015352

DOBACK: SUBTST <<WRITE BACKGROUND PATTERNS>>

\*\*\*\*\*  
 :\*SUBTEST WRITE BACKGROUND PATTERNS  
 :\*\*\*\*\*

7340 015352 005037 002110  
 7341 015356  
 7342 015362 004737 047020  
 7343 015366  
 7344 015402  
 7345 015416 004737 020274  
 7346 015422 005037 002106  
 7347 015426  
 7348 015434  
 7349 015434  
 7350 015450 000207

CLR PATTERN  
 FOR BANK := #0 TO LASTBANK  
 CALL EXBANK  
 IF ACFLAG IS TRUE AND RRFLAG IS FALSE  
 SET HEADER,MUT  
 CALL MKTEST ;CALL MJTEST WOULD ALSO WORK  
 CLR MUT  
 SET HEADER  
 END ;OF IF ACFLAG  
 END ;OF FOR BANK  
 RETURN

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 193  
MTEST MODES

7353  
7354  
7355 015452

## .SBTTL MTEST MODES

```
BAFPAF: SUBTST <<BANKS FORWARD,PATTERNS FORWARD      **RECURSIVE**>>
:*****
:*SUBTEST      BANKS FORWARD,PATTERNS FORWARD      **RECURSIVE**
:*****
:      CLR      BANK      ;SET BANK TO 0
:      ;START OF BANK LOOP
1$:      CALL    CXBANK      ;EXAMINE BANK
:      TST      ACFLAG      ;CAN WE ACCESS THIS BANK?
:      BEQ      4$           ;NO - GO TO BANK LOOP TERMINATION
:      TST      RRFLAG      ;RELOCATION REQUIRED?
:      BNE      4$           ;YES - GO TO BANK LOOP TERMINATION
:      CLR      PATTERN      ;SET PATTERN TO 0
:      ;START OF PATTERN LOOP
2$:      CALL    MTEST      ;GO TEST CORRECT MEMORY
:      ;TERMINATION OF PATTERN LOOP
:      CALL    INCPAT      ;GO SEE IF THIS IS THE LAST PATTERN
:      BNE      2$         ;NO - LOOP ON THIS PATTERN
:      ;TERMINATION OF BANK LOOP
4$:      CLR      CONTFLAG
:      CALL    INCBNK      ;NEXT HIGHER BANK
:      BGE      1$         ;IF NOT DONE - LOOP ON THIS BANK
:      ;END OF LOOPS
:      TST      RLFLAG      ;HAVE WE BEEN RELOCATED?
:      BEQ      5$         ;NO - SKIP
:      RETURN      ;YES - RETURN
5$:      CALL    RELOCATE      ;MOVE & MAP PROGRAM
:      ON.ERROR THEN $RETURN
:      ;**NOTE** RECURSIVE CALL
:      CALL    BAFPAF      ;CALL SELF
:      CALL    UNRELOCATE    ;UNMOVE & UNMAP PROGRAM
:      RETURN
```

7356 015452 005037 002100  
7357  
7358 015456 004737 047020  
7359 015462 005737 002114  
7360 015466 001412  
7361 015470 005737 002122  
7362 015474 001007  
7363 015476 005037 002110  
7364  
7365 015502 004737 016752  
7366  
7367 015506 004737 047464  
7368 015512 001373  
7369  
7370 015514 005037 002220  
7371 015520 004737 047510  
7372 015524 002354  
7373  
7374 015526 005737 002124  
7375 015532 001401  
7376 015534 000207  
7377 015536 004737 045172  
7378 015542  
7379  
7380 015546 004737 015452  
7381 015552 004737 046056  
7382 015556 000207

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 195  
BANKS FORWARD,PATTERNS FORWARD \*\*RECURSIVE\*\*

7385 015560

```

BAFPAR: SUBTST <<BANKS FORWARD,PATTERNS REVERSE **RECURSIVE**>>
:*****
:*SUBTEST      BANKS FORWARD,PATTERNS REVERSE **RECURSIVE**
:*****
1$: CLR      BANK          ;SET BANK TO 0
   ;START OF BANK LOOP
   CALL     EXBANK        ;EXAMINE BANK
   TST      ACFLAG        ;CAN WE ACCESS THIS BANK?
   BEQ      4$            ;NO - GO TO BANK LOOP TERMINATION
   TST      RRFLAG        ;RELOCATION REQUIRED?
   BNE      4$            ;YES - GO TO BANK LOOP TFRMINATION
   CALL     SETPAT        ;SET HIGH PATTERN FOR CORRECT MEMORY
   ;START OF PATTERN LOOP
2$: CALL     MTEST        ;GO TEST CORRECT MEMORY
   ;T-RMINATION OF PATTERN LOOP
   DEC      PATTERN        ;IS THIS THE LAST PATTERN?
   BPL      2$            ;NO - LOOP ON THIS PATTERN
   ;TERMINATION OF BANK LOOP
4$: CLR      CONTFLAG
   CALL     INCBNK        ;NEXT HIGHER BANK
   BGE      1$            ;IF NOT DONE - LOOP ON THIS BANK
   ;END OF LOOPS
   TST      RLFLAG        ;HAVE WE BEEN RELOCATED?
   BEQ      5$            ;NO - SKIP
   RETURN   ;YES - RETURN
5$: CALL     RELOCATE      ;MOVE & MAP PROGRAM
   ON.ERROR THEN $RETURN
   ;**NOTE** RECURSIVE CALL
   CALL     BAFPAR        ;CALL SELF
   CALL     UNRELOCATE    ;UNMOVE & UNMAP PROGRAM
   RETURN

```

```

7386 015560 005037 002100
7387
7388 015564 004737 047020
7389 015570 005737 002114
7390 015574 001412
7391 015576 005737 002122
7392 015602 001007
7393 015604 004737 047500
7394
7395 015610 004737 016752
7396
7397 015614 005337 002110
7398 015620 100373
7399
7400 015622 005037 002220
7401 015626 004737 047510
7402 015632 002354
7403
7404 015634 005737 002124
7405 015640 001401
7406 015642 000207
7407 015644 004737 045172
7408 015650
7409
7410 015654 004737 015560
7411 015660 004737 046056
7412 015664 000207

```

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 197  
BANKS FORWARD,PATTERNS REVERSE \*\*RECURSIVE\*\*

7415 015666

```
BAWPAF: SUBIST <<BANKS WORST FIRST,PATTERNS FORWARD **RECURSIVE**>>
:*****
:*SUBTEST      BANKS WORST FIRST,PATTERNS FORWARD **RECURSIVE**
:*****
      CLR      BANK      ;SET BANK TO 0
      ;START OF BANK LOOP
1$:   CALL     E)BANK      ;EXAMINE BANK
      TST      ACFLAG     ;CAN WE ACCESS THIS BANK?
      BEQ      4$         ;NO - GO TO BANK LOOP TERMINATION
      TST      BMFLAG     ;IS THIS BAD MEMORY (WORST FIRST)?
      BEQ      4$         ;NO - GO TO BANK LOOP TERMINATION
      TST      RRFLAG     ;RELOCATION REQUIRED?
      BNE      4$         ;YES - GO TO BANK LOOP TERMINATION
      CLR      PATTERN    ;SET PATTERN TO 0
      ;START OF PATTERN LOOP
2$:   CALL     MTEST      ;GO TEST CORRECT MEMORY
      ;TERMINATION OF PATTERN LOOP
      CALL     INCPAT     ;GO SEE IF THIS IS THE LAST PATTERN
      BNE      2$         ;NO - LOOP ON THIS PATTERN
      ;TERMINATION OF BANK LOOP
4$:   CLR      CONTFLAG
      CALL     INCBNK     ;NEXT HIGHER BANK
      BGE      1$         ;IF NOT DONE - LOOP ON THIS BANK
      ;END OF LOOPS
      COM      WORST      ;IS THIS AN EVEN NUMBERED PASS?
      BNE      5$         ;YES - SKIP
      ;**NOTE** RECURSIVE CALL
      CALL     BAWPAF     ;CALL SELF
      RETURN
5$:   TST      RLFLAG     ;HAVE WE BEEN RELOCATED?
      BEQ      6$         ;NO - SKIP
      RETURN          ;YES - RETURN
6$:   CALL     RELOCATE    ;MOVE & MAP PROGRAM
      ON.ERROR THEN $RETURN
      ;**NOTE** RECURSIVE CALL
      CALL     BAWPAF     ;CALL SELF
      CALL     UNRELOCATE ;UNMOVE & UNMAP PROGRAM
      RETURN
```

```
7416 015666 005037 002100
7417
7418 015672 004737 047020
7419 015676 005737 002114
7420 015702 001415
7421 015704 005737 002126
7422 015710 001412
7423 015712 005737 002122
7424 015716 001007
7425 015720 005037 002110
7426
7427 015724 004737 016752
7428
7429 015730 004737 047464
7430 015734 001373
7431
7432 015736 005037 002220
7433 015742 004737 047510
7434 015746 002351
7435
7436 015750 005137 002564
7437 015754 001003
7438
7439 015756 004737 015666
7440 015762 000207
7441 015764 005737 002124
7442 015770 001401
7443 015772 000207
7444 015774 004737 045172
7445 016000
7446
7447 016004 004737 015666
7448 016010 004737 046056
7449 016014 000207
```



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 199  
BANKS WORST FIRST,PATTERNS FORWARD \*\*RECURSIVE\*\*

7452 016016

```
BAWPAR: SUBTST <<BANKS WORST FIRST,PATTERNS REVERSE **RECURSIVE**>>
:*****
:*SUBTEST      BANKS WORST FIRST,PATTERNS REVERSE **RECURSIVE**
:*****
7453 016016 005037 002100      CLR      BANK      ;SET BANK TO 0
7454                                ;START OF BANK LOOP
7455 016022 004737 047020 1$:  CALL      EXBANK      ;EXAMINE BANK
7456 016026 005737 002114      TST      ACFLAG      ;CAN WE ACCESS THIS BANK?
7457 016032 001415      BEQ      4$      ;NO - GO TO BANK LOOP TERMINATION
7458 016034 005737 002126      TST      BMFLAG      ;IS THIS BAD MEMORY (WORST FIRST)
7459 016040 001412      BEQ      4$      ;NO - GO TO BANK LOOP TERMINATION
7460 016042 005737 002122      TST      RRFLAG      ;RELOCATION REQUIRED?
7461 016046 0010J7      BNE      4$      ;YES - GO TO BANK LOOP TERMINATION
7462 016050 004737 047500      CALL      SETPAT      ;SET HIGH PATTERN FOR CORRECT MEMORY
7463                                ;START OF PATTERN LOOP
7464 016054 004737 016752 2$:  CALL      MTEST      ;GO TEST CORRECT MEMORY
7465                                ;TERMINATION OF PATTERN LOOP
7466 016060 005337 002110      DEC      PATTERN      ;IS THIS THE LAST PATTERN?
7467 016064 100373      BPL      2$      ;NO - LOOP ON THIS PATTERN
7468                                ;TERMINATION OF BANK LOOP
7469 016066 005037 10 220 4$:  CLR      CONTFLAG
7470 016072 004737 1,510      CALL      INCBNK      ;NEXT HIGHER BANK
7471 016076 002351      BGE      1$      ;IF NOT DONE - LOOP ON THIS BANK
7472                                ;END OF LOOPS
7473 016100 005137 002564      COM      WORST      ;IS THIS AN EVEN NUMBERED PASS?
7474 016104 001003      BNE      5$      ;YES - SKIP
7475                                ;**NOTE** RECURSIVE CALL
7476 016106 004737 016016      CALL      BAWPAR      ;CALL SELF
7477 016112 000207      RETURN
7478 016114 005737 002124 5$:  TST      RLFLAG      ;HAVE WE BEEN RELOCATED?
7479 016120 001401      BEQ      6$      ;NO - SKIP
7480 016122 000207      RETURN      ;YES - RETURN
7481 016124 004737 045172 6$:  CALL      RELOCATE      ;MOVE 3 MAP PROGRAM
7482 016130      ON.ERROR THEN $RETURN
7483                                ;**NOTE** RECURSIVE CALL
7484 016134 004737 016016      CALL      BAWPAR      ;CALL SELF
7485 016140 004737 046056      CALL      UNRELOCATE      ;UNMOVE & UNMAP PROGRAM
7486 016144 000207      RETURN
```

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 201  
BANKS WORST FIRST,PATTERNS REVERSE \*\*RECURSIVE\*\*

7489 016146

```

PAFBAF: SUBTST <<PATTERNS FORWARD,BANKS FORWARD      **RECURSIVE**>>
:*****
:*SUBTEST      PATTERNS FORWARD,BANKS FORWARD      **RECURSIVE**
:*****
      CLR      PATTERN      ;SET PATTERN TO 0
      ;START OF PATTERN LOOP
1$:    CLR      BANK        ;SET BANK TO 0
      ;START OF BANK LOOP
2$:    CALL     EXBANK       ;EXAMINE BANK
      CALL     BANKOK       ;CORRECT MEMORY FOR THIS BANK?
      BNE      4$           ;NO - GO TO BANK LOOP TERMINATOR
      TST      ACFLAG       ;CAN WE ACCESS THIS BANK?
      BEQ      4$           ;NO - GO TO BANK LOOP TERMINATION
      TST      RRFLAG       ;RELOCATION REQUIRED?
      BNE      4$           ;YES - GO TO BANK LOOP TERMINATION
      CALL     MTEST        ;GO TEST CORRECT MEMORY
      ;TERMINATION OF BANK LOOP
4$:    CLR      CONFLAG
      CALL     INCBNK       ;NEXT HIGHER BANK
      BGE      2$           ;IF NOT DONE - LOOP ON THIS BANK
      ;TERMINATION OF PATTERN LOOP
      CALL     INCRPT       ;NEXT HIGHER PATTERN
      BNE      1$           ;OK - LOOP; ELSE CONTINUE
      ;END OF LOOPS
      COM      TMFLAG       ;COMPLEMENT TYPE OF MEMORY
      BEQ      5$           ;IS THIS AN EVEN NUMBER PASS?
      ;YES - SKIP
      ;**NOTE** RECURSIVE CALL
      CALL     PAFBAF       ;CALL SELF
      RETURN
5$:    TST      RLFLAG       ;HAVE WE BEEN RELOCATED?
      BEQ      6$           ;NO - SKIP
      RETURN             ;YES - RETURN
6$:    CALL     RELOCATE     ;MOVE & MAP PROGRAM
      ON.ERROR THEN $RETURN
      ;**NOTE** RECURSIVE CALL
      CALL     PAFBAF       ;CALL SELF
      CALL     UNRELOCATE   ;UNMOVE & UNMAP PROGRAM
      RETURN

```

```

7490 016146 005037 002110
7491
7492 016152 005037 002100
7493
7494 016156 004737 047020
7495 016162 004737 047446
7496 016166 001010
7497 016170 005737 002114
7498 016174 001405
7499 016176 005737 002122
7500 016202 001002
7501 016204 004737 016752
7502
7503 016210 005037 002220
7504 016214 004737 047510
7505 016220 002356
7506
7507 016222 004737 047464
7508 016226 001351
7509
7510 016230 005137 002132
7511
7512 016234 001403
7513
7514 016236 004737 016146
7515 016242 000207
7516 016244 005737 002124
7517 016250 001401
7518 016252 000207
7519 016254 004737 045172
7520 016260
7521
7522 016264 004737 016146
7523 016270 004737 046056
7524 016274 000207

```

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 203  
 PATTERNS FORWARD,BANKS FORWARD \*\*RECURSIVE\*\*

7527 016276

```

PAFBAW: SUBTST <<PATTERNS FORWARD,BANKS WORST FIRST **RECURSIVE**>>
:*****
:*SUBTEST PATTERNS FORWARD,BANKS WORST FIRST **RECURSIVE**
:*****
      CLR PATTERN ;SET PATTERN TO 0
      ;START OF PATTERN LOOP
1$: CLR BANK ;SET BANK TO 0
      ;START OF BANK LOOP
2$: CALL EXBANK ;EXAMINE BANK
      CALL BANKOK ;CORRECT MEMORY FOR THIS BANK?
      BNE 4$ ;NO - GO TO BANK LOOP TERMINATOR
      TST ACFLAG ;CAN WE ACCESS THIS BANK?
      BEQ 4$ ;NO - GO TO BANK LOOP TERMINATION
      TST BFLAG ;IS THIS BAD MEMORY (WORST FIRST)
      BEQ 4$ ;NO - GO TO BANK LOOP TERMINATION
      TST RRFLAG ;RELOCATION REQUIRED?
      BNE 4$ ;YES - GO TO BANK LOOP TERMINATION
      CALL MTEST ;GO TEST CORRECT MEMORY
      ;TERMINATION OF BANK LOOP
4$: CLR CONFLAG
      CALL INCBNK ;NEXT HIGHER BANK
      BGE 2$ ;IF NOT DONE - LOOP ON THIS BANK
      ;TERMINATION OF PATTERN LOOP
      CALL INCRPT ;NEXT HIGHER PATTERN
      BNE 1$ ;OK - LOOP; ELSE CONTINUE
      ;END OF LOOPS
      COM TMFLAG ;COMPLEMENT TYPE OF MEMORY
      BEQ 5$ ;IS THIS AN EVEN NUMBER PASS?
      ;**NOTE** RECURSIVE CALL
      CALL PAFBAW ;CALL SELF
      RETURN
5$: COM WORST ;4TH PASS?
      BNE 6$ ;YES - SKIP
      ;**NOTE** RECURSIVE CALL
      CALL PAFBAW ;CALL SELF
      RETURN
6$: TST RLFLAG ;HAVE WE BEEN RELOCATED?
      BEQ 7$ ;NO - SKIP
      RETURN ;YES - RETURN
7$: CALL RELOCATE ;MOVE & MAP PROGRAM
      ON.ERROR THEN $RETURN
      ;**NOTE** RECURSIVE CALL
      CALL PAFBAW ;CALL SELF
      CALL UNRELOCATE ;UNMOVE & UNMAP PROGRAM
      RETURN

```

7528 016276 005037 002110  
 7529  
 7530 016302 005037 002100  
 7531  
 7532 016306 004737 047020  
 7533 016312 004737 047446  
 7534 016316 001013  
 7535 016320 005737 002114  
 7536 016324 001410  
 7537 016326 005737 002126  
 7538 016332 001405  
 7539 016334 005737 002122  
 7540 016340 001002  
 7541 016342 004737 016752  
 7542  
 7543 016346 005037 002220  
 7544 016352 004737 047510  
 7545 016356 002353  
 7546  
 7547 016360 004737 047464  
 7548 016364 001346  
 7549  
 7550 016366 005137 002132  
 7551  
 7552 016372 001403  
 7553  
 7554 016374 004737 016276  
 7555 016400 000207  
 7556 016402 005137 002564  
 7557 016406 001003  
 7558  
 7559 016410 004737 016276  
 7560 016414 000207  
 7561 016416 005737 002124  
 7562 016422 001401  
 7563 016424 000207  
 7564 016426 004737 045172  
 7565 016432  
 7566  
 7567 016436 004737 016276  
 7568 016442 004737 046056  
 7569 016446 000207

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 205  
 PATTERNS FORWARD,BANKS WORST FIRST \*\*RECURSIVE\*\*

7572 016450

```

PARBAF: SUBST <<PATTERNS REVERSE,BANKS FORWARD **RECURSIVE**>>
:*****
:*SUBTEST PATTERNS REVERSE,BANKS FORWARD **RECURSIVE**
:*****
7573 016450 004737 047500 CALL HIPAT ;SET HIGHEST PATTERNS
7574 ;START OF PATTERN LOOP
7575 016454 005037 002100 1$: CLR BANK ;SET BANK TO 0
7576 ;START OF BANK LOOP
7577 016460 004737 047020 2$: CALL EXBANK ;EXAMINE BANK
7578 016464 004737 047446 CALL BANKOK ;CORRECT MEMORY FOR THIS BANK?
7579 016470 001010 BNE 4$ ;NO - GO TO BANK LOOP TERMINATOR
7580 016472 005737 002114 TST ACFLAG ;CAN WE ACCESS THIS BANK?
7581 016476 001405 BEQ 4$ ;NO - GO TO BANK LOOP TERMINATION
7582 016500 005737 002122 TST RRFLAG ;RELOCATION REQUIRED?
7583 016504 001002 BNE 4$ ;YES - GO TO BANK LOOP TERMINATION
7584 016506 004737 016752 CALL MTEST ;GO TEST CORRECT MEMORY
7585 ;TERMINATION OF BANK LOOP
7586 016512 005037 002220 4$: CLR CONTFLAG
7587 016516 004737 047510 CALL INCBNK ;NEXT HIGHER BANK
7588 016522 002356 BGE 2$ ;IF NOT DONE - LOOP ON THIS BANK
7589 ;TERMINATION OF PATTERN LOOP
7590 016524 005337 002110 DEC PATTERN ;NEXT LOWER PATTERN
7591 016530 100351 BPL 1$ ;OK - LOOP; ELSE CONTINUE
7592 ;END OF LOOPS
7593 016532 005137 002132 COM TMFLAG ;COMPLEMENT TYPE OF MEMORY
7594 ;IS THIS AN EVEN NUMBER PASS?
7595 016536 001403 BEQ 5$ ;YES - SKIP
7596 ;**NOTE** RECURSIVE CALL
7597 016540 004737 016450 CALL PARBAF ;CALL SELF
7598 016544 000207 RETURN
7599 016546 005737 002124 5$: TST RLFLAG ;HAVE WE BEEN RELOCATED?
7600 016552 001401 BEQ 6$ ;NO - SKIP
7601 016554 000207 RETURN ;YES - RETURN
7602 016556 004737 045172 6$: CALL RELOCATE ;MOVE & MAP PROGRAM
7603 016562 ON.ERROR THEN $RETURN
7604 ;**NOTE** RECURSIVE CALL
7605 016566 004737 016450 CALL PARBAF ;CALL SELF
7606 016572 004737 046056 CALL UNRELOCATE ;UNMOVE & UNMAP PROGRAM
7607 016576 000207 RETURN

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 207  
 PATTERNS REVERSE,BANKS FORWARD \*\*RECURSIVE\*\*

7610 016600

```

PARBAW: SUBTST <<PATTERNS REVERSE,BANKS WORST FIRST **RECURSIVE**>>
:*****
:*SUBTEST      PATTERNS REVERSE,BANKS WORST FIRST **RECURSIVE**
:*****
CALL      HIPAT      ;SET HIGHEST PATTERN
;START OF PATTERN LOOP
1$: CLR      BANK      ;SET BANK TO 0
;START OF BANK LOOP
2$: CALL     EXBANK    ;EXAMINE BANK
CALL      BANKOK      ;CORRECT MEMORY FOR THIS BANK?
BNE       4$          ;NO - GO TO BANK LOOP TERMINATOR
TST       ACFLAG      ;CAN WE ACCESS THIS BANK?
BEQ       4$          ;NO - GO TO BANK LOOP TERMINATION
TST       BMFLAG      ;IS THIS BAD MEMORY (WORST FIRST)
BEQ       4$          ;NO - GO TO BANK LOOP TERMINATION
TST       RRFLAG      ;RELOCATION REQUIRED?
BNE       4$          ;YES - GO TO BANK LOOP TERMINATION
CALL      MTEST       ;GO TEST CORRECT MEMORY
;TERMINATION OF BANK LOOP
4$: CLR     CONTFLAG
CALL      INCBANK     ;NEXT HIGHER BANK
BGE       2$          ;IF NOT DONE - LOOP ON THIS BANK
;TERMINATION OF PATTERN LOOP
DEC       PATTERN     ;NEXT LOWER PATTERN
BPL       1$          ;OK - LOOP; ELSE CONTINUE
;END OF LOOPS
COM       TMFLAG      ;COMPLEMENT TYPE OF MEMORY
;IS THIS AN EVEN NUMBER PASS?
BEQ       5$          ;YES - SKIP
;***NOTE** RECURSIVE CALL
CALL      PARBAW      ;CALL SELF
RETURN
5$: COM     WORST      ;4TH PASS?
BNE       6$          ;YES - SKIP
;***NOTE** RECURSIVE CALL
CALL      PARBAW      ;CALL SELF
RETURN
6$: TST     RLFLAG     ;HAVE WE BEEN RELOCATED?
BEQ       7$          ;NO - SKIP
RETURN    ;YES - RETURN
7$: CALL    RELOCATE   ;MOVE & MAP PROGRAM
ON.ERROR THEN $RETURN
;***NOTE** RECURSIVE CALL
CALL      PARBAW      ;CALL SELF
CALL      UNRELOCATE  ;UNMOVE & UNMAP PROGRAM
RETURN

```

```

7611 016600 004737 047500
7612
7613 016604 005037 002100
7614
7615 016610 004737 047020
7616 016614 004737 047446
7617 016620 001013
7618 016622 005737 002114
7619 016626 001410
7620 016630 005737 002126
7621 016634 001405
7622 016636 005737 002122
7623 016642 001002
7624 016644 004737 016752
7625
7626 016650 005037 002220
7627 016654 004737 047510
7628 016660 002353
7629
7630 016662 005337 002110
7631 016666 100346
7632
7633 016670 005137 002132
7634
7635 016674 001403
7636
7637 016676 004737 016600
7638 016702 000207
7639 016704 005137 002564
7640 016710 001003
7641
7642 016712 004737 016600
7643 016716 000207
7644 016720 005737 002124
7645 016724 001401
7646 016726 000207
7647 016730 004737 045172
7648 016734
7649
7650 016740 004737 016600
7651 016744 004737 046056
7652 016750 000207

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 209  
 PATTERNS REVERSE,BANKS WORST FIRST \*\*RECURSIVE\*\*

7655 016752

MTEST: SUBTST &lt;&lt;SUBR SETUP MEMORY TEST&gt;&gt;

\*\*\*\*\*  
 :SUBTEST SUBR SETUP MEMORY TEST  
 \*\*\*\*\*

7656 016752  
 7657 016760  
 7658 016766 005037 002262  
 7659 016772 005737 002116  
 7660 016776 001413  
 7661 017000  
 7662 017000  
 7663 017006  
 7664 017014 004737 017046  
 7665 017020  
 7666 017020  
 7667 017020 004737 020274  
 7668 017024 000402  
 7669 017026 004737 020514  
 7670 017032 005037 002106  
 7671 017036  
 7672 017044 000207

SET HEADER ;INITIALIZE HEADER MESSAGE TYPEOUT  
 SET MUT ;INDICATE THERE IS A MEMORY UNDER TEST  
 CLR PASFLG  
 TST MKFLAG ;ECC?  
 BEQ MT1 ;NO - SKIP  
 BEGIN HOLDLOOP  
 IF CONTFLAG IS TRUE THEN LEAVE HOLDLOOP  
 IF SKIPMK IS FALSE  
 CALL MKCONTROL  
 END; OF IF SKIPMK  
 END HOLDLOOP  
 CALL MKTEST ;YES - DO ECC TESTS  
 BR MT2  
 MT1: CALL MJTEST ;DO PARITY TESTS  
 MT2: CLR MUT ;NOW - NO MEMORY UNDER TEST  
 SET HEADER ;ALLOW HEADERS NORMAL  
 RETURN

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 211  
 SUBR SETUP MEMORY TEST

7675 017046

```

MKCONTROL:SUBTST      <<SUBR TEST ECC CSR LOGIC DISPATCH>>
:*****
:SUBTEST      SUBR      TEST ECC CSR LOGIC DISPATCH
:*****
:THE NEXT TWO MODULES SOLVE THE PROBLEM OF
:HOW TO RUN THE CSR TESTS ON EACH ECC MEMORY
:
:IF SELONLY IS TRUE THEN $RETURN
:IF INHECC IS TRUE THEN $RETURN
PUSH      BANK,R0,R1,R2,R3
MOV      #FIRST,CSRFBANK      ;SET FIRST TEST ADDRESS TO FIRST ADDR.
MOV      #LAST,CSRLBANK
CLR      CSRINT
CLR      SPLTCSR
CLR      CSRLOOP      ; AND ZERO THE LOOP COUNTER
MOV      BANKINDEX,R0      ;GET THE BANK INDEX
MOV      CONFIG(R0),R1      ;GET CSR NUMBER
SWAB      R1
BIC      #^C17,R1
ASL      R1
MOV      R1,CSRHOLD      ;STORE IN THE LOW BYTE
TST      INTFLAG      ;IS THIS BANK INTERLEAVED?
BEQ      1$      ;BRANCH IF NOT INTERLEAVED
INC      SPLTCSR
MOV      #12000,CSRLBANK      ;WE MUST LOOP TWICE FOR AN INTERLEAVED BANK
INC      CSRLOOP
INC      CSRINT
MOV      CONFIG(R0),R1      ;GET THE INTERLEAVE CSR NUMBER
ASH      #-3,R1
BIC      #^C17000,R1
BIS      R1,CSRHOLD      ;STORE IT IN CSRHOLD'S UPPER BYTE
CLR      R3
1$: MKLOOP: MOVB      CSRHOLD(R3),CSRNO      ;CLEAR ANY UNNECESSARY BITS
BIC      #^C36,CSRNO
FOR MKCNT := #0 TO CSRINT
FOR CSRFIRST := CSRFBANK TO CSRLBANK BY #4000
MAP BANK      ;MAP TEST SPACE TO BANK
INvalidate      ;INVALIDATE BACKGROUND PATTERN
BEGIN CSRSTUFF
CLR      SUCCESS
IF ACFLAG IS TRUE AND RRFLAG IS FALSE
MOV      CSRFIRST,CSRLAST
ADD      #4000,CSRLAST
FOR TESTADD := CSRFIRST TO CSRLAST BY #4
MOV      TESTADD,TESTADD+2
TST      SPLTCSR
BEQ      1$
ADD      #40000,TESTADD+2
BR      2$
1$: ADD      #2,TESTADD+2
2$: CALL      SBETEST
ON.NOERROR
CACHOFF      ;TURN CACHE OFF
CLR      NOPAR      ;INDICATE PARITY ACTION
FOR I := #0 TO #27
SET      HEADER
CLR      PASFLG

```

```

7676
7677
7678
7679 017046
7680 017056
7681 017066
7682 017102 012737 060000 002230
7683 017110 012737 157776 002232
7684 017116 005037 002234
7685 017122 005037 002236
7686 017126 005037 002326
7687 017132 013700 002102
7688 017136 016001 002650
7689 017142 000301
7690 017144 042701 177760
7691 017150 006301
7692 017152 010137 002522
7693 017156 005737 002134
7694 017162 001421
7695 017164 005237 002236
7696 017170 012737 120000 002232
7697 017176 005237 002326
7698 017202 005237 002234
7699 017206 016001 002650
7700 017212 072127 177775
7701 017216 042701 160777
7702 017222 050137 002522
7703 017226 005003
7704 017230 116337 002522 002150
7705 017236 042737 177741 002150
7706 017244
7707 017250
7708 017256
7709 017272 104511
7710 017274
7711 017274 005037 002330
7712 017300
7713 017314 013737 002224 002226
7714 017322 062737 004000 002226
7715 017330
7716 017336 013737 002406 002410
7717 017344 005737 002236
7718 017350 001404
7719 017352 062737 040000 002410
7720 017360 000403
7721 017362 062737 000002 002410
7722 017370 004737 017670
7723 017374
7724 017376 104424
7725 017400 005037 002074
7726 017404
7727 017410
7728 017416 005037 002262

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 211-1  
 SUBR TEST ECC CSR LOGIC DISPATCH

7729	017422				LET R0 := 1	
7730	017426				PUSH R3	;SAVE LOOP COUNTER
7731	017430	010637	002144		MOV SP,CTLKVEC	;SAVE VECTOR IN CSR OF ^K
7732	017434	62737	000002	002144	SUB #2,CTLKVEC	
7733	017442	004737	020174		CALL CSRCASE	
7734	017446				POP R3	;RESTORE LOOP COUNTER
7735	017450				END ;OF FOR I	
7736	017464	104423			CACHON	;TURN CACHE ON
7737	017466				SET SUCCESS	
7738	017474				LEAVE CSRSTUFF	
7739	017476				END ;OF ON.NOERROR	
7740	017476				END ;OF FOR TESTADD	
7741	017514				END ;OF IF	
7742	017514				END CSRSTUFF	
7743	017514				IF SUCCESS IS FALSE	
7744	017522				TYPE MSGA34	
7745	017526				TYPOCS BANK,<TYPES BANK NUMBER>,3	
7746	017536				TYPE MSGB34	
7747	017542	004737	057476		CALL PERBNK	
7748	017546				END ;OF IF SUCCESS	
7749	017546				END ;OF FOR CSRFIRST	
7750	017564	005237	002236		INC SPLTCSR	
7751	017570				END ;OF FOR MKCNT	
7752	017604	062737	000002	002230	ADD #2,CSRFBNK	
7753	017612	012737	000001	002236	MOV #1,SPLTCSR	
7754	017620	005203			INC R3	
7755	017622	020337	002326		CMP R3,CSRLOOP	
7756	017626	003002			BGT 1\$	
7757	017630	000137	017230		JMP MKLOOP	
7758	017634	104472			1\$: ECCINIT	;TRAP ON DOUBLE BIT ERRORS (NORMAL)
7759	017636				SET CONTFLAG	
7760	017644	005037	002236		CLR SPLTCSR	
7761	017650				POP R3,R2,R1,R0,BANK	
7762	017664	000207			RETURN	
7763	017666	000000			MKCNT: .WORD 0	;COUNTER FOR MKLOOP



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 213  
 SUBR TEST ECC CSR LOGIC DISPATCH

7766 017670

SBETEST:SUBTST &lt;&lt;CHECK FOR SBE FREE LOCATIONS&gt;&gt;

\*\*\*\*\*  
 \*SUBTEST CHECK FOR SBE FREE LOCATIONS  
 \*\*\*\*\*

7767

;IN ORDER TO DETERMINE IF A LOCATION IS SBE FREE I DO THIS

7768

;WRITE ZEROS WITH ECC DISABLE

7769

;READ ZEROS BACK

7770

;IF NOT ZEROS THEN RETURN ERROR

7771

7772

;WRITE ZEROS WITH ECC ENABLED BUT TRAPS DISABLED

7773

7774

;READ ZEROS BACK

7775

;IF NOT ZEROS THEN RETURN ERROR

7776

7777

;TEST THE LOCATION FROM THE PAR'S (WITH NO PROGRAM FETCHES)

7778

;IF THERE WERE ANY SBE'S OR DBE'S THEN RETURN ERROR

7779

7780

;COMPLIMENT ZEROS TO ONES WITH ECC DISABLE

7781

;READ ONES BACK

7782

;IF NOT ONES THEN RETURN ERROR

7783

7784

;WRITE 100,100000,00000 (CHECKBITS COMPLIMENT OF BEFORE)

7785

; WITH ECC ENABLED BUT TRAPS DISABLED

7786

;TEST THE LOCATION FROM THE PAR'S (WITH NO PROGRAM FETCHES)

7787

;IF THERE WERE ANY SBE'S OR DBE'S THEN RETURN ERROR

7788

7789

;IF NONE OF THE ABOVE FORCES A RETURN ERROR THEN RETURN NO.ERROR

7790

;ENABL LSB

7791 017670

PUSH R0,R1,R4

;PUSH R0,R1,R4 ONTO STACK

7792 017676 013701 002406

MOV TESTADD,R1

7793 017702 013704 002410

MOV TESTADD+2,R4

7794 017706

TESTAREA

;ENTER TEST MODE

7795 017714 104424

CACHOFF

;TURN CACHE OFF

7796 017716 104471

ECC1DIS

;DISABLE ECC ON 1 SELECTED CSR

7797 017720

CLEAR (R1),(R4)

7798 017724 005711

TST (R1)

7799 017726 001107

BNE SBENT

7800 017730 005714

TST (R4)

7801 017732 001105

BNE SBENT

7802

7803 017734 104503

CLR1CSR

;CLEAR 1 SELECTED CSR

7804 017736

CLEAR (R1),(R4)

7805 017742 005711

TST (R1)

7806 017744 001100

BNE SBENT

7807 017746 005714

TST (R4)

7808 017750 001076

BNE SBENT

7809

7810 017752 104510

TSTREAD

;TEST LOC (R1) &amp; TST FOR SBE (WITHOUT FETCHES)

7811 017754

IF #BIT15:BIT4 SET IN CSR

7812 017764

SET SKPERR

;DISABLE ERRGEN'S ERROR PRINTOUT

7813 017772 104512

ERRGEN

7814 017774 013700 002454

OV E:RADD,R0

7815 020000 072027 177774

ASH #-4,R0

7816 020004 042700 177600

BIC #^C177,R0

7817 020010

IF BANK EQ R0 THEN GOTO SBENT

7818 020016

END: OF IF #9IT15

7819 020016 104471

ECC1DIS

;DISABLE ECC ON 1 SELECTED CSR

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 213-1  
CHECK FOR SBE FREE LOCATIONS

7820	020020	005111		COM	(R1)	
7821	020022	005114		COM	(R4)	
7822	020024	023711	002600	CMP	ONES,(R1)	
7823	020030	001046		BNE	SBENT	
7824	020032	023714	002600	CMP	ONES,(R4)	
7825	020036	001043		BNE	SBENT	
7826						
7827	020040	104503		CLR1CSR		;CLEAR 1 SELECTED CSR
7828	020042	005011		CLR	(R1)	
7829	020044	012714	100000	MOV	#BIT15,(R4)	
7830	020050	005711		TST	(R1)	
7831	020052	001035		BNE	SBENT	
7832	020054	022714	100000	CMP	#BIT15,(R4)	
7833	020060	001032		BNE	SBENT	
7834						
7835	020062	104510		TSTREAD		;TEST LOC (R1) & TST FOR SBE (WITHOUT FETCHES)
7836	020064			IF #BIT15!BIT4 SET.IN CSR		
7837	020074			SET SKPERR		;DISABLE ERRGEN'S ERROR PRINTOUT
7838	020102	104512		ERRGEN		
7839	020104	013700	002454	MOV	ERRADD,R0	
7840	020110	072027	177774	ASH	#-4,R0	
7841	020114	042700	177600	BIC	#^C177,R0	
7842	020120			IF BANK EQ R0 THEN GOTO SBENT		
7843	020126			END; OF IF #BIT15		
7844						
7845	020126	104417		KERNEL		;ENTER KERNEL MODE
7846	020130	104473		ECC1INIT		;INITIALIZE 1 SELECTED CSR
7847	020132	104423		CACHON		;TURN CACHE ON
7848	020134			POP	R4,R1,R0	;POP R0,R1 & R4 FROM STACK
7849	020142			\$RETURN	NOERROR	
7850						
7851	020146	104503		SBENT: CLR1CSR		;CLEAR 1 SELECTED CSR
7852	020150			CLEAR	(R1),(R4)	
7853	020154	104417		KERNEL		;ENTER KERNEL MODE
7854	020156	104473		ECC1INIT		;INITIALIZE 1 SELECTED CSR
7855	020160	104423		CACHON		;TURN CACHE ON
7856	020162			POP	R4,R1,R0	;POP R0,R1 & R4 FROM STACK
7857	020170			\$RETURN	ERROR	
7858				.DSABL	LSB	

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 215  
CHECK FOR SBE FREE LOCATIONS

7861 020174

CSRCASE:SUBTST &lt;&lt;CSR PATTERN CASE STATEMENT&gt;&gt;

\*\*\*\*\*  
:SUBTEST CSR PATTERN CASE STATEMENT  
\*\*\*\*\*

CASE R0

:WARNING IF YOU CHANGE THIS TABLE ALSO  
:CHANGE '\$DDW0' - '\$DDW5' (THE PATTERN BIT MAP)

7862 020174

7863

7864

7865

7866

7867 020204 021750

7868

7869

7870

7871

7872 020206 026554

7873 020210 022350

7874 020212 026614

7875 020214 026264

7876 020216 022616

7877 020220 026336

7878 020222 026406

7879 020224 026450

7880 020226 026514

7881 020230 026654

7882 020232 026714

7883

7884 020234 022046

7885

7886

7887

7888

7889 020236 023522

7890 020240 022102

7891 020242 022160

7892 020244 022264

7893 020246 022440

7894 020250 022516

7895

7896

7897

7898 020252 026760

7899 020254 026760

7900 020256 026760

7901 020260 026760

7902 020262 026760

7903 020264

7904 020272 000207

MKCSRT: :PAT TIME DESCRIPTION  
MT0006 :<1 SEC INITIAL DATA TEST

## MS11-P ECC TESTS

:  
: MT0044 : 1 SEC SHIFTING 1/0'S THROUGH CHECK BITS  
: MT0014 : 1 SEC BASIC DOUBLE ERROR TEST  
: MT0045 :<1 SEC SYNDROMES TO CSR ON DOUBLE BIT ERROR TEST  
: MT0036 : 1 SEC CORRECTION CODE TEST  
: MT0020 : 1 SEC SYNDROMES TO CSR ON SINGLE BIT ERROR TEST  
: MT0037 :<1 SEC ECC DISABLE TEST  
: MT0041 : 1 SEC ADDRESS TO CSR ON DOUBLE BIT ERROR  
: MT0042 :<1 SEC EXTENDED ADDRESS TO CSR ON ERROR TEST  
: MT0043 :<1 SEC WRITE BYTE CLEARS SBE TEST  
: MT0046 : 1 SEC CHECK SINGLE BIT ERRORS WITH ECC DISABLED TEST  
: MT0047 :<1 SEC NO CSR UPDATE ON SBE WITH EXSISTING DBE

BYTE ADDRESSING TEST

## MS11-M ECC TESTS

:  
: MT0025 :<1 SEC INTERRUPT ENABLE TEST  
: MT0011 :<2 SEC CREATE SINGLE BIT ERROR TEST  
: MT0012 :<1 SEC WRITE BYTE CLEARS SBE TEST  
: MT0013 : 1 SEC CREATE DOUBLE BIT ERROR TEST  
: MT0015 : 1 SEC WRITE INHIBIT OF BYTE WITH DBE  
: MT0016 :<1 SEC WRITE INHIBIT OF WORD WITH DBE

MT0999 : 0 SEC NULL TEST  
MT0999 : 0 SEC NULL TEST  
MT0999 : 0 SEC NULL TEST  
MT0999 : 0 SEC NULL TEST  
MT0999 : 0 SEC NULL TEST

END :OF CASE R0  
RETURN

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 217  
CSR PATTERN CASE STATEMENT

7907 020274

MKTEST: SUBTST &lt;&lt;SUBR ECC TEST DISPATCH&gt;&gt;

```
*****
:SUBTEST      SUBR      ECC TEST DISPATCH
*****
```

```
7908 020274
7909 020312 104470
7910 020314
7911 020316 104502
7912 020320
7913 020320 012737 000002 002074
7914 020326 012737 000002 002322
7915 020334 013700 002110
7916 020340 006300
7917 020342
7918 020362 104511
7919 020364
7920 020364 010637 002144
7921 020370 162737 000002 002144
7922 020376 004770 020434
7923 020402
7924 020420 104506
7925 020422
7926 020424 104472
7927 020426
7928 020426 005037 002074
7929 020432 000207
```

```
IF #SWO SET.IN @SWR OR ACTFLAG IS TRUE
  ECCDIS ;DISABLE ERROR CORRECTION
ELSE
  CLRCSR ;CLEAR ALL CSR'S
END ;OF IF
MOV #2,NOPAR ;INDICATE PARITY ACTION
MOV #2,PCBUMP ;TRAPS ADD 2 TO PC
MOV PATTERN,RO ;GET PATTERN NUMBER
ASL RO ;MAKE IT A WORD ADDRESS
IF MKPAT(RO) NE #MT0034 AND MKPAT(RO) NE #MT0999
  INVALDATE ;INVALIDATE BACKGROUND PATTERN ON 'BANK'
END ;OF IF MKPAT(RO)
MOV SP,CTLKVEC ;SAVE VECTOR IN CASE OF ^K
SUB #2,CTLKVEC
CALL @MKPAT(RO) ;INDEX OFF TABLE
IF #SWO SET.IN @SWR OR ACTFLAG IS TRUE
  ENASBE ;TRAP ON SINGLE BIT ERRORS
ELSE
  ECCINIT ;TRAP ON DOUBLE BIT ERRORS (NORMAL)
END ;OF IF #SWO
CLR NOPAR ;INDICATE PARITY ACTION
RETURN
```

:WARNING IF YOU CHANGE THIS TABLE ALSO

:CHANGE '\$DDW0' - '\$DDW5' (THE PATTERN BIT MAP)

:PAT TIME DISCRPTION

MKPAT: :NOTE MT0034 MUST BE FIRST &amp; LAST

```
MT0034 :<1 SEC ;SOFT ERROR - BACKGROUND PATTERN TEST
MT0017 :<1 SEC ;HOLDING 1'S & 0'S TEST
MT0007 :<1 SEC ;ADDRESS BIT TEST
MT0001 :<1 SEC ;ADDRESS TEST
MT0002 :<1 SEC ;COMPLEMENT ADDRESS TEST
MT0004 : 1 SEC ;ROTATING ZEROS TEST
MT0005 : 1 SEC ;ROTATING ONES TEST
MT0021 : 1 SEC ;MARCHING 0'S & 1'S TEST
MT0022 :10 SEC ;REFRESH & SHIFTING DIAGONAL TEST
MT0026 :<1 SEC ;RANDOM DATA TEST
MT0024 :20 SEC ;FAST GALLOPING PATTERN TEST
MT0031 : 3 SEC ;SOB-A-LONG TEST
MT0032 :<1 SEC ;WRITE RECOVERY TEST
MT0033 :35 SEC ;BRANCH GOBBLE TEST
MT0034 :<1 SEC ;SOFT ERROR - BACKGROUND PATTERN TEST
:NOTE MT0034 MUST BE FIRST & LAST
MT0999 : 0 SEC ;NULL TEST
MT0999 : 0 SEC ;NULL TEST
MT0999 : 0 SEC ;NULL TEST
MT0999 : 0 SEC ;NULL TEST
MT0999 : 0 SEC ;NULL TEST
MT0999 : 0 SEC ;NULL TEST
MT0999 : 0 SEC ;NULL TEST
MT0999 : 0 SEC ;NULL TEST
MT0999 : 0 SEC ;NULL TEST
MT0999 : 0 SEC ;NULL TEST
```

```
7930
7931
7932
7933
7934 020434
7935 020434 026000
7936 020436 022574
7937 020440 022004
7938 020442 020760
7939 020444 021100
7940 020446 021472
7941 020450 021614
7942 020452 022706
7943 020454 023160
7944 020456 023600
7945 020460 023256
7946 020462 025070
7947 020464 025260
7948 020466 025612
7949 020470 026000
7950
7951 020472 026760
7952 020474 026760
7953 020476 026760
7954 020500 026760
7955 020502 026760
7956 020504 026760
7957 020506 026760
7958 020510 026760
7959 020512 026760
```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 219  
SUBR ECC TEST DISPATCH

7962 020514

MJTEST: SUBTST &lt;&lt;SUBR PARITY TEST DISPATCH&gt;&gt;

\*\*\*\*\*  
:SUBTEST SUBR PARITY TEST DISPATCH  
\*\*\*\*\*

7963 020514 012737 000002 002074  
7964 020522 012737 000002 002322  
7965 020530 012737 060000 002406  
7966 020536 012737 060002 002410  
7967 020544 013700 002110  
7968 020550 006300  
7969 020552  
7970 020572 104511  
7971 020574  
7972 020574 010637 002144  
7973 020600 162737 000002 002144  
7974 020606 004770 020620  
7975 020612 005037 002074  
7976 020616 006207

MOV #2,NOPAR ;INDICATE PARITY ACTION  
MOV #2,PCBUMP ;TRAPS ADD 2 TO PC  
MCV #FIRST,TESTADD  
MOV #FIRST+2,TESTADD+2  
MOV PATTERN,R0 ;GET PATTERN NUMBER  
ASL R0 ;MAKE IT A WORD ADDRESS  
IF MJPAT(R0) NE #MT0034 AND MJPAT(R0) NE #MT0999  
INVALIDATE ;INVALIDATE BACKGROUND PATTERN ON 'BANK'  
END ;OF IF MJPAT(R0)  
MOV SP,CTLKVEC ;SAVE VECTOR IN CASE OF ^K  
SUB #2,CTLKVEC  
CALL @MJPAT(R0) ;INDEX OFF TABLE  
CLR NOPAR ;INDICATE PARITY ACTION  
RETURN

;WARNING IF YOU CHANGE THIS TABLE ALSO  
;CHANGE '\$DDW0' - '\$DDW5' (THE PATTERN BIT MAP)

	:PAT	TIME	DISCRIPTION
MJPAT:	:NOTE MT0034 MUST BE FIRST & LAST		
	MT0034	:<1 SEC	:SOFT ERROR - BACKGROUND PATTERN TEST
	MT0006	:<1 SEC	:INITIAL DATA TEST
	MT0017	:<1 SEC	:HOLDING 1'S & 0'S TEST
	MT0007	:<1 SEC	:ADDRESS BIT TEST
	MT0001	:<1 SEC	:ADDRESS TEST
	MT0002	:<1 SEC	:COMPLEMENT ADDRESS TEST
	MT0003	:1 SEC	:3 XOR 9 WORST CASE NOISE TEST
	MT0004	:1 SEC	:ROTATING ZEROS TEST
	MT0005	:1 SEC	:ROTATING ONES TEST
	MT0021	:1 SEC	:MARCHING 0'S & 1'S TEST
	MT0035	:<1 SEC	:WORSE CASE NOISE PARITY TEST
	MT0022	:10 SEC	:REFRESH TEST
	MT0023	:10 SEC	:SHIFTING DIAGONAL TEST
	MT0026	:<1 SEC	:RANDOM DATA TEST
	MT0024	:20 SEC	:FAST GALLOPING PATTERN TEST
	MT0031	:3 SEC	:SOB-A-LONG TEST
	MT0032	:<1 SEC	:WRITE RECOVERY TEST
	MT0033	:35 SEC	:BRANCH GOBBLE TEST
	MT0034	:<1 SEC	:SOFT ERROR - BACKGROUND PATTERN TEST
	:NOTE MT0034 MUST BE FIRST & LAST		
	MT0999	:0 SEC	:NULL TEST
	MT0999	:0 SEC	:NULL TEST
	MT0999	:0 SEC	:NULL TEST
	MT0999	:0 SEC	:NULL TEST
	MT0999	:0 SEC	:NULL TEST
	MT0999	:0 SEC	:NULL TEST

7981  
7982 020620  
7983 020620 026000  
7984 020622 021750  
7985 020624 022574  
7986 020626 022004  
7987 020630 020760  
7988 020632 021100  
7989 020634 021240  
7990 020636 021472  
7991 020640 021614  
7992 020642 022706  
7993 020644 026152  
7994 020646 023160  
7995 020650 023212  
7996 020652 023600  
7997 020654 023256  
7998 020656 025070  
7999 020660 025260  
8000 020662 025612  
8001 020664 026000  
8002  
8003 020666 026760  
8004 020670 026760  
8005 020672 026760  
8006 020674 026760  
8007 020676 026760

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 220  
PATTERNS

8009  
8010  
8011  
8012 020700

## .SBTTL PATTERNS

```
.SBTTL MEMORY TEST SETUP ROUTINES
MT0000: SUBTST <<MT0000 SETUP DATA PATTERN TEST>>
*****
;*SUBTEST MT0000 SETUP DATA PATTERN TEST
*****
CLR REALPAT ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
MOV #FIRST,R0
MOV #SIZE,R1
CALL REGCOPY
CMP #1,PROTYP ;ARE WE ON AN 11/44?
BEQ 1$ ;BRANCH IF YES
MOV #MTP000,SUPDOADD ;ELSE DO PATTERN IN MAIN MEMORY
CALL SUPDO3
RETURN
1$: BMOV #MTP000
CALL SUPDO1 ;DO IT IN SUPERVISOR MODE
RETURN
```

8013 020700 005037 002274  
8014 020704 012700 060000  
8015 020710 012701 040000  
8016 020714 004737 041064  
8017 020720 022737 000001 003752  
8018 020726 001406  
8019 020730 012737 027400 002260  
8020 020737 004737 027206  
8021 020740 000207  
8022 020744  
8023 020752 004737 027030  
8024 020756 000207  
8025 020760

```
MT0001: SUBTST <<MT0001 SETUP ADDRESS TEST>>
*****
;*SUBTEST MT0001 SETUP ADDRESS TEST
*****
MOV #1,REALPAT ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
MOV #FIRST,R0
MOV #SIZE,R1
TST NOSUPER
BNE 2$
CMP SIPAR5,SIPAR6
BNE 4$
BR 3$
2$: CMP UIPAR5,UIPAR6
BNE 4$
3$: MOV #30000,R1
4$: CLR R2
CALL REGCOPY
CMP #1,PROTYP ;IS THIS AN 11/44?
BEQ 1$ ;BRANCH IF IT IS
MOV #MTP001,SUPDOADD ;SET UP CALLING ADDRESS
CALL SUPDO3
RETURN
1$: BMOV #MTP001
CALL SUPDO1 ;DO IT IN SUPERVISOR MODE
RETURN
```

8026 020760 012737 000001 002274  
8027 020766 012700 060000  
8028 020772 012701 040000  
8029 020776 005737 002452  
8030 021002 001005  
8031 021004 023737 172252 172254  
8032 021012 001007  
8033 021014 000404  
8034 021016 023737 177652 177654  
8035 021024 001002  
8036 021026 012701 030000  
8037 021032 005002  
8038 021034 004737 041064  
8039 021040 022737 000001 003752  
8040 021046 001406  
8041 021050 012737 027424 002260  
8042 021056 004737 027206  
8043 021062 000207  
8044 021064  
8045 021072 004737 027030  
8046 021076 000207  
8047 021100

```
MT0002: SUBTST <<MT0002 SETUP COMPLEMENT ADDRESS TEST>>
*****
;*SUBTEST MT0002 SETUP COMPLEMENT ADDRESS TEST
*****
MOV #2,REALPAT ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
MOV #LAST+2,R0
MOV #SIZE,R1
MOV #FIRST,R4
MOV #100001,R5
TST NOSUPER
BNE 2$
CMP SIPAR5,SIPAR6
BNE 4$
```

8048 021100 012737 000002 002274  
8049 021106 012700 160000  
8050 021112 012701 040000  
8051 021116 012704 060000  
8052 021122 012705 100001  
8053 021126 005737 002452  
8054 021132 001005  
8055 021134 023737 172252 172254  
8056 021142 001013

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 220-1  
 MT0002 SETUP COMPLEMENT ADDRESS TEST

8057	021144	000404				BR	3\$	
8058	C 1146	023737	177652	177654	2\$:	CMP	UIPAR5,UIPAR6	
8059	021154	001006				BNE	4\$	
8060	021156	012701	030000		3\$:	MOV	#30000,R1	
8061	021162	012700	140000			MOV	#140000,R0	
8062	021166	012705	120001			MOV	#120001,R5	
8063	021172	012702	000001		4\$:	MOV	#1,R2	
8064	021176	010103				MOV	R1,R3	
8065	021200	022737	000001	003752		CMP	#1,PROTYP	;IS THIS AN 11/44?
8066	021206	001406				BEQ	1\$	;BRANCH IF TRUE
8067	021210	012737	027456	002260		MOV	#MTP002,SUPDOADD	;SET UP CALLING ADDRESS
8068	C21216	004737	027206			CALL	SUPD03	
8069	021222	000207				RETURN		
8070	021224				1\$:	BMOV	MTP002	
8071	021232	004737	027030			CALL	SUPD01	
8072	021236	000207				RETURN		

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 222  
MT0002 SETUP COMPLEMENT ADDRESS TEST

8075 021240

MT0003: SUBTST &lt;&lt;MT0003 SETUP 3 XOR 9 WORST CASE NOISE TEST&gt;&gt;

\*\*\*\*\*  
:SUBTEST MT0003 SETUP 3 XOR 9 WORST CASE NOISE TEST  
\*\*\*\*\*

8076 021240					IF EUFLAG IS TRUE THEN \$RETURN	
8077 021250	012737	000003	002274		MOV #3,REALPAT	;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
8078 021256	005037	002322			CLR PCBUMP	;TRAPS DO NOT ADD TO PC
8079 021262	004737	04074		1\$:	CALL FLIPJARN	;SETUP WARNING CONSTANTS & R2
8080 021266	012701	060000		2\$:	MOV #FIRST,R1	;R1 <-- STARTING ADDRESS
8081 021272	012703	020000			MOV #20000,R3	
8082 021276	072527	177770			ASH #8,R3	;R3 <-- R3 / 256.
8083 021302	012702	000004			MOV #4,R2	;SMALL LOOP SIZE
8084 021306	012705	000100			MOV #64,R5	;MEDIUM LOOP SIZE
8085 021312	022737	000001	003752		CMP #1,PROTYP	;IS THIS AN 11/44?
8086 021320	001415				BEQ 3\$	;BRANCH IF IT IS
8087 021322	104415				SAVREG	
8088 021324	012737	027510	002260		MOV #MTPA03,SUPDOADD	
8089 021332	004737	027206			CALL SUPD03	;DO IT IN MAIN MEMORY
8090 021336	104416				RESREG	
8091 021340	012737	027550	002260		MOV #MTPB03,SUPDOADD	
8092 021346	004737	027222			CALL SUPD04	
8093 021352	000442				BR 4\$	
8094 021354				3\$:	BMOV MTPA03	
8095 021362	104415				SAVREG	
8096 021364	004737	027030			CALL SUPD01	
8097 021370					BMOV MTPB03	
8098 021376					BMOV MTPC03,KDPA0,8.	
8099 021410					BMOV MTPD03,SDPA0,8.	
8100 021422	012737	172360	177642		MOV #KDPA0,UIPA1	;SET UP PAR LINKS
8101 021430	012737	172260	172374		MOV #SDPA0,KDPA6	
8102 021436	012737	177644	172276		MOV #UIPA2,SDPA7	
8103 021444	012737	001032	172272		MOV #1032,SDPA5	;CHANGE INST TO BR .+66 (BR TO KDPA1)
8104 021452	104416				RESREG	
8105 021454	004737	027044			CALL SUPD02	
8106 021460	022737	000003	002602	4\$:	CMP #3,FLIPLOC	;DONE WITH 4 PATTERNS
8107						;[(0,177777):(177777,0):(401,177777):(177777,401)]?
8108 021466	001275				BNE 1\$	;NO - LOOP
8109 021470	000207				RETURN	
8110						
8111 021472						

MT0004: SUBTST &lt;&lt;MT0004 SETUP ROTATING ZEROS TEST&gt;&gt;

\*\*\*\*\*  
:SUBTEST MT0004 SETUP ROTATING ZEROS TEST  
\*\*\*\*\*

8112 021472	012737	000004	002274		MOV #4,REALPAT	;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
8113 021500	012737	000004	002322		MOV #4,PCBUMP	;TRAPS ADD 4 TO PC
8114 021506	013702	002600			MOV ONES,R2	
8115 021512	004737	041224			CALL BACKGND	;WRITE BACKGROUND OF ONES
8116 021516	012700	060000			MOV #FIRST,R0	
8117 021522	012701	040000			MOV #SIZE,R1	
8118 021526	022737	000001	003752		CMP #1,PROTYP	;IS THIS AN 11/44?
8119 021534	001406				BEQ 1\$	;BRANCH IF IT IS
8120 021536	012737	027646	002260		MOV #MTPA04,SUPDOADD	;SET UP LINKS
8121 021544	004737	027222			CALL SUPD04	
8122 021550	000207				RETURN	
8123 021552				1\$:	BMOV MTPA04	
8124 021560					BMOV MTPB04,KDPA0,8.	
8125 021572	012737	172360	177652		MOV #KDPA0,UIPA5	



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 222-1  
 MT0004 SETUP ROTATING ZEROS TEST

8126 021600 012737 177654 172376  
 8127 021606 004737 027044  
 8128 021612 000207  
 8129 021614

```

      MOV      #UIPAR6,KDPAR7
      CALL     SUPD02
      RETURN
MT0005: SUBTST <<MT0005      SETUP ROTATING ONES TEST>>
:*****
:*SUBTEST      MT0005  SETUP ROTATING ONES TEST
:*****
      MOV      #5,REALPAT      ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
      MOV      #4,PCBUMP      ;TRAPS ADD 4 TO PC
      CLR      R2
      CALL     BACKGND      ;WRITE BACKGROUND OF ZEROS
      MOV      #FIRST,R0
      MOV      #SIZE,R1
      CMP      #1,PROTYP      ;IS THIS AN 11/44?
      BEQ      1$            ;BRANCH IF IT IS
      MOV      #MTP005,SUPDOADD ;SET UP LINKS
      MOV      #MTP005+14,MTPB04+16
      CALL     SUPD04
      MOV      #MTPA04+14,MTPB04+16 ;RESET TEST'S ORIGINAL VALUE
      RETURN
1$:    BMOV     MTP005
      BMOV     MTPB04,KDPAR0,8.
      MOV      #KDPAR0,UIPAR5
      MOV      #UIPAR6,KDPAR7
      CALL     SUPD02
      RETURN

```

8130 021614 012737 000005 002274  
 8131 021622 012737 000004 002322  
 8132 021630 005002  
 8133 021632 004737 041224  
 8134 021636 012700 060000  
 8135 021642 012701 040000  
 8136 021646 022737 000001 003752  
 8137 021654 001414  
 8138 021656 012737 027722 002260  
 8139 021664 012737 027736 027720  
 8140 021672 004737 027222  
 8141 021676 012737 027662 027720  
 8142 021704 000207  
 8143 021706  
 8144 021714  
 8145 021726 012737 172360 177652  
 8146 021734 012737 177654 172376  
 8147 021742 004737 027044  
 8148 021746 000207

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 224  
 MT0005 SETUP ROTATING CNES TEST

8151 021750

MT0006: SUBTST <<MT0006 SETUP INITIAL DATA TEST>>

\*\*\*\*\*  
 :\*SUBTEST MT0006 SETUP INITIAL DATA TEST  
 :\*\*\*\*\*

8152 021750 012737 000006 002274  
 8153 021756 012737 000004 002322  
 8154 021764 012701 002406  
 8155 021770 012737 027756 002260  
 8156 021776 004737 027206  
 8157 022002 000207  
 8158 022004

MOV #6,REALPAT ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY  
 MOV #4,PCBUMP ;TRAPS ADD 4 TO PC  
 MOV #TESTADD,R1  
 MOV #MTP006,SUPDOADD  
 CALL SUPDO3 ;DO IT IN SUPERVISOR MODE  
 RETURN

MT0007: SUBTST <<MT0007 SETUP ADDRESS BIT TEST>>

\*\*\*\*\*  
 :\*SUBTEST MT0007 SETUP ADDRESS BIT TEST  
 :\*\*\*\*\*

8159 022004 012737 000007 002274  
 8160 022012 005002  
 8161 022014 004737 041224  
 8162 022020 012701 060000  
 8163 022024 012702 000001  
 8164 022030 050201  
 8165 022032 012737 030156 002260  
 8166 022040 004737 027206  
 8167 022044 000207  
 8168 022046

MOV #7,REALPAT ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY  
 CLR R2  
 CALL BACKGND ;OF ZEROS  
 MOV #FIRST,R1  
 MOV #1,R2  
 BIS R2,R1  
 MOV #MTP007,SUPDOADD  
 CALL SUPDO3 ;DO IT IN SUPERVISOR MODE  
 RETURN

MT0010: SUBTST <<MT0010 SETUP BYTE ADDRESSING TEST>>

\*\*\*\*\*  
 :\*SUBTEST MT0010 SETUP BYTE ADDRESSING TEST  
 :\*\*\*\*\*

8169 022046 012737 000010 002274  
 8170 022054 012737 000004 002322  
 8171 022062 013704 002406  
 8172 022066 012737 030256 002260  
 8173 022074 004737 027206  
 8174 022100 000207

MOV #10,REALPAT ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY  
 MOV #4,PCBUMP ;TRAPS ADD 4 TO PC  
 MOV TESTADD,R4  
 MOV #MTP010,SUPDOADD  
 CALL SUPDO3 ;DO IT IN SUPERVISOR MODE  
 RETURN

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 226  
 MT0010 SETUP BYTE ADDRESSING TEST

8177 022102

MT0011: SUBTST &lt;&lt;MT0011 SETUP CREATE SINGLE BIT ERROR TEST&gt;&gt;

\*\*\*\*\*  
 :\*SUBTEST MT0011 SETUP CREATE SINGLE BIT ERROR TEST  
 :\*\*\*\*\*

8178 022102

IF ACTFLAG IS TRUE OR APTFLAG IS TRUE

8179 022116

IF \$PASS NE #0 THEN \$RETURN

8180 022126

END; OF IF ACTFLAG

8181 022126

IF PMEMFLG IS TRUE THEN \$RETURN ;EXIT IF NOT MS11-M

8182 022136

012737 000011 002274

MOV #11,REALPAT

;SETUP PATTERN NUMBER FOR TYPEOUT &amp; DISPLAY

8183 022144

012737 030364 002260

MOV #MTP011,SUPDOADD

8184 022152

004737 027206

CALL SUPD03

;DO IT IN SUPERVISOR MODE

8185 022156

000207

RETURN

8186 022160

MT0012: SUBTST &lt;&lt;MT0012 SETUP WRITE BYTE CLEARS SBE TEST&gt;&gt;

\*\*\*\*\*  
 :\*SUBTEST MT0012 SETUP WRITE BYTE CLEARS SBE TEST  
 :\*\*\*\*\*

8187 022160

IF ACTFLAG IS TRUE OR APTFLAG IS TRUE

8188 022174

IF \$PASS NE #0 THEN \$RETURN

8189 022204

END; OF IF ACTFLAG

8190 022204

IF PMEMFLG IS TRUE THEN \$RETURN ;IS THIS A MS11-M?

8191 022214

012737 000012 002274

MOV #12,REALPAT

;SETUP PATTERN NUMBER FOR TYPEOUT &amp; DISPLAY

8192 022222

013700 002102

MOV BANKINDEX,R0

8193 022226

IF #BIT12 SET IN CONFIG+2(R0)

8194 022236

012705 040000

MOV #40000,R5

8195 022242

ELSE

8196 022244

012705 000002

MOV #2,R5

8197 022250

END; OF IF #BIT12

8198 022250

012737 031162 002260

MOV #MTP012,SUPDOADD

8199 022256

004737 027206

CALL SUPD03

;DO IT IN SUPERVISOR MODE

8200 022262

000207

RETURN

8201 022264

MT0013: SUBTST &lt;&lt;MT0013 SETUP CREATE DOUBLE BIT ERROR TEST&gt;&gt;

\*\*\*\*\*  
 :\*SUBTEST MT0013 SETUP CREATE DOUBLE BIT ERROR TEST  
 :\*\*\*\*\*

8202 022264

IF ACTFLAG IS TRUE OR APTFLAG IS TRUE

8203 022300

IF \$PASS NE #0 THEN \$RETURN

8204 022310

END; OF IF ACTFLAG

8205 022310

IF PMEMFLG IS TRUE THEN \$RETURN ;EXIT IF NOT MS11-M

8206 022320

012737 000013 002274

MOV #13,REALPAT

;SETUP PATTERN NUMBER FOR TYPEOUT &amp; DISPLAY

8207 022326

012737 031550 002260

MOV #MTP013,SUPDOADD

8208 022334

012737 000003 002074

MOV #3,NOPAR

;INDICATE PARITY ACTION

8209 022342

004737 027206

CALL SUPD03

;DO IT IN SUPERVISOR MODE

8210 022346

000207

RETURN

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 227  
 MT0013 SETUP CREATE DOUBLE BIT ERROR TEST

8212 022350

MT0014: SUBTST &lt;&lt;MT0014 SETUP BASIC DOUBLE BIT ERROR TEST&gt;&gt;

;\*\*\*\*\*

;\*SUBTEST MT0014 SETUP BASIC DOUBLE BIT ERROR TEST

;\*\*\*\*\*

8213 022350

IF ACTFLAG IS TRUE OR APTFLAG IS TRUE

8214 022364

IF \$PASS NE #0 THEN \$RETURN

8215 022374

END; OF IF ACTFLAG

8216 022374

IF PMEMFLG IS FALSE THEN \$RETURN ;EXIT IF NOT MS11-P

8217 022404

012737 000014 002274

MOV #14,REALPAT

;SETUP PATTERN NUMBER FOR TYPEOUT &amp; DISPLAY

8218 022412

004737 046706

CALL MAPKERNAL

;MAP KERNAL SPACE

8219 022416

LET R1 := #100000

;SETUP TEST ADDRESS

8220 022422

004737 041324

CALL GETCSR

;GET CSR INFO FROM CONFIGURATION TABLE

8221 022426

004737 032264

CALL MTP014

;DO BASIC DCUBLE BIT ERROR TEST

8222 022432

004737 046774

CALL UNMAP

;UNMAP KERNAL SPACE

8223 022436

000207

RETURN

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 229  
 MT0014 SETUP BASIC DOUBLE BIT ERROR TEST

8226 022440

MT0015: SUBTST <<MT0015 SETUP WRITE INHIBIT OF BYTE WITH DBE>>  
 :\*\*\*\*\*  
 :\*SUBTEST MT0015 SETUP WRITE INHIBIT OF BYTE WITH DBE  
 :\*\*\*\*\*

8227 022440

IF ACTFLAG IS TRUE OR APTFLAG IS TRUE

8228 022454

IF \$PASS NE #0 THEN \$RETURN

8229 022464

END :OF IF ACTFLAG

8230 022464

IF PMEMFLG IS TRUE THEN \$RETURN ;EXIT IF NOT MS11-M

8231 022474

012737 000015 002274

MOV #15,REALPAT

;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY

8232 022502

012737 032510 002260

MOV #MTP015,SUPDOADD

8233 022510

004737 027206

CALL SUPDO3

;DO IT IN SUPERVISOR MODE

8234 022514

000207

RETURN

8235 022516

MT0016: SUBTST <<MT0016 SETUP WRITE INHIBIT OF WORD WITH DBE>>  
 :\*\*\*\*\*  
 :\*SUBTEST MT0016 SETUP WRITE INHIBIT OF WORD WITH DBE  
 :\*\*\*\*\*

8236 022516

IF PMEMFLG IS TRUE THEN \$RETURN ;EXIT IF NOT MS11-M

8237 022526

IF ACTFLAG IS TRUE OR APTFLAG IS TRUE

8238 022542

IF \$PASS NE #0 THEN \$RETURN

8239 022552

END :OF IF ACTFLAG

8240 022552

012737 000016 002274

MOV #16,REALPAT

;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY

8241 022560

012737 033254 002260

MOV #MTP016,SUPDOADD

8242 022566

004737 027206

CALL SUPDO3

;DO IT IN SUPERVISOR MODE

8243 022572

000207

RETURN

8244 022574

MT0017: SUBTST <<MT0017 SETUP HOLDING 1'S & 0'S>>  
 :\*\*\*\*\*  
 :\*SUBTEST MT0017 SETUP HOLDING 1'S & 0'S  
 :\*\*\*\*\*

8245 022574

012737 000017 002274

MOV #17,REALPAT

;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY

8246 022602

012737 034036 002260

MOV #MTP017,SUPDOADD

8247 022610

004737 027206

CALL SUPDO3

;DO IT IN SUPERVISOR MODE

8248 022614

000207

RETURN

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 231  
 MT0017 SETUP HOLDING 1'S & 0'S

8251 022616

MT0020: SUBTST <<MT0020 SETUP SYNDROMES TO CSR ON SINGLE BIT ERROR>>

\*\*\*\*\*  
 :SUBTEST MT0020 SETUP SYNDROMES TO CSR ON SINGLE BIT ERROR  
 :\*\*\*\*\*

8252 022616

8253 022632

8254 022642

8255 022642

8256 022650

8257 022660

8258 022664

8259 022670

8260 022674

8261 022700

8262 022704

012737 000020 002274

004737 046706

004737 041324

004737 034114

004737 046774

000207

IF ACTFLAG IS TRUE OR APTFLAG IS TRUE

IF \$PASS NE #0 THEN \$RETURN

END: OF IF ACTFLAG

MOV #20,REALPAT

IF PMEMFLG IS FALSE THEN \$RETURN

CALL MAPKERNAL

LET R1 := #100000

CALL GETCSR

CALL MTP020

CALL UNMAP

RETURN

;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY

;EXIT IF NOT MS11-P

;MAP KERNAL SPACE

;SETUP TEST ADDRESS

;GET CSR INFO FROM CONFIGURATION TABLE

;DO SYNDROMES TO CSR ON SINGLE ERROR TEST

;UNMAP KERNAL SPACE

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 232  
 MT0020 SETUP SYNDROMES TO CSR ON SINGLE BIT ERROR

8264 022706

MT0021: SUBTST &lt;&lt;MT0021 SETUP MARCHING 0'S &amp; 1'S TEST&gt;&gt;

\*\*\*\*\*  
 :SUBTEST MT0021 SETUP MARCHING 0'S & 1'S TEST  
 \*\*\*\*\*

8265	022706				SET NOSCOPE	
8266	022714	012737	000021	002274	MOV #21,REALPAT	;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
8267	022722	013702	002616		MOV BAKPAT,R2	
8268	022726	004737	041224		CALL BACKGND	
8269	022732	010203			MOV R2,R3	
8270	022734	000303			SWAB R3	
8271	022736	012701	160000		MOV #LAST+2,R1	
8272	022742	010105			MOV R1,R5	
8273	022744	012704	060000		MOV #FIRST,R4	
8274	022750	022737	000001	003752	CMP #1,PROTYP	;IS THIS AN 11/44?
8275	022756	001441			BEQ 1\$	;BRANCH IF IT IS
8276	022760	022737	000003	003752	CMP #3,PROTYP	;IS THIS AN 11/24?
8277	022766	001407			BEQ 3\$	;BRANCH IF SO
8278	022770	022737	000007	002100	CMP #7,BANK	
8279	022776	001003			BNE 3\$	
8280	023000	012701	140000		MOV #140000,R1	
8281	023004	010105			MOV R1,R5	
8282	023006	012737	034430	002260	3\$: MOV #MTPA21,SUPDOADD	
8283	023014	004737	027206		CALL SUPD03	
8284	023020	012737	034460	002260	MOV #MTPB21,SUPDOADD	
8285	023026	004737	027222		CALL SUPD04	
8286	023032	010401			MOV R4,R1	
8287	023034	012737	034514	002260	MOV #MTPC21,SUPDOADD	
8288	023042	004737	027222		CALL SUPD04	
8289	023046	012737	034550	002260	MOV #MTPD21,SUPDOADD	
8290	023054	004737	027222		CALL SUPD04	
8291	023060	000434			BR 2\$	
8292	023062	022737	000177	002100	1\$: CMP #177,BANK	
8293	023070	001003			BNE 4\$	
8294	023072	012701	140000		MOV #140000,R1	
8295	023076	010105			MOV R1,R5	
8296	023100				4\$: BMOV MTPA21	
8297	023106	004737	027030		CALL SUPD01	
8298						
8299	023112				BMOV MTPB21	
8300	023120	004737	027044		CALL SUPD02	
8301						
8302	023124	010401			MOV R4,R1	
8303	023126				BMOV MTPC21	
8304	023134	004737	027044		CALL SUPD02	
8305						
8306	023140				BMOV MTPD21	
8307	023146	004737	027044		CALL SUPD02	
8308	023152	005037	002434		2\$: CLR NOSCOPE	
8309	023156	000207			RETURN	

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 233  
 MT0021 SETUP MARCHING 0'S & 1'S TEST

8311 023160

MT0022: SUBTST <<MT0022 SETUP REFRESH & SHIFTING DIAGONAL TEST>>

\*\*\*\*\*  
 :\*SUBTEST MT0022 SETUP REFRESH & SHIFTING DIAGONAL TEST  
 :\*\*\*\*\*

8312 023160 004737 026774  
 8313 023164  
 8314 023170 012737 000022 002274  
 8315 023176 012737 034600 002260  
 8316 023204 004737 027206  
 8317 023210 000207  
 8318  
 8319 023212

CALL KAMITEST ;CHECK FOR KAMIKAZE MODE  
 ON.ERROR THEN \$RETURN ;IF NOT IN KAMIKAZE MODE RETURN  
 MOV #22,REALPAT ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY  
 MOV #MTP022,SUPDOADD  
 CALL SUPDO3 ;DO IT IN SUPERVISOR MODE  
 RETURN

MT0023: SUBTST <<MT0023 SHIFTING DIAGONAL TEST>>

\*\*\*\*\*  
 :\*SUBTEST MT0023 SHIFTING DIAGONAL TEST  
 :\*\*\*\*\*

8320 023212 004737 026774  
 8321 023216  
 8322 023222 012737 000023 002274  
 8323 023230 012737 034600 002260  
 8324 023236  
 8325 023244 004737 027206  
 8326 023250 005037 002002  
 8327 023254 000207

CALL KAMITEST ;CHECK FOR KAMIKAZE MODE  
 ON.ERROR THEN \$RETURN ;IF NOT IN KAMIKAZE MODE RETURN  
 MOV #23,REALPAT ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY  
 MOV #MTP022,SUPDOADD  
 SET DIAGFLAG ;IDENTIFY DIAGONAL TEST TO MTP022  
 CALL SUPDO3 ;DO IT IN SUPERVISOR MODE  
 CLR DIAGFLAG  
 RETURN



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 234  
MT0023 SHIFTING DIAGONAL TEST

8329 023256

MT0024: SUBTST &lt;&lt;MT0024 SETUP FAST GALLOPING PATTERN TEST&gt;&gt;

\*\*\*\*\*  
:SUBTEST MT0024 SETUP FAST GALLOPING PATTERN TEST  
\*\*\*\*\*

8330	023256	004737	026774		CALL	KAMITEST		:CHECK FOR KAMIKAZE MODE
8331	023262				ON ERROR THEN	\$RETURN		:IF NOT IN KAMIKAZE MODE RETURN
8332	023266				SET	NOSCOPE		
8333	023274	012737	000024	002274	MOV	#24,REALPAT		:SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
8334	023302	013702	002616		MOV	BAKPAT,R2		
8335	023306	004737	041224		CALL	BACKGND		
8336	023312	010203			MOV	R2,R3		
8337	023314	010304			MOV	R3,R4		
8338	023316	000304			SWAB	R4		
8339	023320	012701	060000		MOV	#FIRST,R1		
8340	023324	012705	157776		MOV	#LAST,R5		
8341	023330	022737	000001	003752	CMP	#1,PROTYP		
8342	023336	001417			BEQ	1\$		
8343	023340	022737	000003	003752	CMP	#3,PROTYP		
8344	023346	001406			BEQ	3\$		
8345	023350	022737	000007	002100	CMP	#7,BANK		
8346	023356	001002			BNE	3\$		
8347	023360	012705	137776		MOV	#137776,R5		
8348	023364	104415			3\$: SAVREG			
8349	023366	012737	035314	002260	MOV	#MTPB24,SUPDOADD		
8350	023374	000440			BR	2\$		
8351	023376	022737	000177	002552	1\$: CMP	#177,LASTBANK		
8352	023404	001002			BNE	4\$		
8353	023406	012705	137776		MOV	#137776,R5		
8354	023412	104415			4\$: SAVREG			
8355	023414				BMOV	MTPA24		
8356	023422				BMOV	MTPB24,SDPAR0,8.		
8357	023434				BMOV	MTPC24,KDPAR0,8.		
8358	023446	012737	172260	002260	MOV	#SDPAR0,SUPDOADD		
8359	023454	012737	172260	177676	MOV	#SDPAR0,UDPAR7		:SET UP PAR LINKS
8360	023462	012737	172360	172272	MOV	#KDPAR0,SDPAR5		
8361	023470	012737	177660	172374	MOV	#UDPAR0,KDPAR6		
8362	023476	004737	027222		2\$: CALL	SUPD04		
8363								
8364								
8365	023502	104416						
8366	023504	000302			RESREG			
8367	023506	000303			SWAB	R2		
8368	023510	004737	027222		SWAB	R3		
8369	023514	005037	002434		CALL	SUPD04		
8370	023520	000207			CLR	NOSCOPE		
8371	023522				RETURN			

MT0025: SUBTST &lt;&lt;MT0025 SETUP INTERRUPT ENABLE TEST&gt;&gt;

\*\*\*\*\*  
:SUBTEST MT0025 SETUP INTERRUPT ENABLE TEST  
\*\*\*\*\*

8372	023522							
8373	023536							
8374	023546							
8375	023546							
8376	023556	012737	000025	002274	IF ACTFLAG IS TRUE OR APTFLAG IS TRUE			
8377	023564	012737	035346	002260	IF \$PASS NE #0 THEN \$RETURN			
8378	023572	004737	027206		END ;OF IF ACTFLAG			
8379	023576	000207			IF PMEMFLG IS TRUE THEN \$RETURN		:EXIT IF NOT MS11-P	
					MOV	#25,REALPAT		:SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
					MOV	#MTP025,SUPDOADD		
					CALL	SUPD03		:DO IT IN SUPERVISOR MODE
					RETURN			

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 236  
MT0025 SETUP INTERRUPT ENABLE TEST

8382 023600

MT0026: SUBTST &lt;&lt;MT0026 SETUP RANDOM DATA TEST&gt;&gt;

\*\*\*\*\*  
:SUBTEST MT0026 SETUP RANDOM DATA TEST  
\*\*\*\*\*

8383	023600	012737	000026	002274	MOV	#26,REALPAT	
8384	023606	005037	002322		CLR	PCBUMP	:TRAPS DO NOT ADD TO THE PC
8385	023612	013703	002570		MOV	SEEDLO,R3	:INITIALIZE RANDOM NUMBERS
8386	023616	013702	002560		MOV	SEEDHI,R2	
8387	023622	010305			MOV	R3,R5	
8388	023624	010204			MOV	R2,R4	
8389	023626	012701	060000		MOV	#FIRST,R1	
8390	023632	012700	020000		MOV	#SIZE/2,R0	
8391	023636	022737	000001	003752	CMP	#1,PROTYP	:DO WE HAVE AN 11/44?
8392	023644	001437			BEQ	1\$	:BRANCH IF WE DO
8393	023646	022737	060003	003752	CMP	#3,PROTYP	:11/24?
8394	023654	001406			BEQ	3\$	:BRANCH IF SO
8395	023656	022737	000007	002100	CMP	#7,BANK	
8396	023664	001002			BNE	3\$	
8397	023666	012700	014000		MOV	#14000,R0	
8398	023672	104415			3\$: SAVREG		
8399	023674	012737	036020	036120	MOV	#MTPA26+4,MTPD26+14	
8400	023702	012737	036014	002260	MOV	#MTPA26,SUPDOADD	
8401	023710	004737	027206		CALL	SUPDC3	
8402	023714	005037	036044		CLR	RANODD	:FOR ERROR REPORTING
8403	023720	012737	036034	036120	MOV	#MTPB26+4,MTPD26+14	:SET UP NEXT LINK
8404	023726	012737	036030	002260	MOV	#MTPB26,SUPDOADD	
8405	023734	104416			RESREG		
8406	023736	004737	027206		CALL	SUPDO3	
8407	023742	000452			BR	2\$	
8408	023744	022737	000177	002100	1\$: CMP	#177,BANK	
8409	023752	001002			BNE	4\$	
8410	023754	012700	014000		MOV	#14000,R0	
8411	023760	104415			4\$: SAVREG		
8412	023762				BMOV	MTPA26	:WRITE ROUTINE TO FAST MEMORY
8413	023770				BMOV	MTPC26,KDPA0,8.	:RANDOM SUBPROGRAM TO FAST MEMORY
8414	024002	012737	000730	172376	MOV	#730,KDPA0,8.	:WRITES 'BR .-116' IN (BR SDPA0)
8415	024010				BMOV	MTPD26,SDPA0,8.	:RANDOM SUBSUBPROGRAM TO FAST MEMORY
8416	024022	012737	172360	177642	MOV	#KDPA0,UIPA1	
8417	024030	012737	177644	172274	MOV	#UIPA2,SDPA6	
8418	024036	004737	027030		CALL	SUPDO1	:WRITE RANDOM DATA
8419	024042	005037	036044		CLR	RANODD	:FOR ERROR REPORTING
8420	024046				BMOV	MTPB26	:READ ROUTINE TO FAST MEMORY
8421	024054	012737	172360	177642	MOV	#KDPA0,UIPA1	:SET UP PAR LINK
8422	024062	104416			RESREG		
8423	024064	004737	027030		CALL	SUPDO1	:READ RANDOM DATA
8424	024070	010337	002570		2\$: MOV	R3,SEEDLO	:UPDATE FOR NEW RANDOM NUMBERS
8425	024074	010237	002566		MOV	R2,SEEDHI	
8426	024100	000207			RETURN		

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 238  
MT0026 SETUP RANDOM DATA TEST

8429 024102

MT0027: SUBTST &lt;&lt;MT0027 UNIQUE BANK TEST&gt;&gt;

\*\*\*\*\*  
:SUBTEST MT0027 UNIQUE BANK TEST  
\*\*\*\*\*

:MAKE SURE THAT EACH BANK CAN HAVE UNIQUE DATA  
:WRITE AND READ THE BANK NUMBER IN EACH BANK (EXCEPT WHERE THE PROGRAM IS)  
MOV #27,REALPAT ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY

CLRCR ;CLEAR CSRS  
CMP #1,PROTYP ;IS THIS AN 11/44?  
BEQ 1\$ ;BRANCH IF TRUE  
MOV #SUPDO3,LINK1 ;SET UP LINK  
BR STAR27 ;BRANCH TO RUN

1\$: BMOV MTP034  
WARN7: MOV #UIPAR3,SUPDOADD  
MOV #SUPDO1,LINK1 ;SET UP LINK  
SET NOFSMODE

STAR27: FOR I := #1 TO #2  
FOR BANK := #0 TO LASTBANK

CALL EXBANK  
IF ACFLAG IS TRUE AND RRFLAG IS FALSE  
INvalidate ;INvalidate BACKGROUND PATTERN ON 'BANK'

LET R2 := BANK  
MOV #FIRST,R0  
MOV R0,R4  
MOV #SIZE,R1  
MOV R1,R3

IF I EQ #1  
CMP #1,PROTYP  
BEQ 2\$  
MOV #MTP034,SUPDOADD  
2\$: CALL @LINK1

END :OF IF  
IF I EQ #2  
CMP #1,PROTYP  
BEQ 3\$  
MOV #MTP034+6,SUPDOADD

3\$: CALL SUPDO3  
END :OF IF

END :OF IF

END :OF FOR BANK

END :OF FOR I

IF FS7FLAG IS TRUE

CLR NOFSMODE

RETURN

END :OF IF FS7FLAG

FOR I := #1 TO #2

FOR BANK := LASTBANK DOWNT0 #0

CALL EXBANK

IF ACFLAG IS TRUE AND RRFLAG IS FALSE

LET R2 := BANK

COM R2

MOV #FIRST,R0

MOV R0,R4

MOV #SIZE,R1

MOV R1,R3

IF I EQ #1

CMP #1,PROTYP

8430  
8431  
8432 024102 012737 000027 002274  
8433 024110 104502  
8434 024112 022737 000001 003752  
8435 024120 001404  
8436 024122 012737 027206 002516  
8437 024130 000414  
8438 024132  
8439 024140 012737 177646 002260  
8440 024146 012737 027030 002516  
8441 024154  
8442 024162  
8443 024170  
8444 024174 004737 047020  
8445 024200  
8446 024214 104511  
8447 024216  
8448 024222 012700 060000  
8449 024226 010004  
8450 024230 012701 040000  
8451 024234 010103  
8452 024236  
8453 024246 022737 000001 003752  
8454 024254 001403  
8455 024256 012737 036340 002260  
8456 024264 004777 156226  
8457 024270  
8458 024270  
8459 024300 022737 000001 003752  
8460 024306 001403  
8461 024310 012737 036346 002260  
8462 024316 004737 027206  
8463 024322  
8464 024322  
8465 024322  
8466 024336  
8467 024352  
8468 024360 005037 002422  
8469 024364 000207  
8470 024366  
8471 024366  
8472 024374  
8473 024402 004737 047020  
8474 024406  
8475 024422  
8476 024426 005102  
8477 024430 012700 060000  
8478 024434 010004  
8479 024436 012701 040000  
8480 024442 010103  
8481 024444  
8482 024454 022737 000001 003752

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 238-1  
MT0027 UNIQUE BANK TEST

8483 024462 001403  
8484 024464 012737 036340 002260  
8485 024472 004777 156020  
8486 024476  
8487 024476  
8488 024506 022737 000001 003752  
8489 024514 001403  
8490 024516 012737 036346 002260  
8491 024524 004737 027206  
8492 024530  
8493 024530  
8494 024530  
8495 024544  
8496 024560 005037 002422  
8497 024564 000207

BEQ 4\$  
MOV #MTP034,SUPDOADD  
4\$: CALL @LINK1  
END :OF IF  
IF 1 EQ #2  
CMP #1,PROTYP  
BEQ 5\$  
MOV #MTP034+6,SUPDOADD  
5\$: CALL SUPDO3  
END :OF IF  
END :OF IF  
END :CF FOR BANK  
END :OF FOR I  
CLR NOFSMODE  
RETURN

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 240  
MT0027 UNIQUE BANK TEST

8500 024566

MT0030: SUBTST &lt;&lt;MT0030 SETUP FLUSH OUT DBE'S TEST&gt;&gt;

\*\*\*\*\*  
:SUBTEST MT0030 SETUP FLUSH OUT DBE'S TEST  
\*\*\*\*\*

8501 024566 005037 002262  
8502 024572  
8503 024600 012737 000030 002274  
8504 024606 012737 000001 002074  
8505 024614 022737 000001 003752  
8506 024622 001007  
8507 024624  
8508 024632 012737 027030 002516  
8509 024640 000406  
8510 024642 012737 027206 002516 4\$:  
8511 024650 012737 036122 002260  
8512 024656 104470 1\$:  
8513 024660  
8514 024674  
8515 024700 004737 047020  
8516 024704  
8517 024712  
8518 024726 012701 040000  
8519 024732 012700 060000  
8520 024736 004777 155554  
8521 024742  
8522 024742  
8523 024742  
8524 024756  
8525 024764  
8526 024772 104502  
8527 024774 004737 045172  
8528 025000  
8529 025002 104472  
8530 025004  
8531 025020 000207  
8532 025022  
8533 025022 013737 002304 002100  
8534 025030 004737 047020  
8535 025034 004737 024600  
8536 025040 104472  
8537 025042 004737 046056  
8538 025046 000207  
8539 025050  
8540 025050 104472  
8541 025052  
8542 025066 000207

CLR PASFLG  
SET FULLREL  
MTA030: MOV #30,REALPAT ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY  
MOV #1,NOPAR ;INDICATE COUNT PARITY ERRORS  
CMP #1,PROTYP  
BNE 4\$  
BMOV MTP030  
MO, #SUPD01,LINK1  
BR 1\$  
4\$: MOV #SUPD03,LINK1  
MOV #MTP030,SUPDOADD  
1\$: ECCDIS ;DISABLE ERROR CORRECTION  
SET NOFSMODE,NOSCOPE  
FOR BANK := #0 TO LASTBANK  
CALL EXBANK  
IF MKFLAG IS TRUE  
IF ACFLAG IS TRUE AND RRFLAG IS FALSE  
MOV #SIZE,R1  
MOV #FIRST,R0  
CALL @LINK1  
END ;OF IF ACFLAG  
END ;OF IF MKFLAG  
END ;OF FOR  
IF PASFLG IS FALSE  
SET PASFLG  
CLRCRS ;CLEAR CSRS  
CALL RELOCATE  
ON.ERROR  
ECCINIT ;TRAP ON DOUBLE BIT ERRORS (NORMAL)  
CLEAR NOFSMODE,NOSCOPE,FULLREL  
RETURN  
END ;OF ON.ERROR  
MOV NEWBANK,BANK  
CALL EXBANK  
CALL MTA030  
ECCINIT ;TRAP ON DOUBLE BIT ERRORS (NORMAL)  
CALL UNRELOCATE  
RETURN  
END ;OF IF PASFLG  
ECCINIT ;TRAP ON DOUBLE BIT ERRORS (NORMAL)  
CLEAR NOFSMODE,NOSCOPE,FULLREL  
RETURN

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 242  
 MT0030 SETUP FLUSH OUT DBE'S TEST

8545 025070

MT0031: SUBTST &lt;&lt;MT0031 SETUP SOB-A-LONG TEST&gt;&gt;

\*\*\*\*\*  
 ;SUBTEST MT0031 SETUP SOB-A-LONG TEST  
 ;\*\*\*\*\*

8546 025070	004737	026774		CALL	KAMITEST		;CHECK FOR KAMIKAZE MODE
8547 025074				ON.ERROR THEN	\$RETURN		;IF NOT IN KAMIKAZE MODE RETURN
8548 025100				SET	NOSCOPE		
8549 025106	012737	000031	002274	MOV	#31,REALPAT		;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
8550 025114	005037	002074		CLR	NOPAR		;SETUP PARITY ACTION
8551 025120				MAP	BANK		;MAP FIRST SO BLOCK MOVE WORKS
8552 025134				TESTAREA			;ENTER TEST MODE
8553 025142				BMOV	MT0031,FIRST,SOBLENGTH/2		
8554 025154	104417			KERNEL			;ENTER KERNEL MODE
8555 025156	013702	002556		MOV	SOBK,R2		
8556 025162	010200			MOV	R2,R0		
8557 025164	012701	100776		MOV	#100776,R1		
8558 025170	012705	060056		MOV	#FIRST+SOBLENGTH,R5		;COMPLEMENT OF INSTRUCTION 'SOB R0,DOT'
8559 025174	012737	060002	002260	MOV	#FIRST+2,SUPDOADD		
8560 025202	012737	160000	002516	MOV	#LAST+2,LINK1		
8561 025210	005737	002452		TEST	NOSUPER		
8562 025214	001005			BNE	1\$		
8563 025216	023737	172252	172254	CMP	SIPAR5,SIPAR6		
8564 025224	001405			BEQ	2\$		
8565 025226	000407			BR	3\$		
8566 025230	023737	177652	177654	CMP	UIPAR5,UIPAR6		
8567 025236	001003			BNE	3\$		
8568 025240	012737	140000	002516	MOV	#140000,LINK1		
8569 025246	004737	027222		CALL	SUPD04		
8570 025252	005037	002434		CLR	NOSCOPE		
8571 025256	000207			RETURN			

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 244  
MT0031 SETUP SOB-A-LONG TEST

8574 025260

MT0032: SUBTST &lt;&lt;MT0032 SETUP WRITE RECOVERY TEST&gt;&gt;

\*\*\*\*\*  
:SUBTEST MT0032 SETUP WRITE RECOVERY TEST  
\*\*\*\*\*

8575	025260	004737	026774		CALL	KAMITEST		;CHECK FOR KAMIKAZE MODE
8576	025264				ON.ERROR	THEN \$RETURN		;IF NOT IN KAMIKAZE MODE RETURN
8577	025270				SET	NOSCOPE		
8578	025276	012737	000032	002274	MOV	#32,REALPAT		;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
8579	025304	005037	002074		CLR	NOPAR		;SETUP PARITY ACTION
8580	025310				MAP	BANK		;MAP FIRST SO THAT THE BLOCK MOVE WORKS
8581	025324	012700	010247		MOV	#10247,R0		;OP CODE OF INSTRUCTION 'MOV R2,-(PC)'
8582	025330	012701	177667		MOV	#177667,R1		;OP CODE OF COMPLEMENT OF INSTRUCTION 'JMP (R0)'
8583	025334	012702	020000		MOV	#SIZE/2,R2		;USED FOR 1/2 BANK LOOP
8584	025340	010237	002516		MOV	R2,LIN		
8585	025344	012703	060000		MOV	#FIRST,R3		
8586	025350	012704	160000		MOV	#LAST+2,R4		
8587	025354	005037	002520		CLR	LINK2		
8588	025360	005737	002452		TST	NOSUPER		
8589	025364	001005			BNE	1\$		
8590	025366	023737	172252	172254	CMP	SIPAR5,SIPAR6		
8591	025374	001405			BEQ	2\$		
8592	025376	000415			BR	3\$		
8593	025400	023737	177652	177654	CMP	UIPAR5,UIPAR6	1\$:	
8594	025406	001011			BNE	3\$		
8595	025410	012704	140000		MOV	#140000,R4	2\$:	
8596	025414	012702	014000		MOV	#14000,R2		
8597	025420	010237	002516		MOV	R2,LINK1		
8598	025424	012737	000001	002520	MOV	#1,LINK2		
8599								
8600	025432				3\$:	TESTAREA		;ENTER TEST MODE
8601						;MOVE TEST TO MEMORY UNDER TEST		
8602	025440	010023			4\$:	MOV	R0,(R3)+	
8603	025442	010144				MOV	R1,-(R4)	
8604	025444	077203				SQB	R2,4\$	
8605								
8606	025446	022737	000001	003752	CMP	#1,PROTYP		
8607	025454	001003			BNE	5\$		
8608						;MOVE LAST PART OF TEST TO FASTCITY		
8609	025456				BMOV	MTP032		
8610	025464	104417			5\$:	KERNEL		;ENTER KERNEL MODE
8611								
8612	025466	012702	005141		MOV	#5141,R2		;OP CODE OF INSTRUCTION 'COM -(R1)'
8613	025472	012700	025610		MOV	#10\$,R0		;ADDRESS TO RETURN TO IN R0
8614	025476	012701	160000		MOV	#LAST+2,R1		;TOP OF BANK
8615	025502	012737	060000	002260	MOV	#FIRST,SUPDOADD		
8616	025510	005737	002520		TST	LINK2		
8617	025514	001402			BEQ	6\$		
8618	025516	012701	140000		MOV	#140000,R1		
8619	025522	004737	027222		6\$:	CALL	SUPD04	
8620	025526	012703	020000		MOV	#SIZE/2,R3		
8621	025532	012705	000110		MOV	#110,R5		
8622	025536	012704	060000		MOV	#FIRST,R4		
8623	025542	005737	002520		TST	LINK2		
8624	025546	001402			BEQ	7\$		
8625	025550	012703	014000		MOV	#14000,R3		
8626	025554	022737	000001	003752	7\$:	CMP	#1,PROTYP	
8627	025562	001406			BEQ	8\$		

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 244-1  
MT0032 SETUP WRITE RECOVERY TEST

8628	025564	012737	036210	002260		MOV	#MTP032,SUPDOADD
8629	025572	004737	027222			CALL	SUPDO4
8630	025576	000402				BR	9\$
8631	025600	004737	027044		8\$:	CALL	SUPDO2
8632	025604	005037	002434		9\$:	CLR	NOSCOPE
8633	025610	000207			10\$:	RETURN	
8634							
8635							

;THIS RETURN ACTS AS A NORMAL RETURN FROM MT0032  
;ALSO A RETURN FROM THE "CALL SUPDO4" ABOVE



CZMSPA0 MS11-L/M/P MEMOF DIAG. MACRO M1113 26-APR-82 09:41 PAGE 246  
MT0032 SETUP WRITE RECOVERY TEST

8638 025612

MT0033: SUBTST &lt;&lt;MT0033 SETUP BRANCH GOBBLE TEST&gt;&gt;

\*\*\*\*\*  
:SUBTEST MT0033 SETUP BRANCH GOBBLE TEST  
\*\*\*\*\*

8639 025612 004737 026774

CALL KAMITEST ;CHECK FOR KAMIKAZE MODE  
ON.ERROR THEN \$RETURN ;IF NOT IN KAMIKAZE MODE RETURN

8640 025616

SET NOSCOPE

8641 025622

8642 025630 012737 000033 002274

MOV #33,REALPAT

;SETUP PATTERN NUMBER FOR TYPEOUT &amp; DISPLAY

8643 025636 005037 002074

CLR NOPAR

;SETUP PARITY ACTION

8644 025642

MAP BANK

;MAP FIRST SO THAT BLOCK MOVE WORKS

8645

8646 025656

TESTAREA

;ENTER TEST MODE

8647 025664

BMOV MTP033,FIRST,GBLENGTH/2

8648 025676 104417

KERNEL

;ENTER KERNEL MODE

8649

8650 025700 012705 060076

MOV #FIRST+GBLENGTH,R5

8651 025704 012737 060004 002260

MOV #FIRST+4,SUPDOADD

8652 025712 012701 060002

MOV #FIRST+2,R1

8653 025716 012702 060003

MOV #FIRST+3,R2

8654 025722 012737 160000 002516

MOV #LAST+2,LINK1

8655 025730 005737 002452

TST NOSUPER

8656 025734 001005

BNE 1\$

8657 025736 023737 172252 172254

CMP SIPAR5,SIPAR6

8658 025744 001405

BEQ 2\$

8659 025746 000407

BR 3\$

8660 025750 023737 177652 177654 1\$:

CMP UIPAR5,UIPAR6

8661 025756 001003

BNE 3\$

8662 025760 012737 140000 002516 2\$:

MOV #140000,LINK1

8663

8664 025766 004737 027222 3\$:

CALL SUPD04

8665 025772 005037 002434

CLR NOSCOPE

8666 025776 000207

RETURN

8667

8668 026000

MT0034: SUBTST &lt;&lt;MT0034 SOFT ERROR - BACKGROUND PATTERN TEST&gt;&gt;

\*\*\*\*\*  
:SUBTEST MT0034 SOFT ERROR - BACKGROUND PATTERN TEST  
\*\*\*\*\*

8669 026000 012737 000034 002274

MOV #34,REALPAT

8670 026006 012700 060000

MOV #FIRST,R0

8671 026012 012701 040000

MOV #SIZE,R1

8672 026016 013702 002604

MOV SOFTPAT,R2

8673 026022 010103

MOV R1,R3

8674 026024 013705 002102

MOV BANKINDEX,R5

8675 026030 010004

MOV R0,R4

8676 026032 022737 000001 003752

CMP #1,PROTYP

;IS THIS AN 11/44?

8677 026040 001006

BNE 1\$

;BRANCH IF NOT

8678 026042

BMOV MTP034

8679 026050 012737 177646 002260

MOV #UIPAR3,SUPDOADD

8680 026056

1\$:

IF #BIT13 SET IN CONFIG+2(R5)

;BACKGROUND PATTERN IS VALID

8681

8682 026066 022737 000001 003752

CMP #1,PROTYP

8683 026074 001403

BEQ 2\$

8684 026076 012737 036346 002260

MOV #MTP034+6,SUPDOADD

8685 026104 004737 027206

2\$:

CALL SUPD03

;READ IT

8686 026110

ELSE

;BACKGROUND PATTERN HAS BEEN INVALIDATED

8687

8688 026112 022737 000001 003752

CMP #1,PROTYP

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 246-1  
 MT0034 SOFT ERROR - BACKGROUND PATTERN TEST

```

8689 026120 001406          BEQ 3$
8690 026122 012737 036340 002260      MOV #MTP034,SUPDOADD
8691 026130 004737 027206          CALL SUPD03
8692 026134 000402          BR 4$
8693 026136 004737 027030 3$:      CALL SUPD01      ;WRITE IT
8694 026142 052765 020000 002652 4$:      BIS #BIT13,CONFIG+2(R5)      ;VALIDATE IT
8695 026150          END :OF IF #BIT13
8696 026150 000207          RETURN
8697
8698 026152

```

MT0035: SUBTST <<MT0035 SETUP WORST CASE NOISE PARITY TEST>>

\*\*\*\*\*  
 ;\*SUBTEST MT0035 SETUP WORST CASE NOISE PARITY TEST  
 ;\*\*\*\*\*

```

8699 026152 012737 000035 002274      MOV #35,REALPAT      ;SET UP TEST NUMBER FOR DISPLAY
8700 026160 013703 002102      MOV BANKINDEX,R3
8701 026164 016301 002650      MOV CONFIG(R3),R1
8702 026170 000301          SWAB R1
8703 026172 042701 177760      BIC #^C17,R1
8704 026176 006301          ASL R1
8705 026200 010137 002150      MOV R1,CSRNO
8706 026204 023737 002150 002526      CMP CSRNO,PGMCSR
8707 026212 001001          BNE 1$
8708 026214 000207          RETURN
8709 026216 012702 052524 1$:      MOV #52524,R2
8710 026222 004737 041224          CALL BACKGND      ;WRITE BACKGROUND OF ALMOST ALT. 1'S AND 0'S
8711 026226 012737 036364 002260      MOV #MTP035,SUPDOADD
8712 026234 004737 027206          CALL SUPD03
8713 026240          IF QVFLAG IS TRUE THEN $RETURN
8714 026250 005102          COM R2
8715 026252 004737 041224          CALL BACKGND      ;WRITE COMPLEMENT PATTERN INTO MUT
8716 026256 004737 027222          CALL SUPD04
8717 026262 000207          RETURN

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 247  
 MT0035 SETUP WORST CASE NOISE PARITY TEST

8719 026264

MT0036: SUBTST <<MT0036 SETUP CORRECTION CODE TEST>>

\*\*\*\*\*  
 :SUBTEST MT0036 SETUP CORRECTION CODE TEST  
 :\*\*\*\*\*

8720 026264				IF PMEMFLG IS FALSE THEN \$RETURN ; IF NOT MS11-P THEN EXIT
8721 026274	012737	000036	002274	MOV #36,REALPAT ; SET UP TEST NUMBER FOR TYPEOUT AND DISPLAY
8722 026302	004737	041324		CALL GETCSR ; GET CSR INFO FROM CONFIG TABLE
8723 026306	005037	002262		CLR PASFLG ; CLEAR LOOP COUNTER
8724 026312	005000			CLR R0 ; GET TEST DATA
8725 026314	012701	100000		MOV #100000,R1 ; GET FIRST ADDRESS IN BANK
8726 026320	004737	046706		CALL MAPKERNAL ; MAP KIPARS AND 6 TO BANK
8727 026324	004737	036526		CALL MTP036 ; EXECUTE TEST
8728 026330	004737	046774		CALL UNMAP ; REMAP KERNAL SPACE
8729 026334	000207			RETURN

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 248  
 MT0036 SETUP CORRECTION CODE TEST

8731 026336

MT0037: SUBTST <<MT0037 SETUP ECC DISABLE TEST>>

\*\*\*\*\*  
 :\*SUBTEST MT0037 SETUP ECC DISABLE TEST  
 :\*\*\*\*\*

8732 026336  
 8733 026346 012737 000037 002274  
 8734 026354 012701 100000  
 8735 026360 005000  
 8736 026362 004737 046706  
 8737 026366 004737 041324  
 8738 026372 004737 036752  
 8739 026376 004737 046774  
 8740 026402 000207

IF PMEMFLG IS FALSE THEN \$RETURN ;RETURN IF NOT A MS11-P  
 MOV #37,REALPAT ;SETUP PATTERN AND NUMBER FOR TYPEOUT AND DISPLAY  
 MOV #100000,R1 ;SET UP TEST ADDRESS  
 CLR R0 ;CLEAR DATA TO BE WRITTEN  
 CALL MAPKERNAL ;MAP THIS TEST TO KERNEL SPACE  
 CALL GETCSR ;GET CSRINFO FROM CONFIG TABLE  
 CALL MTP037 ;CHECK ECC DISABLE  
 CALL UNMAP ;REMAP KERNEL SPACE  
 RETURN

8741  
 8742  
 8743 026404

MT0040: SUBTST <<MT0040 >>

\*\*\*\*\*  
 :\*SUBTEST MT0040  
 :\*\*\*\*\*

8744 026404 000207

RETURN ;

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 249  
MT0040

8746 026406

MT0041: SUBTST <<MT0041 SETUP ADDRESS TO CSR ON DOUBLE BIT ERROR TEST>>

\*\*\*\*\*  
:SUBTEST MT0041 SETUP ADDRESS TO CSR ON DOUBLE BIT ERROR TEST  
\*\*\*\*\*

8747 026406  
8748 026416 012737 000041 002274  
8749 026424 004737 041324  
8750 026430  
8751 026436  
8752 026442 004737 027206  
8753 026446 000207

IF PMEMFLG IS FALSE THEN \$RETURN ;EXIT IF NOT MS11-P  
MOV #41,REALPAT ;SETUP PATTERN AND NUMBER FOR TYPEOUT AND DISPLAY  
CALL GETCSR ;GET CSR NUMBER AND ADDRESS FROM CONFIGURATION TABLE  
LET SUPDOADD := #MTP041 ;SET UP TEST ADDRESS  
LET R1 := #FIRST ;SET UP FIRST ADDRESS  
CALL SUPDO3 ;EXECUTE ADDRESS TO CSR TEST IN SUPVISIOR MODE  
RETURN ;

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 250  
 MT0041 SETUP ADDRESS TO CSR ON DOUBLE BIT ERROR TEST

8755 026450

MT0042: SUBTST <<MT0042 SETUP EXTENDED UNIBUS ADDRESS TO CSR TEST>>

\*\*\*\*\*  
 :\*SUBTEST MT0042 SETUP EXTENDED UNIBUS ADDRESS TO CSR TEST  
 :\*\*\*\*\*

8756 026450  
 8757 026460 012737 000042 002274  
 8758 026466 012701 100000  
 8759 026472 004737 046706  
 8760 026476 004737 041324  
 8761 026502 004737 037176  
 8762 026506 004737 046774  
 8763 026512 000207

IF PMEMFLG IS FALSE THEN \$RETURN ;EXIT IF NOT MS11-P  
 MOV #42,REALPAT ;SETUP PATTERN AND NUMBER FOR TYPEOUT AND DISPLAY  
 MOV #100000,R1 ;SET UP TEST ADDRESS  
 CALL MAPKERNAL ;MAP TO KERNEL SPACE  
 CALL GETCSR ;SET UP CSRINFO FROM CONFIGURATION TABLE  
 CALL MTP042 ;CHECK EXTENDED UNIBUS ADDRESS TO CSR  
 CALL UNMAP ;REMAP KERNEL SPACE  
 RETURN

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 251  
 MT0042 SETUP EXTENDED UNIBUS ADDRESS TO CSR TEST

8765 026514

MT0043: SUBTST <<MT0043 SETUP WRITE BYTE CLEARS SBE TEST>>

\*\*\*\*\*  
 :SUBTEST MT0043 SETUP WRITE BYTE CLEARS SBE TEST  
 \*\*\*\*\*

8766 026514  
 8767 026524 012737 000043 002274  
 8768 026532 004737 046706  
 8769 026536  
 8770 026542 004737 037432  
 8771 026546 004737 046774  
 8772 026552 000207  
 8773 026554

IF PMEMFLG IS FALSE THEN \$RETURN ;EXIT IF NOT MS11-P  
 MOV #43,REALPAT ;SET UP TEST NUMBER FOR TYPEOUT AND DISPLAY  
 CALL MAPKERNAL ;MAP TO KERNEL SPACE  
 LET R1 := #100000 ;SET UP TEST ADDRESS  
 CALL MTP043 ;PERFORM WRITE BYTE TEST  
 CALL UNMAP ;REMAP KERNEL SPACE  
 RETURN

MT0044: SUBTST <<MT0044 SETUP SHIFTING 1/0'S THROUGH THE CHECK BITS TEST>>

\*\*\*\*\*  
 :SUBTEST MT0044 SETUP SHIFTING 1/0'S THROUGH THE CHECK BITS TEST  
 \*\*\*\*\*

8774 026554  
 8775 026564 012737 000044 002274  
 8776 026572 004737 046706  
 8777 026576  
 8778 026602 004737 037626  
 8779 026606 004737 046774  
 8780 026612 000207  
 8781 026614

IF PMEMFLG IS FALSE THEN \$RETURN ;EXIT IF NOT MS11-P  
 MOV #44,REALPAT ;SET UP TEST NUMBER FOR TYPEOUT AND DISPLAY  
 CALL MAPKERNAL ;MAP TO KERNEL SPACE  
 LET R1 := #100000 ;SET UP TEST ADDRESS  
 CALL MTP044 ;PERFORM SHIFTING 1/0'S THROUGH THE CHECK BITS  
 CALL UNMAP ;REMAP KERNEL SPACE  
 RETURN

MT0045: SUBTST <<MT0045 SETUP SYNDROMES TO CSR ON DOUBLE BIT ERROR>>

\*\*\*\*\*  
 :SUBTEST MT0045 SETUP SYNDROMES TO CSR ON DOUBLE BIT ERROR  
 \*\*\*\*\*

8782 026614  
 8783 026624 012737 000045 002274  
 8784 026632 004737 046706  
 8785 026636  
 8786 026642 004737 040142  
 8787 026646 004737 046774  
 8788 026652 000207  
 8789 026654

IF PMEMFLG IS FALSE THEN \$RETURN ;EXIT IF NOT MS11-P  
 MOV #45,REALPAT ;SET UP TEST NUMBER FOR TYPEOUT AND DISPLAY  
 CALL MAPKERNAL ;MAP TO KERNEL SPACE  
 LET R1 := #100000 ;SET UP TEST ADDRESS  
 CALL MTP045 ;PERFORM SYNDROMES TO CSR ON DOUBLE BIT ERROR  
 CALL UNMAP ;REMAP KERNEL SPACE  
 RETURN

MT0046: SUBTST <<MT0046 SETUP CHECK SINGLE BIT ERRORS WITH ECC DISABLED TEST>>

\*\*\*\*\*  
 :SUBTEST MT0046 SETUP CHECK SINGLE BIT ERRORS WITH ECC DISABLED TEST  
 \*\*\*\*\*

8790 026654  
 8791 026664 012737 000046 002274  
 8792 026672 004737 046706  
 8793 026676  
 8794 026702 004737 040330  
 8795 026706 004737 046774  
 8796 026712 000207  
 8797 026714

IF PMEMFLG IS FALSE THEN \$RETURN ;EXIT IF NOT MS11-P  
 MOV #46,REALPAT ;SET UP TEST NUMBER FOR TYPEOUT AND DISPLAY  
 CALL MAPKERNAL ;MAP TO KERNEL SPACE  
 LET R1 := #100000 ;SET UP TEST ADDRESS  
 CALL MTP046 ;PERFORM TRAPS DETECTED ON SBE WITH ECC DISABLED TEST  
 CALL UNMAP ;REMAP KERNEL SPACE  
 RETURN

MT0047: SUBTST <<MT0047 SETUP NO CSR UPDATE ON SBE WITH EXSISTING DBE TEST>>

\*\*\*\*\*  
 :SUBTEST MT0047 SETUP NO CSR UPDATE ON SBE WITH EXSISTING DBE TEST  
 \*\*\*\*\*

8798 026714  
 8799 026724 012737 000047 002274  
 8800 026732 004737 046706  
 8801 026736  
 8802 026742  
 8803 026746 004737 040670  
 8804 026752 004737 046774  
 8805 026756 000207

IF PMEMFLG IS FALSE THEN \$RETURN ;EXIT IF NOT MS11-P  
 MOV #47,REALPAT ;SET UP TEST NUMBER FOR TYPEOUT AND DISPLAY  
 CALL MAPKERNAL ;MAP TO KERNEL SPACE  
 LET R1 := #100000 ;SET UP TEST ADDRESS  
 LET R2 := #120000 ;SECOND TEST ADDRESS  
 CALL MTP047 ;PERFORM NO UPDATE TO CSR ON SBE WITH DBE  
 CALL UNMAP ;REMAP KERNEL SPACE  
 RETURN

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 253  
 MT0047 SETUP NO CSR UPDATE ON SBE WITH EXSISTING DBE TEST

8808 026760

MT0999: SUBTST <<MT0999 SETUP NULL TEST>>

\*\*\*\*\*  
 :\*SUBTEST MT0999 SETUP NULL TEST  
 :\*\*\*\*\*

8809 026760 005037 002274

8810 026764

8811 026772 000207

8812

8813 026774

CLR REALPAT  
 SET NULLFLAG  
 RETURN

KAMITEST:SUBTST <<CHECK FOR KAMIKAZE MODE>>

\*\*\*\*\*  
 :\*SUBTEST CHECK FOR KAMIKAZE MODE  
 :\*\*\*\*\*

8814 026774

8815 027016

8816 027022

8817 027024

8818 027030

IF KAMIKAZE IS TRUE OR ACTFLAG IS TRUE OR APTFLAG IS TRUE  
 \$RETURN NOERROR ;RUN THE TEST  
 ELSE  
 \$RETURN ERROR ;DON'T RUN THE TEST  
 END ;OF IF KAMIKAZE



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 255  
CHECK FOR KAMIKAZE MODE

8821 027030

SUPD01: SUBTST <<SUBR EXECUTE PATTERN IN SUPERVISOR>>

\*\*\*\*\*

\*SUBTEST SUBR EXECUTE PATTERN IN SUPERVISOR

\*\*\*\*\*

8822 027030  
8823 027044 004737 060450  
8824 027050  
8825 027060 010037 002156  
8826 027064 012700 002160  
8827 027070 010120  
8828 027072 010220  
8829 027074 010320  
8830 027076 010420  
8831 027100 010520  
8832 027102 010620  
8833 027104 013700 002156  
8834 027110 012737 027124 002606  
8835 027116 013737 002606 002610  
8836 027124 012700 002174  
8837 027130 014006  
8838 027132 014005  
8839 027134 014004  
8840 027136 014003  
8841 027140 014002  
8842 027142 014001  
8843 027144 014000  
8844 027146  
8845 027154 012706 000740  
8846 027160 104424  
8847 027162 004737 177640  
8848 027166 104423  
8849 027170 104417  
8850 027172 000004  
8851 027174  
8852 027204 000207

MAP BANK ;MAP SUPERVISOR SPACE (TEST AREA) TO BANK  
SUPD02: CALL GETDIS  
PUSH \$LPERR,\$LPADR  
MOV R0,SUPDR0  
MOV #SUPDR1,R0  
MOV R1,(R0)+  
MOV R2,(R0)+  
MOV R3,(R0)+  
MOV R4,(R0)+  
MOV R5,(R0)+  
MOV SP,(R0)+  
MOV SUPDR0,R0  
MOV #TAG4\$,\$LPADR  
MOV \$LPADR,\$LPERR  
TAG4\$: MOV #SUPDR6+2,R0  
MOV -(R0),SP  
MOV -(R0),R5  
MOV -(R0),R4  
MOV -(R0),R3  
MOV -(R0),R2  
MOV -(R0),R1  
MOV -(R0),R0  
SUPERVISOR ;ENTER SUPERVISOR MODE  
MOV #SUPSTK,SSP  
CACHOFF ;TURN CACHE OFF  
CALL FASTCITY ;CALL TO THE USER INSTRUCTION PAR'S  
CACHON ;TURN CACHE ON  
KERNEL ;ENTER KERNEL MODE  
SCOPE  
POP \$LPADR,\$LPERR  
RETURN

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 257  
 SUBR EXECUTE PATTERN IN SUPERVISOR

8855	027206				SUPD03: MAP	BANK	
8856	027222	004737	060450		SUPD04: CALL	GETDIS	
8857	027226				PUSH	\$LPERR,\$LPADR	
8858	027236	010037	002156		MOV	R0,SUPDR0	
8859	027242	012700	002160		MOV	#SUPDR1,R0	
8860	027246	010120			MOV	R1,(R0)+	
8861	027250	010220			MOV	R2,(R0)+	
8862	027252	010320			MOV	R3,(R0)+	
8863	027254	010420			MOV	R4,(R0)+	
8864	027256	010520			MOV	R5,(R0)+	
8865	027260	010620			MOV	SP,(R0)+	
8866	027262	013700	002156		MOV	SUPDR0,R0	
8867	027266	012737	027302	002606	MOV	#TBG4\$,\$LPADR	
8868	027274	013737	002606	002610	MOV	\$LPADR,\$LPERR	
8869	027302	012700	002174		TBG4\$: MOV	#SUPDR6+2,R0	
8870	027306	014006			MOV	-(R0),SP	
8871	027310	014005			MOV	-(R0),R5	
8872	027312	014004			MOV	-(R0),R4	
8873	027314	014003			MOV	-(R0),R3	
8874	027316	014002			MOV	-(R0),R2	
8875	027320	014001			MOV	-(R0),R1	
8876	027322	014000			MOV	-(R0),R0	
8877	027324				TESTAREA		
8878	027332	005737	002452		TST	NOSUPER	
8879	027336	001403			BEQ	1\$	
8880	027340	012706	000700		MOV	#USESTK,USP	
8881	027344	000402			BR	2\$	
8882	027346	012706	000740		1\$: MOV	#SUPSTK,SSP	
8883	027352	104424			2\$: CACHOFF		
8884	027354	004777	152700		CALL	@SUPDOADD	
8885	027360	104423			CACHON		
8886	027362	104417			KERNEL		
8887	027364	000004			SCOPE		
8888	027366				POP	\$LPADR,\$LPERR	
8889	027376	000207			RETURN		

;MAP SUPERVISOR SPACE (TEST AREA) TO BANK

;ENTER SUPERVISOR MODE

;TURN CACHE OFF

;TURN CACHE ON

;ENTER KERNEL MODE

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 259  
MEMORY TEST PATTERN ROUTINES

```

8892      .SBTTL  MEMORY TEST PATTERN ROUTINES
8893      *****
8894      : PATTERN REGISTER CONVENTIONS
8895      : R0      FIRST ADDRESS OF PATTERN (FIRST, LAST+2, ETC)
8896      : R1      NUMBER OF ADDRESSES IN PATTERN (SIZE)
8897      : R2      DATA FOR PATTERN (ONES, 52525, ETC)
8898      : R3      COPY OF R1 (IF NECESSARY)
8899      : R4      COPY OF R0 (IF NECESSARY)
8900      : R5      COPY OF R2 (IF NECESSARY)
8901      *****
8902 027400 MTP000: SUBTST <<MTP000      BASIC DATA TEST>>
      *****
      : *SUBTEST      MTP000  BASIC DATA TEST
      *****
8903 027400 010220 1$:  MOV      R2,(R0)+      ;V177640
8904 027402 077102      SOB      R1,MTP000      ;V177642
8905 027404 000240      NOP              ;V177644
8906 027406 012401 2$:  MOV      (R4)+,R1      ;V177646
8907 027410 020102      CMP      R1,R2      ;V177650
8908 027412 001402      BEQ      3$          ;V177652
8909 027414 104430      PERR02      ;V177654
8910 027416 000240      NOP              ;V177656
8911 027420 077306 3$:  SOB      R3,2$      ;V177660
8912 027422 000207      RETURN      ;V177662
8913 027424 MTP001: SUBTST <<MTP001      ADDRESS TEST>>
      *****
      : *SUBTEST      MTP001  ADDRESS TEST
      *****
8914 027424 010220 3$:  MOV      R2,(R0)+      ;V177640
8915 027426 062702 000002 ADD      #2,R2      ;V177642
8916 027432 077104      SOB      R1,3$      ;V177646
8917 027434 000240      NOP              ;V177650
8918 027436 012400 1$:  MOV      (R4)+,R0      ;V177652
8919 027440 020005      CMP      R0,R5      ;V177654
8920 027442 001401      BEQ      2$          ;V177656
8921 027444 104427      PERR01      ;V177660
8922 027446 062705 000002 ADD      #2,R5      ;V177662
8923 027452 077307      SOB      R3,1$      ;V177666
8924 027454 000207      RETURN      ;V177672
8925 027456 MTP002: SUBTST <<MTP002      COMPLEMENT ADDRESS TEST (WRITE DOWN, READ UP)>>
      *****
      : *SUBTEST      MTP002  COMPLEMENT ADDRESS TEST (WRITE DOWN, READ UP)
      *****
8926 027456 010540 3$:  MOV      R5,-(R0)      ;V177640
8927 027460 062705 000002 ADD      #2,R5      ;V177642
8928 027464 077104      SOB      R1,3$      ;V177646
8929 027466 000240      NOP              ;V177650
8930 027470 162702 000002 1$:  SUB      #2,R2      ;V177652
8931 027474 012401      MOV      (R4)+,R1      ;V177656
8932 027476 020102      CMP      R1,R2      ;V177660
8933 027500 001401      BEQ      2$          ;V177662
8934 027502 104430      PERR02      ;V177664
8935 027504 077307 2$:  SOB      R3,1$      ;V177666
8936 027506 000207      RETURN      ;V177670

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 261  
MTP002 COMPLEMENT ADDRESS TEST (WRITE DOWN, READ UP)

8939 027510

MTPA03: SUBTST &lt;&lt;MTPA03 3 XOR 9 WORST CASE NOISE TEST (WRITE)&gt;&gt;

\*\*\*\*\*  
:SUBTEST MTPA03 3 XOR 9 WORST CASE NOISE TEST (WRITE)  
\*\*\*\*\*

8940  
8941  
8942  
8943  
8944  
8945  
8946 027510 010421  
8947 027512 010421  
8948 027514 077203  
8949 027516 005104  
8950 027520 052704  
8951 027522 000401  
8952 027524 012702 000004  
8953 027530 077511  
8954 027532 005104  
8955 027534 052704  
8956 027536 000401  
8957 027540 012705 000100  
8958 027544 077317  
8959 027546 000207  
8960  
8961  
8962 027550

:R1 = ADDRESS  
:R2 = SMALL LOOP CONSTANT  
:R3 = NUM OF ADD TO TEST (LARGE LOOP)  
:R4 = GOOD DATA  
:R5 = MEDIUM LOOP CONSTANT  
.ENABL LSB  
1\$: MOV R4,(R1)+ :V177640  
MOV R4,(R1)+ :V177642  
SOB R2,1\$ :V177644  
COM R4 :V177646  
BIS (PC)+,R4 :V177650  
WARN2: 401 :V177652 WARNING LOCATION IS MODIFIED BEFORE LOADING  
MOV #4,R2 :V177654  
SOB R5,1\$ :V177660  
COM R4 :V177662  
BIS (PC)+,R4 :V177664  
WARN3: 401 :V177666 WARNING LOCATION IS MODIFIED BEFORE LOADING  
MOV #64,R5 :V177670  
SOB R3,1\$ :V177674  
RETURN :V177676  
.DSABL LSB

MTPB03: SUBTST &lt;&lt;MTPB03 3 XOR 9 WORST CASE NOISE TEST (READ)&gt;&gt;

\*\*\*\*\*  
:SUBTEST MTPB03 3 XOR 9 WORST CASE NOISE TEST (READ)  
\*\*\*\*\*

8963  
8964 027550 000137 027610  
8965 027554 077203  
8966 027556 005104  
8967 027560 052704  
8968 027562 000401  
8969 027564 012702 000004  
8970 027570 077511  
8971 027572 005104  
8972 027574 052704  
8973 027576 000401  
8974 027600 012705 000100  
8975 027604 077317  
8976 027606 000207  
8977

.ENABL LSB  
1\$: JMP @MTPC03 :V177640 GO TO V172360  
SOB R2,1\$ :V177644  
COM R4 :V177646  
BIS (PC)+,R4 :V177650  
WARN4: 401 :V177652 WARNING LOCATION IS MODIFIED BEFORE LOADING  
MOV #4,R2 :V177654  
SOB R5,1\$ :V177660  
COM R4 :V177662  
BIS (PC)+,R4 :V177664  
WARN5: 401 :V177666 WARNING LOCATION IS MODIFIED BEFORE LOADING  
MOV #64,R5 :V177670  
SOB R3,1\$ :V177674  
RETURN :V177676  
.DSABL LSB

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 263  
 MTPB03 3 XOR 9 WORST CASE NOISE TEST (READ)

8980 027610

```

MTPC03: SUBTST <<MTPC03      TEST DATA SUBPROGRAM>>
:*****
:*SUBTEST      MTPC03 TEST DATA SUBPROGRAM
:*****
      CMP      R4,(R1)+      :V172360
      BEQ      1$           :V172362
      PERR03           :V172364
1$:      CC#      -(R1)      :V172366
      COM      (R1)      :V172370
      JMP      @#MTPD03      :V172372      GO TO V172260
  
```

8981 027610 020421  
 8982 027612 001401  
 8983 027614 104431  
 8984 027616 005141  
 8985 027620 005111  
 8986 027622 000137 027626  
 8987  
 8988 027626

```

MTPD03: SUBTST <<MTPD03      TEST DATA SUBSUBPROGRAM>>
:*****
:*SUBTEST      MTPD03 TEST DATA SUBSUBPROGRAM
:*****
      CMP      R4,(R1)+      :V172260
      BEQ      1$           :V172262
      PERR03           :V172264
1$:      COM      (PC)+      :V172266
      O           :V172270
      BNE      MTPC03      :V172272      GO TO V172360
      JMP      @#MTPB03+4    :V172274      GO TO V177644
  
```

8989 027626 020421  
 8990 027630 001401  
 8991 027632 104431  
 8992 027634 005127  
 8993 027636 000000  
 8994 027640 001363  
 8995 027642 000137 027554

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 265  
MTPD03 TEST DATA SUBSUBPROGRAM

8998 027646

MTPA04: SUBTST &lt;&lt;MTPA04 ROTATING ZEROS TEST&gt;&gt;

\*\*\*\*\*  
: \*SUBTEST MTPA04 ROTATING ZEROS TEST  
\*\*\*\*\*

8999 027646 012705 000010  
9000 027652 010504  
9001 027654 000241  
9002 027656 000137 027702  
9003 027662 016004 177776  
9004 027666 103402  
9005 027670 020204  
9006 027672 001401  
9007 027674 104432  
9008 027676 077115  
9009 027700 000207  
9010  
9011 027702

1\$: MOV #8.,R5 :V177640  
MOV R5,R4 :V177644  
CLC :V177646  
JMP @MTPB04 :V177650  
MOV -2(R0),R4 :V177654  
BCS 2\$ :V177660  
CMP R2,R4 :V177662  
BEQ 3\$ :V177664  
2\$: PERR04 :V177666  
3\$: SOB R1,1\$ :V177670  
RETURN :V177672

MTPB04: SUBTST &lt;&lt;MTPB04 SUBR ROTATING BIT&gt;&gt;

\*\*\*\*\*  
: \*SUBTEST MTPB04 SUBR ROTATING BIT  
\*\*\*\*\*

9012 027702 106110  
9013 027704 077502  
9014 027706 106120  
9015 027710 106110  
9016 027712 077402  
9017 027714 106120  
9018 027716 000137 027662  
9019  
9020 027722

1\$: ROLB (R0) :V172360  
SOB R5,1\$ :V172362  
ROLB (R0)+ :V172364  
2\$: ROLB (R0) :V172366  
SOB R4,2\$ :V172370  
ROLB (R0)+ :V172372  
JMP @MTPA04+14 :V172374

MTP005: SUBTST &lt;&lt;MTP005 ROTATION ONES TEST&gt;&gt;

\*\*\*\*\*  
: \*SUBTEST MTP005 ROTATION ONES TEST  
\*\*\*\*\*

9021 027722 012705 000010  
9022 027726 010504  
9023 027730 000261  
9024 027732 000137 027702  
9025 027736 016004 177776  
9026 027742 103002  
9027 027744 020204  
9028 027746 001401  
9029 027750 104432  
9030 027752 077115  
9031 027754 000207

1\$: MOV #8.,R5 :V177640  
MOV R5,R4 :V177644  
SEC :V177646  
JMP @MTPB04 :V177650  
MOV -2(R0),R4 :V177654  
BCC 2\$ :V177660  
CMP R2,R4 :V177662  
BEQ 3\$ :V177664  
2\$: PERR04 :V177666  
3\$: SOB R1,1\$ :V177670  
RETURN :V177672

IF THIS HAPPENS THE GOOD &amp; BAD MATCH

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 267  
MTP005 ROTATION ONES TEST

9034 027756

MTP006: SUBTST &lt;&lt;MTP006 INITIAL DATA TEST&gt;&gt;

\*\*\*\*\*  
:SUBTEST MTP006 [INITIAL DATA TEST  
\*\*\*\*\*

```

9035
9036
9037 027756 012737 000001 002240      MOV    #1,DATBUF      ;SET THE FIRST TEST BIT
9038 027764 005037 002242      CLR    DATBUF+2      ;CLEAR 2ND WORD
9039 027770 013771 002240 000000 1$:  MOV    DATBUF,@(R1)    ;WRITE TEST WORD 1
9040 027776 013771 002242 000002      MOV    DATBUF+2,@2(R1) ;AND TEST WORD 2
9041 030004 017102 000000      MOV    @2(R1),R2
9042 030010 023702 002240      CMP    DATBUF,R2      ;NOW READ THEM
9043 030014 001401      BEQ    2$      ;BR IF FIRST 16 OK
9044 030016 104433      PERR07      ;ERROR TRAP
9045
9046 030020 017102 000002      2$:  MOV    @2(R1),R2
9047 030024 023702 002242      CMP    DATBUF+2,R2    ;NOW READ SECOND WORD
9048 030030 001401      BEQ    3$      ;BR IF OK
9049 030032 104434      PERR10      ;ERROR TRAP
9050
9051 030034 005737 002242      3$:  TST    DATBUF+2      ;HAS LAST BIT BEEN TESTED ?
9052 030040 100405      BMI    4$      ;MINUS MEANS BIT 31
9053 030042      DLEFT  DATBUF      ;NO, SHIFT TEST BIT LEFT
9054 030052 000746      BR     1$      ;GO WRITE NEW TEST DATA
9055
9056 030054 012737 177776 002240 4$:  MOV    #177776,DATBUF ;PUT A 0 IN BIT 0
9057 030062 012737 177777 002242      MOV    #-1,DATBUF+2 ;AND 1'S IN ALL OTHERS
9058 030070 013771 002240 000000 5$:  MOV    DATBUF,@(R1)    ;WRITE THE DATA
9059 030076 013771 002242 000002      MOV    DATBUF+2,@2(R1) ;2 WORDS WORTH
9060 030104 017102 000000      MOV    @2(R1),R2
9061 030110 023702 002240      CMP    DATBUF,R2      ;NOW READ FIRST WORD
9062 030114 001401      BEQ    6$      ;BR IF OK
9063 030116 104433      PERR07
9064
9065 030120 017102 000002      6$:  MOV    @2(R1),R2
9066 030124 023702 002242      CMP    DATBUF+2,R2    ;NOW, READ SECOND WORD
9067 030130 001401      BEQ    7$      ;BR IF OK
9068 030132 104434      PERR10
9069
9070 030134 005737 002242      7$:  TST    DATBUF+2      ;TESTED BIT 31 YET?
9071 030140 100005      BPL    8$      ;BR IF YES, WE'RE DONE
9072 030142      DLEFT  DATBUF
9073 030152 000746      BR     5$      ;KEEP GOING
9074 030154 000207      8$:  RETURN

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 269  
MTP006 INITIAL DATA TEST

9077 030156

MTP007: SUBTST <<MTP007 ADDRESS BIT TEST>>

\*\*\*\*\*  
:SUBTEST MTP007 ADDRESS BIT TEST  
\*\*\*\*\*

THIS TEST CHECKS TO SEE THAT EACH ADDRESS  
BIT IN EACH 16K BANK CAN BE ASSERTED UNIQUELY.  
IT CHECKS FOR ADDRESS BITS THAT MAY BE STUCK  
HIGH, STUCK LOW OR STUCK TOGETHER.

9078					
9079					
9080					
9081					
9082	030156	111100		MOVB	(R1),R0
9083	030160	105700		TSTB	R0
9084	030162	001401		BEQ	1\$
9085	030164	104435		PERR11	
9086					
9087	030166	105111	1\$:	COMB	(R1)
9088	030170	111100		MOVB	(R1),R0
9089	030172	105700		TSTB	R0
9090	030174	001001		BNE	2\$
9091	030176	104436		PERR12	
9092					
9093	030200	040201	2\$:	BIC	R2,R1
9094	030202	006302		ASL	R2
9095	030204	050201		BIS	R2,R1
9096	030206	011100		MOV	(R1),R0
9097	030210	005700		TST	R0
9098	030212	001401		BEQ	3\$
9099	030214	104437		PERR13	
9100					
9101	030216	005111	3\$:	COM	(R1)
9102	030220	011100		MOV	(R1),R0
9103	030222	005700		TST	R0
9104	030224	001001		BNE	4\$
9105	030226	104400		PERR14	
9106					
9107	030230	022700	4\$:	CMP	#100000,R2
9108	030234	001400		BEQ	5\$
9109	030236	022700		CMP	#10000,R2
9110	030242	001356		BNE	2\$
9111	030244	006302		ASL	R2
9112	030246	012701	160000	MOV	#160000,R1
9113	030252	000752		BR	2\$
9114	030254	000207	5\$:	RETURN	

;READ AND COMPARE FOR ZEROS  
;BR IF OK

;COMPLEMENT THE BYTE  
;READ FOR NON ZEROS  
;BR IF OK

;MASK OFF THE ASSERTED BIT  
;SHIFT R2 FOR NEXT BIT  
;SET THE NEW BIT INTO R1  
;READ THE NEW ADDRESS  
;READ FOR ZEROS

;COMPL THE WORD  
;READ IT AGAIN

;CHECK FOR MSB IN 4K BANK  
;NOT LAST BIT, BRANCH



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 271  
MTP007 ADDRESS BIT TEST

9117 030256

MTP010: SUBTST <<MTP010 BYTE ADDRESSING TEST>>

\*\*\*\*\*  
:SUBTEST MTP010 BYTE ADDRESSING TEST  
\*\*\*\*\*

```

9118
9119
9120 030256 010402
9121 030260 010403
9122 030262 062702 000004
9123 030266 012713 177777
9124 030272 012763 177777 000002
9125 030300 105013
9126 030302 010401
9127 030304 020201
9128 030306 001420
9129 030310 020301
9130 030312 001007
9131 030314 111100
9132
9133 030316 022700 000000
9134 030322 001401
9135 030324 104435
9136
9137 030326 005201
9138 030330 000765
9139 030332 111100
9140 030334 122700 177777
9141 030340 001401
9142 030342 104436
9143
9144 030344 005201
9145 030346 000756
9146 030350 112713 177777
9147 030354 005203
9148 030356 020302
9149 030360 001347
9150 030362 000207

```

```

MOV R4,R2
MOV R4,R3
ADD #4,R2
MOV #-1,(R3)
MOV #-1,2(R3)
1$: CLRB (R3)
MOV R4,R1
2$: CMP R2,R1
BEQ 6$
CMP R3,R1
BNE 4$
MOVB (R1),R0
:WARNING IF YOU OPTOMIZE CHANGE THE PCBUMP FOR THIS ERROR INCASE OF TRAPS
CMP #0,R0
BEQ 3$
PERR11
3$: INC R1
BR 2$
4$: MOVB (R1),R0
CMPB #-1,R0
BEQ 5$
PERR12
5$: INC R1
BR 2$
6$: MOVB #-1,(R3)
INC R3
CMP R3,R2
BNE 1$
RETURN

```

```

:R4 HAS LOWEST ADDRESS
:PUT IT IN R3 ALSO
:POINT R2 TO LAST BYTE +1
:WRITE ALL ONES IN
:THE 4 TEST BYTES
:CLEAR A BYTE
:INITIALIZE R1 FOR EACH PASS
:IF EQUAL, JUST READ LAST BYTE
:BR IF EQUAL
:IS THIS THE BYTE OF ZEROS
:BR IF NOT
:IT IS, COMPARE FOR ZEROS
:NEXT BYTE
:RETURN
:ITS NOT THE BYTE OF 0'S, READ 1'S
:MOVE TO NEXT BYTE
:RESTORE 1'S TO BYTE JUST TESTED
:INC TO NEXT BYTE
:WAS THAT JUST THE LAST ONE?
:BR IF NO

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 273  
MTP010 BYTE ADDRESSING TEST

9153 030364

```

MTP011: SUBTST <<MTP011      SINGLE BIT ERROR TEST>>
:*****
:*SUBTEST      MTP011  SINGLE BIT ERROR TEST
:*****
:  (1)  CREATE A SINGLE BIT ERROR
:  (2)  READ BACK SBE UNCORRECTED (WITH ECC DISABLE)
:  (3)  ENABLE ECC & READ CORRECTED DATA
:  (4)  CHECK THAT THE SBE FLAG WAS SET FROM THE LAST READ
:  (5)  DO (1-4) FOR DATA CONSISTING OF 1 BIT SET IN EACH OF 32
:        POSITIONS OF A DOUBLE WORD
:        THEN DO IT AGAIN FOR 1 BIT CLEARED IN EACH OF 32 POSITIONS OF
:        A DOUBLE WORD
:        IE (64 TIMES)
:  (6)  DO (1-5) FOR A SBE IN EACH OF 32 BIT POSITIONS
:        IE (RUN TEST 64 * 32 = 2048 TIMES)
:CLR1CSR      :CLEAR 1 SELECTED CSR
:TST          PHEBE      :TEST SPECIAL CASE INDICATOR
:BEQ          MTLA11     :BRANCH IF NOT SET
:MOV          SIPAR3,R2  :SAVE CONTENTS OF SIPAR #3
:MOV          SIPAR5,R2  :COPY CONTENTS OF #5 INTO #3
:MOV          R2,R5      :COPY CONTENTS OF #3 INTO #5
:BIG LOOP
:  MTLA11: MOV      #1,DATBUF      :INITIAL DATA
:          CLR      DATBUF+2      :32 BITS WORTH
:          :MEDIUM LOOP
:  MTLB11: MOV      #1,SBEMSK     :INITIAL ERROR MASK
:          CLR      SBEMSK+2      :32 BITS WORTH
:          :LITTLE LOOP
:  MTLA11: MOV      DATBUF,TSTDAT  :
:          MOV      DATBUF+2,TSTDAT+2 :TO SAVE ORIG DATA
:          TSTB     PASFLG      :COMP DATA ON SECOND PASS ONLY
:          BEQ      4$          :BR IF FIRST PASS
:          COM      TSTDAT      :SECOND PASS, COMP BOTH WORDS
:          COM      TSTDAT+2
:          MOV      TSTDAT,R2
:          MOV      TSTDAT+2,R3
:          MOV      #TSTDAT,SOURCE :SET UP ADDRESS FOR CHKGEN
:          CALL     CHKGEN      :GEN CHECKBITS ON TSTDAT
:          :*****
:          ** CREATE A SINGLE BIT ERROR **
:          :*****
:          MOV      SBEMSK,R1
:          XOR      R1,TSTDAT
:          MOV      SBEMSK+2,R1
:          XOR      R1,TSTDAT+2
:          :FIRST TEST ADDRESS
:          MOV      TESTADD,R1   :SECOND TEST ADDRESS
:          MOV      TESTADD+2,R5 :DISABLE ECC ON 1 SELECTED CSR
:          ECC1DIS
:          MOV      TSTDAT,(R1)  :WRITE FIRST 16 BITS
:          CB1CSR
:          MOV      TSTDAT+2,(R5):WRITE GENERATED CHECKBITS IN 1 SELECTED CSR
:          :WRITE SECOND 16 BITS AND
:          :CHECK BITS. WE NOW HAVE CHECKBITS

```

9154  
 9155  
 9156  
 9157  
 9158  
 9159  
 9160  
 9161  
 9162  
 9163  
 9164  
 9165  
 9166  
 9167  
 9168  
 9169  
 9170 030364 104503  
 9171 030366 005737 014004  
 9172 030372 001407  
 9173 030374 013702 172246  
 9174 030400 013737 172252 172246  
 9175 030406 010237 172252  
 9176  
 9177 030412 012737 000001 002240  
 9178 030420 005037 002242  
 9179  
 9180 030424 012737 000001 002250  
 9181 030432 005037 002252  
 9182  
 9183 030436 013737 002240 002244  
 9184 030444 013737 002242 002246  
 9185 030452 105737 002262  
 9186 030456 001404  
 9187 030460 005137 002244  
 9188 030464 005137 002246  
 9189 030470 013702 002244  
 9190 030474 013703 002246  
 9191 030500 012737 002244 002306  
 9192 030506 004737 044366  
 9193  
 9194  
 9195  
 9196 030512 013701 002250  
 9197 030516 074137 002244  
 9198 030522 013701 002252  
 9199 030526 074137 002246  
 9200 030532 013701 002406  
 9201 030536 013705 002410  
 9202 030542 104471  
 9203 030544 013711 002244  
 9204 030550 104475  
 9205 030552 013715 002246  
 9206

CZMSPA0 MS11-1/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 273-1  
MTP011 SINGLE BIT ERROR TEST

```

9207
9208
9209 030556 104471 ECC1DIS ;GENERATED ON DATBUF AND DATA WITH
9210 030560 011100 MOV (R1),R0 ;ONE BIT IN ERROR (AS PER SBEMSK).
9211 030562 020037 002244 CMP R0,TSTDAT ;DISABLE ECC ON 1 SELECTED CSR
9212 030566 001403 BEQ 6$ ;READ THE LOW WORD (UNCORRECTED)
9213 030570 010137 002032 MOV R1,ADDRESS ;BR IF OK
9214 030574 104455 PERR31
9215
9216 030576 011500 6$: MOV (R5),R0
9217 030600 020037 002246 CMP R0,TSTDAT+2 ;READ THE HIGH WORD (UNCORRECTED)
9218 030600 001403 BEQ 7$ ;BR IF OK
9219 030600 010537 002032 MOV R5,ADDRESS
9220 030612 104455 PERR31
9221
9222 030614 7$: IF KFLAG IS FALSE
9223 030622 104426 READCSR
9224 030624 IF #BIT4 OFF.IN CSR OR #BIT15 OFF.IN CSR
9225 030644 104045 ERROR +45
9226 030646 END; OF IF #BIT4
9227 030646 END; OF IF KFLAG
9228 030646 005737 014004 TST PHEBE
9229 030652 001001 BNE 17$
9230 030654 104512 ERGEN
9231 030656 104503 17$: CLR1CSR ;CLEAR 1 SELECTED CSR
9232 030660 011100 MOV (R1),R0
9233 030662 020002 CMP R0,R2 ;SEE IF ITS BEEN CORRECTED
9234 030664 001401 BEQ 8$ ;IT SHOULD HAVE BEEN
9235 030666 104456 PERR32
9236
9237 030670 104510 8$: TSTREAD ;TEST LOC (R1) & TST FOR SBE (WITHOUT FETCHES)
9238 030672 103411 BCS 9$ ;BR IF IT IS SET
9239 030674 SET HEADER ;ENABLE PRINTING OF ERROR HEADER INFO
9240 030702 010137 002032 MOV R1,ADDRESS
9241 030706 104460 PERR34
9242 030710 SET HEADER ;FNABLE PRINTING OF ERROR HEADER INFO
9243
9244 030716 104503 9$: CLR1CSR ;CLEAR 1 SELEC..D CSR
9245 030720 011500 MOV (R5),R0
9246 030722 020003 CMP R0,R5 ;SEE IF ITS BEEN CORRECTED
9247 030724 001401 BEQ 10$ ;BR IF OK
9248 030726 104456 PERR32
9249
9250 030730 104510 10$: TSTREAD ;TEST LOC (R1) & TST FOR SBE (WITHOUT FETCHES)
9251 030732 103411 BCS 11$ ;BR IF YES
9252 030734 SET HEADER ;ENABLE PRINTING OF ERROR HEADER INFO
9253 030742 010137 002032 MOV R1,ADDRESS
9254 030746 104460 PERR34
9255 030750 SET HEADER ;ENABLE PRINTING OF ERROR HEADER INFO
9256 030756 104512 11$: ERGEN ;TEST ERROR ADDRESS
9257 030760 105737 002262 TST PASFLG
9258 030764 100452 BMI 15$
9259 030766 005737 002252 TST SBEMSK+2 ;TEST FOR LAST MASK BIT
9260 030772 100405 BMI 12$ ;MINUS MEANS BIT 31
9261 030774 DLEFT SBEMSK
9262 031004 000614 BR MTL11
9263 031006 12$: IF #SW11 SET.IN @SWR THEN GOTO 13$

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 273-2  
 MTP011 SINGLE BIT ERROR TEST

9264	031016					IF QVFLAG IS TRUE THEN GOTO 13\$
9265	031024	005737	002242			TST DATBUF+2 ;LAST DATA BIT ?
9266	031030	100406				BMI 13\$ ;WHICH IS BIT 31
9267	031032					DLEFT DATBUF
9268	031042	000137	030424			JMP MTLB11
9269	031046	105737	002262	13\$:		TSTB PASFLG ;FIRST OR SECOND PASS ?
9270	031052	001004				BNE 14\$ ;NON ZERO MEANS WE'RE DONE
9271	031054	105237	002262			INCB PASFLG ;NOT DONE, GO DO SECOND PASS
9272	031060	000137	030412			JMP MTLA11
9273	031064	052737	000200	002262	14\$:	BIS #BIT7,PASFLG
9274	031072	005002				CLR R2
9275	031074	005003				CLR R3
9276	031076	005037	002244			CLR TSTDAT
9277	031102	005037	002246			CLR TSTDAT+2
9278	031106	012704	000040			MOV #40,R4
9279	031112	012737	003740	002310	15\$:	MOV #3740,CHECK
9280	031120	074437	002310			XOR R4,CHECK
9281	031124	006304				ASL R4
9282	031126	032704	020000			BIT #BIT13,R4
9283	031132	001002				BNE 16\$
9284	031134	000137	030532			JMP MTLD11
9285						;CLEAR OUT ANY DBE'S OR SBE'S
9286	031140	104471		16\$:		ECC1DIS ;DISABLE ECC ON 1 SELECTED CSR
9287	031142	013701	002406			MOV TESTADD,R1
9288	031146	013705	002410			MOV TESTADD+2,R5
9289	031152					CLEAR (R1),(R5)
9290	031156	104503				CLR1CSR ;CLEAR 1 SELECTED CSR
9291	031160	000207				RETURN

CZMSPA0 MS11-L/P/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 275  
MTP011 SINGLE BIT ERROR TEST

```

MTP012: SUBTST <<MTP012 WRITE BYTE CLEARS SBE TEST>>
:*****
:SUBTEST MTP012 WRITE BYTE CLEARS SBE TEST
:*****
: SINGLE BIT ERROR TEST TO INSURE THAT A WRITE
: BYTE CLEARS SINGLE BIT ERRORS.
: CLEAR 1 SELECTED CSR
CLR1CSR
MOV #1,DATBUF : INITIAL DATA
CLR DATBUF+2 : 32 BITS WORTH
1$: MOV #1,SBEMSK : INITIAL ERROR MASK
CLR SBEMSK+2 : 32 BITS WORTH
2$: MOV DATBUF,TSTDAT : SAVE ORIGINAL DATA
MOV DATBUF+2,TSTDAT+2 : BOTH WORDS
MOV #TSTDAT,SOURCE : NEED ADDRESS FOR CHKGEN
CALL CHKGEN : GENERATE CHECK BITS
MOV SBEMSK,R1
XOR R1,TSTDAT
MOV SBEMSK+2,R1
XOR R1,TSTDAT+2
MOV TESTADD,R4 : FIRST TEST ADDRESS
MOV R4,R1 : PUT IT IN R1 ALSO
ECC1DIS : DISABLE ECC ON 1 SELECTED CSR
MOV TSTDAT,(R1) : WRITE 16 BITS
CB1CSR : WRITE GENERATED CHECKBITS IN 1 SELECTED CSR
ADD R5,R1 : INDEX UP TO SECOND WORD
MOV TSTDAT+2,(R1) : WRITE HIGH WORD+CHECKBITS
CLR1CSR : CLEAR 1 SELECTED CSR
: IT'S DANGEROUS IF WE DON'T
MOV #SBEMSK,R2 : ADDRESS OF ERROR MASK
SUB R5,R1 : RETURN TO FIRST WORD
3$: MOVB #-1,(R1) : WRITE A BYTE OF 1'S
TST KFLAG : IS THIS MF11S-K
BEQ 4$ : BRANCH IF NOT - IT'S MS11-M
BITB #-1,(R2) : DID THIS BYTE HAVE THE BAD BIT IN IT?
BEQ 6$ : NO - BRANCH
4$: TSTREAD : TEST LOC (R1) & TST FOR SBE (WITHOUT FETCHES)
RCC 5$ : NO - SKIP
SET HEADER : ENABLE PRINTING OF ERROR HEADER INFO
MOV R1,ADDRESS
ERROR +17
SET HEADER : ENABLE PRINTING OF ERROR HEADER INFO
5$: MOVB (R1),R0
CMPB #-1,R0 : CHECK DATA
BEQ 7$ : BR IF OK
PERR33
6$: TSTREAD : TEST LOC (R1) & TST FOR SBE (WITHOUT FETCHES)
: READ THE BYTE
: SBE ERROR BIT ONLY SET ?
BCS 5$ : SHOULD BE SET, BR IF OK
SET HEADER : ENABLE PRINTING OF ERROR HEADER INFO
MOV R1,ADDRESS
PERR34
SET HEADER : ENABLE PRINTING OF ERROR HEADER INFO
7$: BITB #-1,(R2) : CHECK FOR LAST BYTE

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 275-1  
MTP012 WRITE BYTE CLEARS SBE TEST

9348	031424	001012		BNE	8\$	:
9349	031426	005202		INC	R2	
9350	031430	005201		INC	R1	:MOVE TO NEXT BYTE
9351	031432	013704	002406	MOV	TESTADD,R4	:FIRST TEST ADDRESS
9352	031436	032701	000002	BIT	#2,R1	:TEST FOR LOWER WORD
9353	031442	001723		BEQ	3\$	:BR IF IT'S LOW 16 BITS
9354	031444	062704	000002	ADD	#2,R4	:ADJUST POINTER FOR ERROR REPT.
9355	031450	000720		BR	3\$	
9356	031452	005737	002252	8\$: TST	SBEMSK+2	:LAST ERROR BIT ?
9357	031456	100405		BMI	9\$	:MINUS MEANS BIT 31
9358	031460			DLEFT	SBEMSK	
9359	031470	000647		BR	2\$	
9360	031472			9\$: IF #SW11 SET.IN @SWR THEN GOTO 10\$		
9361	031502			IF QVFLAG IS TRUE THEN GOTO 10\$		
9362	031510	005737	002242	TST	DATBUF+2	:LAST DATA BIT?
9363	031514	100405		BMI	10\$	:MINUS = BIT 31
9364	031516			DLEFT	DATBUF	
9365	031526	000623		BR	1\$	
9366				:CLEAR OUT ANY DBE'S OR SBE'S		
9367	031530	104471		10\$: ECC1DIS		:DISABLE ECC ON 1 SELECTED CSR
9368	031532	013701	002406	MOV	TESTADD,R1	
9369	031536	005011		CLR	(R1)	
9370	031540	060501		ADD	R5,R1	
9371	031542	005011		CLR	(R1)	
9372	031544	104503		CLR1CSR		:CLEAR 1 SELECTED CSR
9373	031546	000207		RETURN		

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 277  
MTP012 WRITE BYTE CLEARS SBE TEST

9376 031550

MTP013: SUBTST &lt;&lt;MTP013 CREATE DOUBLE BIT ERROR TEST&gt;&gt;

\*\*\*\*\*  
:SUBTEST MTP013 CREATE DOUBLE BIT ERROR TEST  
\*\*\*\*\*

```

9377
9378 031550 104503
9379 031552 012701 002406
9380 031556 005037 002240
9381 031562 005037 002242
9382 031566 012737 000001 002250
9383 031574 005037 002252
9384 031600 012737 000001 002254
9385 031606 005037 002256
9386 031612 013737 002240 002244
9387 031620 013737 002242 002246
9388 031626 105737 002262
9389 031632 001404
9390 031634 005137 002244
9391 031640 005137 002246
9392 031644 104503
9393 031646 023737 002250 002254
9394 031654 001004
9395 031656 023737 002252 002256
9396 031664 001460
9397 031666 012737 002244 002306
9398 031674 004737 044366
9399 031700 013702 002250
9400 031704 074237 002244
9401 031710 013702 002252
9402 031714 074237 002246
9403 031720 013702 002254
9404 031724 074237 002244
9405 031730 013702 002256
9406 031734 074237 002246
9407 031740 104471
9408 031742 013731 002244
9409 031746 104475
9410 031750 013771 002246 000000
9411 031756 104503
9412 031760 162701 000002
9413 031764 005771 000000
9414 031770 104501
9415 031772 103411
9416 031774
9417 032002 011137 002032
9418 032006 104030
9419 032010

          CLR1CSR
          MOV #TESTADD,R1
          CLR DATBUF
          CLR DATBUF+2
          MOV #1,SBEMSK
          CLR SBEMSK+2
          MOV #1,DBEMSK
          CLR DBEMSK+2
          MOV DATBUF,TSTDAT
          MOV DATBUF+2,TSTDAT+2
          TSTB PASFLG ;NO COMPLEMENTING FIRST PASS
          BEQ 5$
          COM TSTDAT ;COMP FIRST WORD
          COM TSTDAT+2 ;SECOND WORD
          CLR1CSR ;CLEAR 1 SELECTED CSR
          CMP SBEMSK,DBEMSK ;CAN'T HAVE THE SAME ERROR BIT SET
          BNE 6$ ;IN BOTH MASKS
          CMP SBEMSK+2,DBEMSK+2 ;COULD BE EQUAL IN SECOND WORD
          BEQ 13$ ;GO MAKE THEM NOT EQUAL
          MOV #TSTDAT,SOURCE ;SOURCE ADDRESS FOR CHKGEN
          CALL CHKGEN ;GO GENERATE CHECK BITS
          MOV SBEMSK,R2
          XOR R2,TSTDAT
          MOV SBEMSK+2,R2
          XOR R2,TSTDAT+2
          MOV DBEMSK,R2
          XOR R2,TSTDAT
          MOV DBEMSK+2,R2
          XOR R2,TSTDAT+2
          16$: ECC1DIS ;DISABLE ECC ON 1 SELECTED CSR
          MOV TSTDAT,@(R1)+ ;WRITE 16 BITS
          CE1CSR ;WRITE GENERATED CHECKBITS IN 1 SELECTED CSR
          MOV TSTDAT+2,@(R1) ;WRITE HIGH WORD
          CLR1CSR ;CLEAR 1 SELECTED CSR
          SUB #2,R1 ;ADJUST TEST ADDRESS
          TST @(R1) ;READ THE LOCATION
          WAS1DBE ;WAS THERE ANY DOUBLE BIT ERRORS ON 1 SELECTED CSR
          BCS 9$ ;IT SHOULD BE SET
          SET HEADER
          MOV (R1),ADDRESS
          ERROR +30
          SET HEADER

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 279  
MTP013 CREATE DOUBLE BIT ERROR TEST

```

9422 032016 104512          9$:  ERGEN
9423 032020 105737 002262    TSTB  'ASFLG
9424 032024 100452          BMI  14$
9425 032026 005737 002256    13$: TST  DBEMSK+2      ;CHECK MASK FOR LAST BIT
9426 032032 100405          BMI  10$      ;MINUS = BIT31
9427 032034          DLEFT DBEMSK
9428 032044 000662          BR  4$
9429 032046          10$:  IF #SW11 SET.IN @SWR THEN GOTO 11$
9430 032056          IF QVFLAG IS TRUE THEN GOTO 11$
9431 032064 005737 002252    TST  SBEMSK+2      ;CHECK SINGLE ERROR MASK TOO
9432 032070 100405          BMI  11$      ;BR IF DONE
9433 032072          DLEFT SBEMSK
9434 032102 000636          BR  3$
9435 032104 105737 002262    11$: TSTB  PASFLG  ;FIRST PASS
9436 032110 001003          BNE  12$      ;NON ZERO MEANS WE'RE DONE
9437 032112 105237 002262    INCB  PASFLG  ;FIRST PASS, NOT DONE
9438          ;CLEAR OUT ANY DBE'S OR SBE'S
9439 032116 000617          BR  1$      ;KFEP GOING
9440 032120 052737 000200 002262 12$: BIS  #BIT7,PASFLG  ;SET UP FOR CHECK BIT TEST
9441 032126 005037 002244    CLR  TSTDAT
9442 032132 005037 002246    CLR  TSTDAT+2
9443 032136 012737 000040 002250    MOV  #40,SBEMSK
9444 032144 012737 000100 002254    MOV  #100,DBEMSK
9445 032152 012737 003740 002310 14$: MOV  #3740,CHECK
9446 032160 013702 002250    MOV  SBEMSK,R2
9447 032164 074237 002310    XOR  R2,CHECK
9448 032170 013702 002254    MOV  DBEMSK,R2
9449 032174 074237 002310    XOR  R2,CHECK
9450 032200 006337 002254    ASL  DBEMSK
9451 032204 032737 020000 002254    BIT  #BIT13,DBEMSK
9452 032212 001652          BEQ  16$
9453 032214 006337 002250    ASL  SBEMSK
9454 032220 032737 004000 002250    BIT  #BIT11,SBEMSK
9455 032226 001006          BNE  15$
9456 032230 013737 002250 002254    MOV  SBEMSK,DBEMSK
9457 032236 006337 002254    ASL  DBEMSK
9458 032242 000743          BR  14$
9459 032244 104471          15$: ECC1DIS      ;DISABLE ECC ON 1 SELECTED CSR
9460 032246 012701 002406    MOV  #TESTADD,R1
9461 032252          CLEAR  @ (R1)+,@ (R1)
9462 032260 104503          CLR1CSR      ;CLEAR 1 SELECTED CSR
9463 032262 000207          RETURN

```



MTP014: SUBTST &lt;&lt;MTP014 BASIC DOUBLE BIT ERROR TEST&gt;&gt;

```
*****
;SUBTEST      MTP014  BASIC DOUBLE BIT ERROR TEST
*****
```

```

: THIS TEST CHECKS THAT A DOUBLE ERROR WILL BE DETECTED
: A BYTE WRITE WITH A DOUBLE ERROR ON A MS11-P
: WILL BE ABOTRED.

```

9468				
9469				
9470				
9471				
9472				
9473	032264	104424		
9474	032266			
9475	032272			
9476	032300			
9477	032306	104513		
9478	032310			
9479	032316	104425		
9480	032320			
9481	032326			
9482	032330	005711		
9483	032332			
9484	032342	104055		
9485	032344			
9486	032344	104426		
9487	032346	042737	020000	002146
9488	032354			
9489	032364			
9490	032372			
9491	032400	104065		
9492	032402			
9493	032402	104473		
9494	032404	005037	002264	
9495	032410			
9496	032410	104473		
9497	032412	005237	002264	
9498	032416	005037	002070	
9499	032422			
9500	032426	105711		
9501	032430			
9502	032440			
9503	032446			
9504	032452			
9505	032460	104056		
9506	032462			
9507	032462	005201		
9508	032464			
9509	032474	005041		
9510	032476	104503		
9511	032500	005037	002070	
9512	032504	104423		
9513	032506	000207		

```

CACHOFF
LET PARCNT := #0
LET NOPAR := #1
LET ADDRESS := #FIRST
CBREG
LET CSR := #3145
LOADCSR
LET GOOD := #103145
LET (R1) := #0
TST (R1)
IF PARCNT NE #1
    ERROR +55
END
READCSR
BIC #BIT13,CSR
IF CSR NE GOOD THEN
    SET HEADER
    LET BAD := CSR
    ERROR +65
END
ECC1INIT
CLR PASSNO
REPEAT
    ECC1INIT
    INC PASSNO
    CLR PARCNT
    LET (R1) := #377
    TSTB (R1)
    IF PARCNT NE #1
        SET HEADER
        LET GOOD := #0
        LET BAD := #377
        ERROR +56
    END
    INC R1
UNTIL PASSNO EQ #2
CLR -(R1)
CLR1CSR
CLR PARCNT
CACHON
RETURN
;TURN OFF CACHE
;CLEAR PARCNT
;SET PARITY ACTION
;SET ADDRESS FOR ERROR REPORT
;ENABLE CHECK/SYNDROME BIT REGISTER
;DBE CHECK BITS FOR CSR
;WRITE DBE CHECK BITS TO CSR
;GOOD DATA
;WRITE ZEROS AND DBL ERROR CHK BITS A=0
;READ A=0 TO GET DOUBLE BIT ERROR
;WAS BUSPBL ASSERTED???
;ERROR CALL ;:MISSED EXPECTED TRAP
;
;READ CSR FOR CORRECT CHECK BITS AND DBE INDICATOR
;CLEAR INHIBIT MODE POINTER FROM DATA IF IT EXISTS1
;CHECK IF DOUBLE ERROR BIT IS SET
;
;BAD DATA
;
;
;ENABLE BUSPBL
;CLEAR LOOP COUNTER
;
;ENABLE BUSPBL
;INCREMENT LOOP COUNTER
;CLEAR PARITY ACTION COUNTER
;WRITE BYTE SHOULD BE ABORTED
;READ R1 TO SEE IF IT IS STILL 0
;WAS WRITE ABORTED???
;
;GOOD DATA
;BAD DATA
;
;AND REPEAT ON HIGH BYTE
;
;CLEAR LUT
;CLEAR CSR
;CLEAR PARITY TRAP COUNTER
;TURN ON CACHE

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 283  
MTP014 BASIC DOUBLE BIT ERROR TEST

9515 032510

```

MTP015: SUBTST <<MTP015      WRITE INHIBIT OF BYTE WITH DBE>>
:*****
:*SUBTEST      MTP015  WRITE INHIBIT OF BYTE WITH DBE
:*****
:CHECK FOR WRITE INHIBIT DURING A WRITE BYTE.
:CHECKS FOR UNCORRECTED DATA.
1$: CLR      DATBUF      ;INITIAL DATA
   CLR      DATBUF+2    ;32 BITS WORTH
2$: MOV      #1,SBEMSK   ;SINGLE ERROR MASK
   CLR      SBEMSK+2    ;
3$: MOV      #1,DBEMSK   ;DOUBLE ERROR MASK
   CLR      DBEMSK+2    ;
4$: MOV      DATBUF,TSTDAT ;PRESERVE ORIG DATA
   MOV      DATBUF+2,TSTDAT+2
   TSTB     PASFLG      ;WHICH PASS ?
   BEQ      5$          ;FIRST PASS, NO COMPLEMENTING
   COM      TSTDAT
   COM      TSTDAT+2    ;SECOND PASS, COMPLEMENT TSTDAT
5$: CLR1CSR             ;CLEAR 1 SELECTED CSR
   CMP      SBEMSK,DBEMSK ;CHECK FOR SAME MASKS
   BNE      6$          ;BR IF NOT EQUAL
   CMP      SBEMSK+2,DBEMSK+2 ;SECOND WORD ALSO
   BEQ      11$         ;BR TO MAKE THEM NOT EQUAL
6$: MOV      #TSTDAT,SOURCE ;ADDRESS FOR CHKGEN
   CALL     CHKGEN      ;GO GENERATE CHECK BITS
   MOV      SBEMSK,R1
   XOR      R1,TSTDAT
   MOV      SBEMSK+2,R1
   XOR      R1,TSTDAT+2
   MOV      DBEMSK,R1
   XOR      R1,TSTDAT
   MOV      DBEMSK+2,R1
   XOR      R1,TSTDAT+2
7$: MOV      #TESTADD,R1 ;TEST LOCATION
   ECC1DIS             ;DISABLE ECC ON 1 SELECTED CSR
   MOV      TSTDAT,@(R1)+ ;WRITE FIRST 16 BITS
   ;LOAD CSR WITH IMAGE FROM R2
   CB1CSR              ;WRITE GENERATED CHECKBITS IN 1 SELECTED CSR
   MOV      TSTDAT+2,@(R1) ;WRITE SECOND 16 BITS + CHECKBITS
   CLR1CSR             ;CLEAR 1 SELECTED CSR
   MOV      TESTADD,R2  ;GET ADDRESS OF TEST LOC
   MOV      R2,R3       ;R2 DESIGNATES FIRST BYTE
   ADD      #3,R3       ;R3 DESIGNATES LAST BYTE
8$: MOVB     #360,(R2)+ ;TRY WRITING A BYTE
   MOV      #TESTADD,R1
   MOV      @(R1),R0
   CMP      TSTDAT,R0   ;CHECK FOR UNCHANGED DATA
   BEQ      9$          ;BR IF OK
   MOV      @(R1),ADDRESS
   PERR31
9$: MOV      @2(R1),R0
   CMP      TSTDAT+2,R0 ;READ SECOND WORD
   BEQ      10$         ;BR IF UNCHANGED
   MOV      @2(R1),ADDRESS
   PERR31

```

```

9516
9517
9518 032510 005037 002240
9519 032514 005037 002242
9520 032520 012737 000001 002250
9521 032526 005037 002252
9522 032532 012737 000001 002254
9523 032540 005037 002256
9524 032544 013737 002240 002244
9525 032552 013737 002242 002246
9526 032560 105737 002262
9527 032564 001404
9528 032566 005137 002244
9529 032572 005137 002246
9530 032576 104503
9531 032600 023737 002250 002254
9532 032606 001004
9533 032610 023737 002252 002256
9534 032616 001474
9535 032620 012737 002244 002306
9536 032626 004737 044366
9537 032632 013701 002250
9538 032636 074137 002244
9539 032642 013701 002252
9540 032646 074137 002246
9541 032652 013701 002254
9542 032656 074137 002244
9543 032662 013701 002256
9544 032666 074137 002246
9545 032672 012701 002406
9546 032676 104471
9547 032700 013731 002244
9548
9549 032704 104475
9550 032706 013771 002246 000000
9551 032714 104503
9552 032716 013702 002406
9553 032722 010203
9554 032724 062703 000003
9555 032730 112722 000360
9556 032734 012701 002406
9557 032740 017100 000000
9558 032744 023700 002244
9559 032750 001404
9560 032752 017137 000000 002032
9561 032760 104455
9562
9563 032762 017100 000002
9564 032766 023700 002246
9565 032772 001404
9566 032774 017137 000002 002032
9567 033002 104455
9568

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 283-1  
MTP015 WRITE INHIBIT OF BYTE WITH DBE

9569	033004	020203		10\$:	CMP	R2,R3		:TESTED LAST BYTE ?
9570	033006	001350			BNE	8\$		:BR IF NO
9571	033010	105737	002262	11\$:	TSTB	PASFLG		
9572	033014	100452			BMI	15\$		:BRANCH IF TESTING CHECK BITS
9573	033016	005737	002256		TST	DBEMSK+2		:CHECKING FOR LAST ERROR BIT
9574	033022	100405			BMI	12\$		:BR IF DONE HERE
9575	033024				DLEFT	DBEMSK		
9576	033034	000643			SR	4\$		
9577	033036			12\$:	IF	#SW11 SET.IN @SWR THEN GOTO 13\$		
9578	033046				IF	QVFLAG IS TRUE THEN GOTO 13\$		
9579	033054	005737	002252		TST	SBEMSK+2		:LAST SBE MASK
9580	033060	100405			BMI	13\$		:BR IF DONE WITH THIS PASS
9581	033062				DLEFT	SBEMSK		
9582	033072	000617			BR	3\$		
9583	033074	105737	002262	13\$:	TSTB	PASFLG	:TEST PASS FLAG	
9584	033100	001003			BNE	14\$		:NON ZERO MEANS WE'RE DONE
9585	033102	105237	002262		INCB	PASFLG	:NOT DONR	
9586	033106	000600			BR	1\$		
9587	033110	052737	000200	002262	14\$:	BIS	#BIT7,PASFLG	
9588	033116	005037	002244		CLR	TSTDAT		
9589	033122	005037	002246		CLR	TSTDAT+2		
9590	033126	012737	000040	002250	MOV	#40,SBEMSK		
9591	033134	012737	000100	002254	MOV	#100,DBEMSK		
9592	033142	012737	003740	002310	15\$:	MOV	#3740,CHECK	
9593	033150	013702	002250		MOV	SBEMSK,R2		
9594	033154	074237	002310		XOR	R2,CHECK		
9595	033160	013702	002254		MOV	DBEMSK,R2		
9596	033164	074237	002310		XOR	R2,CHECK		
9597	033170	006337	002254		ASL	DBEMSK		
9598	033174	032737	020000	002254	BIT	#BIT13,DBEMSK		
9599	033202	001633			BEQ	7\$		
9600	033204	006337	002250		ASL	SBEMSK		
9601	033210	032737	004000	002250	BIT	#BIT11,SBEMSK		
9602	033216	001006			BNE	16\$		
9603	033220	013737	002250	002254	MOV	SBEMSK,DBEMSK		
9604	033226	006337	002254		ASL	DBEMSK		
9605	033232	000743			BR	15\$		
9606	033234	104471		16\$:	ECC1DIS		:DISABLE ECC ON 1 SELECTED CSR	
9607	033236	012701	002406		MOV	#TESTADD,R1	:TEST LOCATION	
9608	033242				CLEAR	@(R1)+,@(R1)	:TO ERASE ANY DBE'S FROM TESTING	
9609					:RESTORE CSR			
9610	033250	104503			CLR1CSR		:CLEAR 1 SELECTED CSR	
9611	033252	000207			RETURN			

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 285  
MTP015 WRITE INHIBIT OF BYTE WITH DBE

9614 033254

MTP016: SUBTST &lt;&lt;MTP016 WRITE INHIBIT OF WORD WITH DBE&gt;&gt;

\*\*\*\*\*  
:SUBTEST MTP016 WRITE INHIBIT OF WORD WITH DBE  
\*\*\*\*\*

9615

9616

9617

9618

9619 033254 005037 002240

9620 033260 005037 002242

9621 033264 012737 000001 002250

9622 033272 005037 002252

9623 033276 012737 000001 002254

9624 033304 005037 002256

9625 033310 013737 002240 002244

9626 033316 013737 002242 002246

9627 033324 105737 002262

9628 033330 001404

9629 033332 005137 002244

9630 033336 005137 002246

9631 033342 023737 002250 002254

9632 033350 001004 002252 002256

9633 033352 023737 002252 002256

9634 033360 001502

9635 033362 012737 002244 002306

9636 033370 004737 044366

9637 033374 013701 002250

9638 033400 074137 002244

9639 033404 013701 002252

9640 033410 074137 002246

9641 033414 013701 002254

9642 033420 074137 002244

9643 033424 013701 002256

9644 033430 074137 002246

9645 033434 012701 002406

9646 033440 104471

9647 033442 013731 002244

9648 033446 104475

9649 033450 013771 002246 000000

9650 033456 105037 002263

9651 033462 162701 000002

9652 033466 104503

9653 033470 012771 177400 000000

9654 033476 012701 002406

9655 033502 017100 000000

9656 033506 023700 002244

9657 033512 001404

9658 033514 017137 000000 002032

9659 033522 104455

9660

9661 033524 062701 000002

9662 033530 017100 000000

9663 033534 023700 002246

9664 033540 001404

9665 033542 017137 000000 002032

9666 033550 104455

:DOUBLE BIT ERROR WRITE CANCEL WITH

:WORD WRITE.

:CHECKS WRITE INHIBIT WITH WORD WRITES TO

:WORD WITH DOUBLE ERROR.

T12A: CLR DATBUF ;BACKGROUND FOR DOUBLE ERRORS

CLR DATBUF+2 ;2 WORDS WORTH

MOV #1,SBEMSK ;SINGLE ERROR MASK

CLR SBEMSK+2 ;

T12B: MOV #1,DBEMSK ;DOUBLE ERROR MASK

CLR DBEMSK+2 ;

1\$: MOV DATBUF,TSTDAT ;DATA FOR TEST

MOV DATBUF+2,TSTDAT+2 ;BOTH WORDS

TSTB PASFLG ;COMP DATA ON SECOND PASS ONLY

BEQ 2\$ ;BR IF FIRST PASS

COM TSTDAT ;COMP FIRST WORD

COM TSTDAT+2 ;NOW SECOND WORD

2\$: CMP SBEMSK,DBEMSK ;CHECK FOR IDENTICAL MASKS

BNE 3\$ ;BR IF DIFFERENT

CMP SBEMSK+2,DBEMSK+2 ;UPPER WORD TOO

BEQ 8\$ ;BR TO MAKE THEM NOT EQUAL

3\$: MOV #TSTDAT,SOURCE ;NEED ADDR OF DATA FOR CHKGFN

CALL CHKGEN ;GO GENERATE CHECK BITS

MOV SBEMSK,R1

XOR R1,TSTDAT

MOV SBEMSK+2,R1

XOR R1,TSTDAT+2

MOV DBEMSK,R1

XOR R1,TSTDAT

MOV DBEMSK+2,R1

XOR R1,TSTDAT+2

4\$: MOV #TESTADD,R1

ECC1DIS ;FIRST TEST ADDRESS

MOV TSTDAT,@(R1)+ ;DISABLE ECC ON 1 SELECTED CSR

CB1CSR ;WRITE FIRST 16 BITS

MOV TSTDAT+2,@(R1) ;WRITE GENERATED CHECKBITS IN 1 SELECTED CSR

UPPFLG ;WRITE SECOND 16 BITS + CHECKBITS

SUB #2,R1 ;SET FOR 2 LOOPS

5\$: CLR1CSR ;POINT TO LOW WORD

MOV #177400,@(R1) ;CLEAR 1 SELECTED CSR

MOV #TESTADD,R1 ;TRY WRITING LOCATION

MOV @(R1),R0

CMP TSTDAT,R0

BEQ 6\$ ;CHECK FOR ORIGINAL DATA

MOV @(R1),ADDRESS ;SHOULD BE UNCHANGED

PERR31

6\$: ADD #2,R1

MOV @(R1),R0

CMP TSTDAT+2,R0

BEQ 7\$ ;THIS SHOULD BE UNCHANGED ALSO

MOV @(R1),ADDRESS

PERR31

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 287  
MTP016 WRITE INHIBIT OF WORD WITH DBE

9669	033552	105737	002263	7\$:	TSTB	UPPFLG		:WHICH LOOP ?
9670	033556	001003			BNE	8\$		:SECOND, BR OUT
9671	033560	105237	002263		INCB	UPPFLG		:FIRST, KEEP GOING
9672	033564	000740			BR	5\$		
9673	033566	105737	002262	8\$:	TSTB	PASFLG		
9674	033572	100454			BMI	12\$		
9675	033574	005737	002256		TST	DBEMSK+2		:LAST BIT ?
9676	033600	100405			BMI	9\$		:MINUS = BIT 31
9677	033602				DLEFT	DBEMSK		
9678	033612	000636			BR	1\$		
9679	033614			9\$:	IF #SW11 SET.IN 3SWR THEN GOTO 10\$			
9680	033624				IF QVFLAG IS TRUE THEN GOTO 10\$			
9681	033632	005737	002252		TST	SBEMSK+2		:LAST BIT IN THIS MASK ?
9682	033636	100406			BMI	10\$		:BR IF LAST BIT
9683	033640				DLEFT	SBEMSK		
9684	033650	000137	033276		JMP	T12B		
9685	033654	105737	002262	10\$:	TSTB	PASFLG	:FIRST PASS ?	
9686	033660	001004			BNE	11\$	:BR IF SECOND	
9687	033662	105237	002262		INCB	PASFLG	:INDICATE SECOND PASS COMING	
9688	033666	000137	033254		JMP	T12A		
9689	033672	052737	000200	002262	11\$:	BIS	#BIT7,PASFLG	
9690	033700	005037	002244		CLR	TSTDAT		
9691	033704	005037	002246		CLR	TSTDAT+2		
9692	033710	012737	000040	002250	MOV	#40,SBEMSK		
9693	033716	012737	000100	002254	MOV	#100,DBEMSK		
9694	033724	012737	003740	002310	12\$:	MOV	#3740,CHECK	
9695	033732	013702	002250		MOV	SBEMSK,R2		
9696	033736	074237	002310		XOR	R2,CHECK		
9697	033742	013702	002254		MOV	DBEMSK,R2		
9698	033746	074237	002310		XOR	R2,CHECK		
9699	033752	006337	002254		ASL	DBEMSK		
9700	033756	032737	020000	002254	BIT	#BIT13,DBEMSK		
9701	033764	001623			BEQ	4\$		
9702	033766	006337	002250		ASL	SBEMSK		
9703	033772	032737	004000	002250	BIT	#BIT11,SBEMSK		
9704	034000	001006			BNE	13\$		
9705	034002	013737	002250	002254	MOV	SBEMSK,DBEMSK		
9706	034010	006337	002254		ASL	DBEMSK		
9707	034014	000743			BR	12\$		
9708	034016	104471		13\$:	ECC1DIS		:DISABLE ECC ON 1 SELECTED CSR	
9709	034020	012701	002406		MOV	#TESTADD,R1	:RESTORE TEST ADDRESS	
9710	034024	005031			CLR	@(R1)+	:CLEAR ANY DBE'S FROM TEST	
9711	034026	005071	000000		CLR	@(R1)		
9712	034032	104503			CLR1CSR		:CLEAR 1 SELECTED MK11 CSR	
9713	034034	000207			RETURN			

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 289  
MTP016 WRITE INHIBIT OF WORD WITH DBE

9716 034036

```

MTP017: SUBTST <<MTP017      HOLDING 1'S & 0'S TEST>>
*****
;SUBTEST      MTP017  HOLDING 1'S & 0'S TEST
*****
;*(1)  THIS TEST CHECKS THE MEMORY FOR THE CAPABILITY
;      OF HOLDING 1'S AND 0'S BY WRITING A BACKGROUND
;      OF 000377 AND READING IT
;*(2)  MEMORY IS WRITTEN USING A BYTE AT A TIME
;*(3)  STEPS 1 & 2 ARE REPEATED WITH A SWAPPED BACKGROUND PATTERN
;NOTE:  THIS TEST WRITES BYTES & READS WORDS
MOV     #FIRST,R1
MOV     R1,R4
MOV     #LAST+2,R5
MOV     #377,R0      ;GET THE PATTERN INTO R0
MOV     R0,R3
SWAB    R3
1$:     MOVB    R0,(R1)+    ;WRITE A BYTE
        MOVB    R3,(R1)+    ;WRITE THE MEMORY WITH THE BYTE STORED IN BAKPAT+1
        CMP     R1,R5      ;COMPARE TEST LOC TO TOP + 2
        BLO     1$         ;BRANCH IF LOWER

2$:     MOV     -(R1),R2
        CMP     R0,R2      ;TEST THE MEMORY TO SEE IF IT CONTAINS
                        ;THE WORD STORED IN BAKPAT
        BEQ     3$
        PERR22

3$:     CMP     R1,R4      ;KEEP ON TESTING THE MEMORY UNTIL
        BHI     2$         ;R1 EQUALS THE LOWEST ADDRESS
        SWAB    R3      ;CHANGE THE DATA PATTERN
        SWAB    R0
        BEQ     1$         ;IF THE DATA PATTERN DOES NOT HAVE LOW
                        ;BYTE =0 THEN FALL THRU

        RETURN

```

9717  
9718  
9719  
9720  
9721  
9722  
9723 034036 012701 060000  
9724 034042 010104  
9725 034044 012705 160000  
9726 034050 012700 000377  
9727 034054 010003  
9728 034056 000303  
9729 034060 110021  
9730 034062 110321  
9731 034064 020105  
9732 034066 103774  
9733  
9734 034070 014102  
9735 034072 020002  
9736  
9737 034074 001401  
9738 034076 104446  
9739  
9740 034100 020104  
9741 034102 101372  
9742 034104 000303  
9743 034106 000300  
9744 034110 001763  
9745  
9746 034112 000207

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 290  
MTP017 HOLDING 1'S & 0'S TEST

9748

LZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 291  
MTP017 HOLDING 1'S & 0'S TEST

9750 034114

MTP020: SUBTST &lt;&lt;MTP020 SYNDROMES TO CSR ON SINGLE BIT ERROR TEST&gt;&gt;

\*\*\*\*\*  
:SUBTEST MTP020 SYNDROMES TO CSR ON SINGLE BIT ERROR TEST  
\*\*\*\*\*

9751  
9752  
9753  
9754  
9755  
9756 034114 104424  
9757 034116 005000  
9758 034120 105037 002262  
9759 034124 104513  
9760 034126  
9761 034126  
9762 034132  
9763 034136  
9764 034142  
9765 034152  
9766 034156  
9767 034160  
9768 034164  
9769 034164  
9770 034164 005237 002320  
9771 034170  
9772 034172  
9773 034176 072227 000005  
9774 034202 052702 000004  
9775 034206  
9776 034212 104425  
9777 034214  
9778 034216 104503  
9779 034220 005711  
9780 034222 104426  
9781 034224 042737 177757 002146  
9782 034232  
9783 034242  
9784 034250  
9785 034256 104060  
9786 034260  
9787 034260 104513  
9788 034262 104426  
9789 034264 042737 174033 002146  
9790 034272  
9791 034276 072327 000005  
9792 034302 052703 000004  
9793 034306  
9794 034314  
9795 034322  
9796 034326  
9797 034334 104042  
9798 034336  
9799 034336 005011  
9800 034340  
9801 034350 006305  
9802 034352  
9803 034354 000261

```

:
: THIS TEST CHECKS TO SEE IF THE SINGLE BIT ERRORS CAUSE THE SBE
: BIT IN THE CSR TO BE SET AND CORRECT SYNDROME BITS ARE GENERATED FOR
: ALL 16 DATA BITS.
:
: CACHOFF
: CLR R0
: CLRB PASFLG
: CBREG
: REPEAT
:   LET PASFLG :B= PASFLG + #1
:   LET R4 := #-1
:   LET BITNO := #0
:   IFB PASFLG EQ #1
:     LET R5 := #1
:   ELSE
:     LET R5 := #177776
:   END
:   REPEAT
:     INC BITNO
:     LET R4 := R4 + #1
:     LET R2 :B= PTABLE(R4)
:     ASH #5,R2
:     BIS #BIT2,R2
:     LET CSR := R2
:     LOADCSR
:     LET (R1) := R0
:     CLR1CSR
:     TST (R1)
:     READCSR
:     BIC #^C20,CSR
:     IF CSR NE #20
:       LET GOOD := #20
:       LET BAD := CSR
:       ERROR +60
:     END
:     CBREG
:     READCSR
:     BIC #^C3744,CSR
:     LET R3 :B= SBESYN(R4)
:     ASH #5,R3
:     BIS #BIT2,R3
:     IF R3 NE CSR
:       SET HEADER
:       LET GOOD := R3
:       LET BAD := CSR
:       ERROR +42
:     END
:     CLR (R1)
:     IFB PASFLG EQ #1
:       ASL R5
:     ELSE
:       SEC
:   END
:
: TURN OFF CACHE
: CLEAR DATA
: CLEAR PASFLG
: ENABLE CHECK/SYNDROME BIT REGISTER
:
: INCREMENT LOOP COUNTER
: INDEX TO SINGLE BIT ERROR TABLE
: CLEAR INNER LOOP COUNTER
: SELECT DATA TO BE CORRECTED BY PASSNO
: DATA=0;BIT TO BE CORRECTED IS A ONE
:
: DATA=177776;BIT TO BE CORRECTED IS A ZERO
:
: INCREMENT BIT POINTER
: POINT TO NEXT SET OF CHECK BITS
: GET NEXT SET OF CHECK BITS
: SHIFT TO LINE UP IN CSR
: ENABLE DIAG MODE
: GET CHECK BITS TO BE WRITTEN
: LOAD CSR WITH DATA
: WRITE DATA TO TEST ADDRESS
: CLEAR CSR
: CORRECT SBE
: READ CSR FOR CORRECT SEE BIT AND SYNDROMES
: CLEAR ALL BUT SBE INDICATOR
: WAS DATA CORRECTED???
:
: NO ERROR
:
: ENABLE SYNDROME BIT REGISTER
: GET SYNDROMES FROM CSR
: MASK SYNDROME BITS
: GET GOOD SYNDROMES
: SHIFT INTO POSITION
: SET DIAG MODE IN DATA
: DO SYNDROME BITS AGREE
:
:
:
: CLEAR LUT
: SHIFT NEW DATA DEPENDING ON PASFLG
: SHIFT BITNO TO THE LEFT
:
: SET CARRY BIT AND.....

```



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 291-1  
 MTP020 SYNDROMES TO CSR ON SINGLE BIT ERROR TEST

9804	034356	006105				ROL R5	;ROTATE LEFT
9805	034360					END	:
9806	034360					UNTIL BITNO EQ #16.	:UNTIL ALL BITS ARE DONE
9807	034370	005100				COM R0	:COMPLEMENT DATA AND REPEAT
9808	034372					UNTILB PASFLG EQ #2	:UNTIL 2 PASSES ARE COMPLETE!
9809	034402	104503				CLR1CSR	:CLEAR CSR
9810	034404	104423				CACHON	:TURN CACHE
9811	034406	000207				RETURN	
9812						:	
9813						:	
9814						:	
9815	034410	016	013	023	SBESYN: .BYTE	16,13,23,25,26,31,32,34,43,45,46,51,52,54,61,64	
	034413	025	026	031			
	034416	032	034	043			
	034421	045	046	051			
	034424	052	054	061			
	034427	064					
9816							

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 292  
 MTP020 SYNDROMES TO CSR ON SINGLE BIT ERROR TEST

9818 034430

MTPA21: SUBTST &lt;&lt;MTPA21 MARCHING 1'S &amp; 0'S PATTERN TEST&gt;&gt;

\*\*\*\*\*  
 :SUBTEST MTPA21 MARCHING 1'S & 0'S PATTERN TEST  
 \*\*\*\*\*

9819  
 9820 034430 014100  
 9821 034432 020200  
 9822 034434 001401  
 9823 034436 104443  
 9824  
 9825 034440 000311  
 9826 034442 011100  
 9827 034444 020300  
 9828 034446 001401  
 9829 034450 104444

1\$: :READ,BYTESWAP-MODIFY,READ,DOWN  
 MOV -(R1),R0 :V177640  
 CMP R2,R0 :V177642  
 BEQ 2\$ :V177644  
 PERR17 :V177646

2\$: SWAB (R1) :V177650  
 MOV (R1),R0 :V177652  
 CMP R3,R0 :V177654  
 BEQ 3\$ :V177656  
 PERR20 :V177660

3\$: CMP R4,R1 :V177662 :DONE?  
 BNE 1\$ :V177664 :NO - LOOP  
 RETURN :V177666 :YES - RETURN

9830  
 9831 034452 020401  
 9832 034454 001365  
 9833 034456 000207

9834  
 9835 034460  
 9836 034460 011100  
 9837 034462 020300  
 9838 034464 001401  
 9839 034466 104444

MTPB21: :READ,BYTESWAP-MODIFY,READ,UP  
 1\$: MOV (R1),R0 :V177640  
 CMP R3,R0 :V177642  
 BEQ 2\$ :V177644  
 PERR20 :V177646

2\$: SWAB (R1) :V177650  
 MOV (R1),R0 :V177652  
 CMP R2,R0 :V177654  
 BEQ 3\$ :V177656  
 PERR17 :V177660

9840  
 9841 034470 000311  
 9842 034472 011100  
 9843 034474 020200  
 9844 034476 001401  
 9845 034500 104443

3\$: ADD #2,R1 :V177662  
 CMP R5,R1 :V177666 :DONE?  
 BNE 1\$ :V177670 :NO - LOOP  
 RETURN :V177672 :YES - RETURN

9846  
 9847 034502 062701 000002  
 9848 034506 020501  
 9849 034510 001363  
 9850 034512 000207

9851  
 9852 034514  
 9853 034514 011100  
 9854 034516 020200  
 9855 034520 001401  
 9856 034522 104443

MTPC21: :READ,BYTESWAP-MODIFY,READ,UP  
 1\$: MOV (R1),R0 :V177640  
 CMP R2,R0 :V177642  
 BEQ 2\$ :V177644  
 PERR17 :V177646

2\$: SWAB (R1) :V177650  
 MOV (R1),R0 :V177652  
 CMP R3,R0 :V177654  
 BEQ 3\$ :V177656  
 PERR20 :V177660

9857  
 9858 034524 000311  
 9859 034526 011100  
 9860 034530 020300  
 9861 034532 001401  
 9862 034534 104444

3\$: ADD #2,R1 :V177662  
 CMP R5,R1 :V177666 :DONE?  
 BNE 1\$ :V177670 :NO - LOOP  
 RETURN :V177672 :YES - RETURN

9863  
 9864 034536 062701 000002  
 9865 034542 020501  
 9866 034544 001363  
 9867 034546 000207

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 294  
MTPA21 MARCHING 1'S & 0'S PATTERN TEST

```
9870 034550
9871 034550 014100
9872 034552 020300
9873 034554 001401
9874 034556 104444
9875
9876 034560 000311
9877 034562 011100
9878 034564 020200
9879 034566 001401
9880 034570 104443
9881
9882 034572 020401
9883 034574 001365
9884 034576 000207
9885
```

MTPD21: ;READ,BYTESWAP-MODIFY,READ,DOWN

```
1$:  MOV    -(R1),R0;V177640
      CMP    R3,R0  ;V177642
      BEQ    2$     ;V177644
      PERR20        ;V177646

2$:  SWAB    (R1)    ;V177650
      MOV    (R1),R0 ;V177652
      CMP    R2,R0  ;V177654
      BEQ    3$     ;V177656
      PERR17        ;V177660

3$:  CMP    R4,R1   ;V177662      ;DONE?
      BNE    1$     ;V177664      ;NO - LOOP
      RETURN        ;V177666      ;YES - RETURN
```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 296  
MTPA21 MARCHING 1'S & 0'S PATTERN TEST

9888 034600

MTP022: SUBTST <<MTP022 REFRESH & SHIFTING DIAGONAL TEST>>  
\*\*\*\*\*  
:SUBTEST MTP022 REFRESH & SHIFTING DIAGONAL TEST  
\*\*\*\*\*  
:(1) WE WRITE A DIAGONAL PATTERN IN MEMORY (WITH CACHE ON).  
:(2) IF A REFRESH TEST WE DISTURB ALL ROWS FOR > 2 MS (WITH CACHE ON).  
:(3) WE READ & CHECK FOR CORRECTNESS THE DIAGONAL PATTERN  
:(WITH CACHE OFF).  
KDIAG=8. ;HOW OFTEN A DIAGONAL STRIPE OCCURS (MUST BE A POWER OF 2)  
FOR EVEN := #1 TO #2 ;FOR DATA & COMPLEMENT DATA

9889  
9890  
9891  
9892  
9893 000010

9894 034600  
9895 034606  
9896 034616  
9897 034622  
9898 034626  
9899 034630  
9900 034634  
9901 034640  
9902 034640

IF EVEN EQ #1  
LET R2 := ZEROS  
LET R3 := ONES  
ELSE  
LET R2 := ONES  
LET R3 := ZEROS  
END ;OF IF EVEN  
FOR STRIPES := #0 TO #KDIAG-1 ;FOR THE NUMBER OF STRIPES

9903  
9904  
9905 034644 104423

9906 034646  
9907 034654  
9908 034660  
9909 034666  
9910 034702  
9911 034714  
9912 034722  
9913 034724  
9914 034730  
9915 034732  
9916 034734  
9917 034740  
9918 034740  
9919 034744  
9920 034750

;WRITE LOOP  
CACHON ;TURN CACHE ON  
LET COUNT := STRIPES  
LET R1 := #FIRST  
WHILE R1 LOS #LAST  
IF COUNT LT #0 THEN LET COUNT := #KDIAG-1  
IF #374 OFF IN R1 THEN LET COUNT := COUNT - #1  
IF COUNT NE #0  
LET (R1) := R2  
LET 2(R1) := R2  
ELSE  
LET (R1) := R3  
LET 2(R1) := R3  
END ;OF IF COUNT  
LET COUNT := COUNT - #1  
LET R1 := R1 + #4  
END ;OF WHILE  
;END OF WRITE LOOP

9921  
9922  
9923 034752  
9924  
9925 034764  
9926 034772  
9927 034776 104424

IF DIAGFLAG IS FALSE THEN \$CALL REFRESH  
;READ LOOP  
LET COUNT := STRIPES  
LET R1 := #FIRST  
CACHOFF ;TURN CACHE OFF

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 297  
 MTP022 REFRESH & SHIFTING DIAGONAL TEST

```

9929 035000      WHILE R1 LOS #LAST
9930 035006      IF COUNT LT #0 THEN LET COUNT := #KDIAG-1
9931 035022      IF #374 OFF.IN R1 THEN LET COUNT := COUNT - #1
9932 035034      IF COUNT NE #0
9933 035042      LET R0 := (R1)
9934 035044      IF R2 NE R0
9935 035050      PERR17
9936 035052      END ;OF IF R2
9937 035052      LET R0 := 2(R1)
9938 035056      IF R2 NE R0
9939 035062      PERR17
9940 035064      END ;OF IF R2
9941 035064      ELSE
9942 035066      LET R0 := (R1)
9943 035070      IF R3 NE R0
9944 035074      PERR20
9945 035076      END ;OF IF R3
9946 035076      LET R0 := 2(R1)
9947 035102      IF R3 NE R0
9948 035106      PERR20
9949 035110      END ;OF IF R3
9950 035110      END ;OF IF COUNT
9951 035110      LET COUNT := COUNT - #1
9952 035114      LET R1 := R1 + #4
9953 035120      END ;OF WHILE
9954          ;END OF READ LOOP
9955
9956 035122      END ;OF FOR STRIPES
9957 035136      END ;OF FOR EVEN
9958 035152      RETURN
9959
9960 035154      REFRESH:SUBTST <<SUBR REFRESH DELAY>>

```

```

:*****
:*SUBTEST      SUBR      REFRESH DELAY
:*****
:DISTURB EACH ROW FOR > 3.2 MS
FOR R0 := #FIRST TO #FIRST+374 BY #4
CALL REFSUB
END ;OF FOR R0
LET R0 := #FIRST+BIT14
WHILE R0 LOS #LAST+BIT14+374
CALL REFSUB
LET R0 := R0 + #4
END ;OF WHILE
RETURN
REFSUB: MOV      #640,R4          ;TIME FOR A > 3.2 MS LOOP
ADD      #2,R0
1$:      COM      -(R0)
COM      (R0)+
COM      (R0)
COM      (R0)
SOB      R4,1$
SUB      #2,R0
RETURN

```

```

9961
9962 035154      004737 035224
9963 035160
9964 035164
9965 035176
9966 035202      004737 035224
9967 035210
9968 035214
9969 035220
9970 035222      000207
9971 035224      012704 000640
9972 035230      062700 000002
9973 035234      005140
9974 035236      005120
9975 035240      005110
9976 035242      005110
9977 035244      077405
9978 035246      162700 000002
9979 035252      000207

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 299  
 SUBR REFRESH DELAY

9982 035254

```

MTPA24: SUBTST <<MTPA24      FAST GALLOPING PATTERN TEST>>
*****
: *SUBTEST      MTPA24  FAST GALLOPING PATTERN TEST
*****
: THE TOTAL TEST (INCLUDING SETUP) IS AS FOLLOWS
: * (1)  THIS TEST WRITES THE MEMORY WITH A BACK GROUND PATTERN
:        STORED AT LOCATION BAKPAT
: * (2)  TEST BEGINS AT LOWEST LOCATION BEING TESTED
:        (LETS NAME IT 'A')
: * (3)  LETS NAME THE 1ST LOCATION IN THE ROW/COLUMN UNDER TEST AS 'B'.
: * (4)  SWAPS BYTES FOR LOCATION 'A'.
: * (5)  READS 'A', READS 'B'
: * (6)  'B' = 'B'+400 (ADDS 64 DOUBLE WORDS TO 'B')
: * (7)  REPEATS STEPS 5 AND 6 UNTIL 'B' IS GREATER THAN THE
: * (8)  END OF THE BANK A+2
: * (9)  REPEATS STEPS 3-8 UNTILL 'A' REACHES THE END OF THE BANK
: * (10) AFTER EXECUTING THE TEST DATA IS COMPLEMENTED
:        AND STEPS 1-9 ARE REPEATED
: REGISTERS ARE USED AS FOLLOWS
: R0     TEST DATA
: R1     'A'
: R2     'B'
: R3     BAKPAT
: R4     SWAPAT
: R5     LAST

```

:NOTE THE PATTERN STARTS AT MTPB24!!!!!!!!!!!!!!

10007							
10008	035254	011100		1\$:	MOV (R1),R0	:V177640	:READ 'A'
10009	035256	020004			CMP R0,R4	:V177642	:CHECK 'A'
10010	035260	001401			BEQ 2\$	:V177644	:BR IF OK
10011	035262	104447			PERR23	:V177646	:REPORT ERROR
10012							
10013	035264	011200		2\$:	MOV (R2),R0	:V177650	:READ 'B'
10014	035266	020003			CMP R0,R3	:V177652	:CHECK 'B'
10015	035270	001401			BEQ 3\$	:V177654	:BR IF OK
10016	035272	104450			PERR24	:V177656	:REPORT ERROR
10017							
10018	035274	062702	000400	3\$:	ADD #400,R2	:V177660	:BUMP 'B'
10019	035300	020205			CMP R2,R5	:V177664	:AT END YET?
10020	035302	101764			BLOS 1\$	:V177666	:BR IF NO
10021							
10022	035304	062701	000002		ADD #2,R1	:V177670	:BUMP 'A'
10023	035310	000137	035314		JMP @MTPB24	:V177674	:GOTO V177260

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 301  
 MTPA24 FAST GALLOPING PATTERN TEST

10026 035314

MTPB24: SUBTST <<MTPB24 FAST GALLOP PART B>>

\*\*\*\*\*  
 :C 9TEST MTPB24 FAST GALLOP PART B  
 :\*\*\*\*\*

10027

10028 035314 010411  
 10029 035316 020105  
 10030 035320 001001  
 10031 035322 000207  
 10032 035324 000137 035330  
 10033  
 10034 035330

:SDPAR'S  
 MOV R4,(R1) ;V172260 ;WRITE 'A'  
 CMP R1,R5 ;V172262 ;DONE?  
 BNE 1\$ ;V172264 ;BR IF NO  
 RETURN ;V172266 ;YES - RETURN  
 1\$: JMP @MTPC24 ;V172270 ;GOTO V172360

MTPC24: SUBTST <<MTPC24 FAST GALLOP PART C>>

\*\*\*\*\*  
 :SUBTEST MTPC24 FAST GALLOP PART C  
 :\*\*\*\*\*

10035

10036 035330 010102  
 10037 035332 011100  
 10038 035334 020004  
 10039 035336 001401  
 10040 035340 104447  
 10041 035342 000137 035274

:KDPAR'S  
 MOV R1,R2 ;V172360 ;RESET 'B' <--- 'A'  
 MOV (R1),R0 ;V172362 ;READ 'A'  
 CMP R0,R4 ;V172364 ;CHECK 'A'  
 BEQ 1\$ ;V172366 ;BR IF OK  
 PERR23 ;V172370 ;REPORT ERROR  
 1\$: JMP @MTPA24+20 ;V172372 ;GOTO V177660

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 303  
MTPC24 FAST GALLOP PART C

10044 035346

```

MTP025: SUBTST <<MTP025      INTERRUPT ENABLE TEST>>
:*****
:*SUBTEST      MTP025  INTERRUPT ENABLE TEST
:*****
      CLR      TSTDAT      ;GENERATE CHECKBITS ON 0,,0
      CLR      TSTDAT+2
      MOV      #TSTDAT,SOURCE
      CALL     CHKGEN
      MOV      #3,NOPAR      ;SETUP PARITY ACTION
      MOV      #TESTADD,R1    ;FIRST TEST ADDRESS
      MOV      #1$,PARTHERE   ;SETUP TRAP DESTINATION
      CALL     MTPA25          ;WRITE DATA & CHECKBITS
      ECC1INIT                ;INITIALIZE 1 SELECTED MK11 CSR
      TST      @2(R1)          ;ACCESS LOCATIONS FOR DBE TRAPS
      TST      @2(R1)
      ;NONE - GOOD - ACCESS FOR SBE TRAPS
      ENA1SBE                ;DISABLE TRAPS ON SBE'S FROM 1 SELECTED CSR
      TST      @2(R1)
      TST      @2(R1)
      BR       2$            ;NONE - GOOD - SKIP
1$: READCSR
      FATAL     27
2$: INC      TSTDAT          ;CHECK FOR CORRECT ACTION ON SBE'S
      CALL     MTPD25        ;IN ALL 4 BYTES
      MOV      #400,TSTDAT
      CALL     MTPD25
      CLR      TSTDAT
      INC      TSTDAT+2
      CALL     MTPD25
      MOV      #400,TSTDAT+2
      CALL     MTPD25
      CLR      TSTDAT+2      ;CHECK FOR CORRECT ACTION ON DBE'S
      MOV      #3,TSTDAT    ;IN ALL 4 BYTES
      CALL     MTPE25
      MOV      #1400,TSTDAT
      CALL     MTPE25
      CLR      TSTDAT
      MOV      #3,TSTDAT+2
      CALL     MTPE25
      MOV      #1400,TSTDAT+2
      CALL     MTPE25
      CLR1CSR                ;CLEAR 1 SELECTED MK11 CSR
      CLR      NOPAR        ;INDICATE PARITY ACTION
      RETURN
MTPD25: CALL     MTPA25      ;WRITE DATA & CHECKBITS
      ECC1DIS                ;DISABLE ECC ON 1 SELECTED CSR
      CALL     MTPB25        ;CHECK FOR NO TRAPS
      ENA1SBE                ;DISABLE TRAPS ON SBE'S FROM 1 SELECTED CSR
      CALL     MTPC25        ;CHECK FOR EXPECTED TRAP
      RETURN

```

10045 035346 005037 002244  
 10046 035352 005037 002246  
 10047 035356 012737 002244 002306  
 10048 035364 004737 044366  
 10049 035370 012737 000003 002074  
 10050 035376 012701 002406  
 10051 035402 012737 035442 002300  
 10052 035410 004737 035664  
 10053 035414 104473  
 10054 035416 005771 000000  
 10055 035422 005771 000002  
 10056  
 10057 035426 104507  
 10058 035430 005771 000000  
 10059 035434 005771 000002  
 10060 035440 000404  
 10061 035442 104426  
 10062 035444  
 10063 035452 005237 002244  
 10064 035456 004737 035612  
 10065 035462 012737 000400 002244  
 10066 035470 004737 035612  
 10067 035474 005037 002244  
 10068 035500 005237 002246  
 10069 035504 004737 035612  
 10070 035510 012737 000400 002246  
 10071 035516 004737 035612  
 10072  
 10073 035522 005037 002246  
 10074 035526 012737 000003 002244  
 10075 035534 004737 035634  
 10076 035540 012737 001400 002244  
 10077 035546 004737 035634  
 10078 035552 005037 002244  
 10079 035556 012737 000003 002246  
 10080 035564 004737 035634  
 10081 035570 012737 001400 002246  
 10082 035576 004737 035634  
 10083 035602 104503  
 10084 035604 005037 002074  
 10085 035610 000207  
 10086  
 10087 035612 004737 035664  
 10088 035616 104471  
 10089 035620 004737 035706  
 10090 035624 104507  
 10091 035626 004737 035746  
 10092 035632 000207



PC	OP	OP2	OP3	OP4	OP5	OP6	OP7	OP8	OP9	OP10	OP11	OP12	OP13	OP14	OP15	OP16	OP17	OP18	OP19	OP20	OP21	OP22	OP23	OP24	OP25	OP26	OP27	OP28	OP29	OP30	OP31	OP32	OP33	OP34	OP35	OP36	OP37	OP38	OP39	OP40	OP41	OP42	OP43	OP44	OP45	OP46	OP47	OP48	OP49	OP50	OP51	OP52	OP53	OP54	OP55	OP56	OP57	OP58	OP59	OP60	OP61	OP62	OP63	OP64	OP65	OP66	OP67	OP68	OP69	OP70	OP71	OP72	OP73	OP74	OP75	OP76	OP77	OP78	OP79	OP80	OP81	OP82	OP83	OP84	OP85	OP86	OP87	OP88	OP89	OP90	OP91	OP92	OP93	OP94	OP95	OP96	OP97	OP98	OP99	OP100	OP101	OP102	OP103	OP104	OP105	OP106	OP107	OP108	OP109	OP110	OP111	OP112	OP113	OP114	OP115	OP116	OP117	OP118	OP119	OP120	OP121	OP122	OP123	OP124	OP125	OP126	OP127	OP128	OP129	OP130	OP131	OP132	OP133	OP134	OP135	OP136	OP137	OP138	OP139	OP140	OP141	OP142	OP143	OP144	OP145	OP146	OP147	OP148	OP149	OP150	OP151	OP152	OP153	OP154	OP155	OP156	OP157	OP158	OP159	OP160	OP161	OP162	OP163	OP164	OP165	OP166	OP167	OP168	OP169	OP170	OP171	OP172	OP173	OP174	OP175	OP176	OP177	OP178	OP179	OP180	OP181	OP182	OP183	OP184	OP185	OP186	OP187	OP188	OP189	OP190	OP191	OP192	OP193	OP194	OP195	OP196	OP197	OP198	OP199	OP200	OP201	OP202	OP203	OP204	OP205	OP206	OP207	OP208	OP209	OP210	OP211	OP212	OP213	OP214	OP215	OP216	OP217	OP218	OP219	OP220	OP221	OP222	OP223	OP224	OP225	OP226	OP227	OP228	OP229	OP230	OP231	OP232	OP233	OP234	OP235	OP236	OP237	OP238	OP239	OP240	OP241	OP242	OP243	OP244	OP245	OP246	OP247	OP248	OP249	OP250	OP251	OP252	OP253	OP254	OP255	OP256	OP257	OP258	OP259	OP260	OP261	OP262	OP263	OP264	OP265	OP266	OP267	OP268	OP269	OP270	OP271	OP272	OP273	OP274	OP275	OP276	OP277	OP278	OP279	OP280	OP281	OP282	OP283	OP284	OP285	OP286	OP287	OP288	OP289	OP290	OP291	OP292	OP293	OP294	OP295	OP296	OP297	OP298	OP299	OP300	OP301	OP302	OP303	OP304	OP305	OP306	OP307	OP308	OP309	OP310	OP311	OP312	OP313	OP314	OP315	OP316	OP317	OP318	OP319	OP320	OP321	OP322	OP323	OP324	OP325	OP326	OP327	OP328	OP329	OP330	OP331	OP332	OP333	OP334	OP335	OP336	OP337	OP338	OP339	OP340	OP341	OP342	OP343	OP344	OP345	OP346	OP347	OP348	OP349	OP350	OP351	OP352	OP353	OP354	OP355	OP356	OP357	OP358	OP359	OP360	OP361	OP362	OP363	OP364	OP365	OP366	OP367	OP368	OP369	OP370	OP371	OP372	OP373	OP374	OP375	OP376	OP377	OP378	OP379	OP380	OP381	OP382	OP383	OP384	OP385	OP386	OP387	OP388	OP389	OP390	OP391	OP392	OP393	OP394	OP395	OP396	OP397	OP398	OP399	OP400	OP401	OP402	OP403	OP404	OP405	OP406	OP407	OP408	OP409	OP410	OP411	OP412	OP413	OP414	OP415	OP416	OP417	OP418	OP419	OP420	OP421	OP422	OP423	OP424	OP425	OP426	OP427	OP428	OP429	OP430	OP431	OP432	OP433	OP434	OP435	OP436	OP437	OP438	OP439	OP440	OP441	OP442	OP443	OP444	OP445	OP446	OP447	OP448	OP449	OP450	OP451	OP452	OP453	OP454	OP455	OP456	OP457	OP458	OP459	OP460	OP461	OP462	OP463	OP464	OP465	
----	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 307  
MTP025 INTERRUPT ENABLE TEST

10138 036014

```

MTPA26: SUBTST <<MTPA26      RANDOM DATA (WRITE)>>
:*****
:*SUBTEST      MTPA26  RANDOM DATA (WRITE)
:*****
1$:      JMP      @MTPC26      :V177640      GOTO V172360
          MOV      R2,(R1)+    :V177644
          MOV      R3,(R1)+    :V177646
          SOB      R0,1$       :V177650
          RETURN               :V177652

```

10139 036014 000137 036064  
10140 036020 010221  
10141 036022 010321  
10142 036024 077005  
10143 036026 000207  
10144  
10145 036030

```

MTPB26: SUBTST <<MTPB26      RANDOM DATA (READ)>>
:*****
:*SUBTEST      MTPB26  RANDOM DATA (READ)
:*****
          .DSABL  AMA
          .ENABL  LSB
1$:      JMP      @MTPC26      :V177640      GOTO V172360
          CMP      R2,(R1)+    :V177644
          BEQ      2$          :V177646
          PERR25   2$          :V177650
2$:      COM      (PC)+        :V177652
          RANODD: 0            :V177654      FOR ERROR REPORTING
          CMP      R3,(R1)+    :V177656
          BEQ      3$          :V177660
          PERR25   3$          :V177662
3$:      COM      RANODD       :V177664
          SOB      R0,1$       :V177670
          RETURN               :V177672
          .DSABL  LSB
          .ENABL  AMA

```

10146  
10147  
10148 036030 000137 036064  
10149 036034 020221  
10150 036036 001401  
10151 036040 104451  
10152 036042 005127  
10153 036044 000000  
10154 036046 020321  
10155 036050 001401  
10156 036052 104451  
10157 036054 005167 177764  
10158 036060 077015  
10159 036062 000207  
10160  
10161  
10162  
10163 036064

```

MTPC26: SUBTST <<RANDOM NUMBER SUBPROGRAM>>
:*****
:*SUBTEST      RANDOM NUMBER SUBPROGRAM
:*****
          :CALLER MUST SETUP
          :      MOV      SEEDLO,R3
          :      MOV      SEEDHI,R2
          :      MOV      R3,R5
          :      MOV      R2,R4
          ASHC     #7,R4        :V172360
          ADD      R3,R5        :V172364
          ADC      R4           :V172366
          ADD      R2,R4        :V172370
          ADD      #1057,R5     :V172372
          NOP                     :V172376      GOTO V172260

```

10164  
10165  
10166  
10167  
10168  
10169 036064 073427 000007  
10170 036070 060305  
10171 036072 005504  
10172 036074 060204  
10173 036076 062705 001057  
10174 036102 000240  
10175  
10176 036104

```

MTPD26: SUBTST <<RANDOM NUMBER SUBSUBPROGRAM>>
:*****
:*SUBTEST      RANDOM NUMBER SUBSUBPROGRAM
:*****
          ADC      R4           :V172260
          ADD      #47401,R4    :V172262
          MOV      R5,R3        :V172266
          MOV      R4,R2        :V172270
          JMP      @MTPA26+4    :V172272      GOTO V177644

```

10177 036104 005504  
10178 036106 062704 047401  
10179 036112 010503  
10180 036114 010402  
10181 036116 000137 036020

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 309  
RANDOM NUMBER SUBSUBPROGRAM

10184 036122

10185 036122 011002  
10186 036124 010220  
10187 036126 077103  
10188 036130 000207  
10189  
10190 036132

```
MTP030: SUBTST <<MT0030      FLUSH OUT DBE'S>>
:*****
:*SUBTEST      MT0030  FLUSH OUT DBE'S
:*****
1$:  MOV      (R0),R2      :V177640
      MOV      R2,(R0)+    :V177642
      SOB      R1,1$       :V177644
      RETURN                     :V177646
```

```
MTP031: SUBTST <<MTP031      SOB-A-LONG TEST>>
:*****
:*SUBTEST      MTP031  SOB-A-LONG TEST
:*****
```

10191  
10192 036132 000000  
10193 036134 077001  
10194 036136 005167 177772  
10195 036142 020167 177766  
10196 036146 001403  
10197 036150 104454  
10198 036152 010167 177756  
10199 036156 005167 177752  
10200 036162 010200  
10201  
10202 036164 010503  
10203 036166 005725  
10204 036170 010504  
10205 036172 020537 002516  
10206 036176 001001  
10207 036200 000207  
10208  
10209 036202 014344  
10210 036204 001376  
10211 036206 000752  
10212 000056  
10213

```
      .DSABL  AMA
      0
1$:  SOB      R0,1$      :MOVE TERMINATOR
      COM      1$      :SOB TILL R0 UNDERFLOWS
      CMP      R1,1$    :WRITE COMPLEMENT OF SOB
      BEQ      2$      :READ & CHECK FOR NOT 'SOB R0,DOT'
      PERR30
2$:  MOV      R1,1$      :CORRECT SOB INSTRUCTION
      COM      1$      :REINITIALIZE SOB CONSTANT
      MOV      R2,R0
      :UPDATE MOVE REGISTERS
      MOV      R5,R3
      TST      (R5)+    :BUMP (SAFELY) BY 2
      MOV      R5,R4
      CMP      R5,R4
      BNE      3$      :DONE?
      RETURN          :NO - SKIP
                        :YES
3$:  MOV      -(R3),-(R4)
      BNE      3$
      BR       1$
SOBLENGTH=.-MTP031
      .ENABL  AMA
```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 311  
 MTP031 SOB-A-LONG TEST

10241 036210

MTP032: SUBTST <<MTP032 WRITE RECOVERY TEST>>

\*\*\*\*\*  
 :\*SUBTEST MTP032 WRITE RECOVERY TEST  
 :\*\*\*\*\*

:THE TEST ACTUALLY EXECUTED ALREADY IN THE MEMORY UNDER TEST.  
 :THIS CODE INSURES THAT IT CHANGED MEMORY TO HAVE  
 :1/2 BANK OF #5141 WHICH IS A 'COM -(R1)' INSTRUCTION AND  
 :1/2 BANK OF #110 WHICH IS A 'JMP (R0)' INSTRUCTION.

10242  
 10243  
 10244  
 10245  
 10246  
 10247 036210 012401  
 10248 036212 020102  
 10249 036214 001401  
 10250 036216 104430  
 10251 036220 077305  
 10252 036222 013703 002516  
 10253 036226 012400  
 10254 036230 020005  
 10255 036232 001401  
 10256 036234 104427  
 10257 036236 077305  
 10258 036240 000207

1\$:	MOV	(R4)+,R1	:V177640	:GET DATA FROM LOWER 1/2 BANK
	CMP	R1,R2	:V177642	:IS IT #5141?
	BEQ	2\$	:V177644	:YES - SKIP
	PERR02		:V177646	:NO - TAKE ERROR TRAP
2\$:	SQB	R3,1\$	:V177650	:LOOP FOR 1/2 BANK
	MOV	@LINK1,R3	:V177652	:RESTORE LOOP SIZE
3\$:	MOV	(R4)+,R0	:V177656	:GET DATA FROM UPPER 1/2 BANK
	CMP	R0,R5	:V177660	:IS IT #110?
	BEQ	4\$	:V177662	:YES - SKIP
	PERR01		:V177664	:NO- TAKE ERROR TRAP
4\$:	SQB	R3,3\$	:V177666	:LOOP FOR 1/2 BANK
	RETURN			

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 313  
MTP032 WRITE RECOVERY TEST

10261 036242

MTP033: SUBTST &lt;&lt;MTP033 BRANCH GOBBLE TEST&gt;&gt;

\*\*\*\*\*  
:SUBTEST MTP033 BRANCH GOBBLE TEST  
\*\*\*\*\*

10262  
10263 036242 000000  
10264 036244 000000  
10265 036246 000261  
10266 036250 105511  
10267 036252 100402  
10268 036254 105212  
10269 036256 000773

.DSABL AMA  
0  
BGTEST: 0 ;MOVE TERMINATOR  
BRGOBB: SEC ;TEST WORD (TWO BYTES)  
ADCB (R1) ;SET CARRY (TO BE ADDED TO 'BGTEST')  
BMI 1\$ ;INCREMENT LOW BYTE OF 'BGTEST'  
INCB (R2) ;BRANCH WHEN BIT7 IS SET  
BR BRGOBB ;INCREMENT HIGH BYTE OF 'BGTEST'  
;LOOP 128 TIMES

10270  
10271  
10272 036260 102401  
10273 036262 104461  
10274  
10275 036264 000242  
10276 036266 105212  
10277 036270 103402  
10278 036272 102001  
10279 036274 100401  
10280 036276 104461

;NOW CHECK FOR CORRECT CONDITION CODES  
1\$: BVS 2\$ ;BR IF V-BIT SET (SHOULD BE)  
PERR35 ;NO - REPORT ERROR AND ABORT TEST  
;COND CODES NOT EQUAL TO 1010  
2\$: CLV ;CLEAR V-BIT  
INCB (R2) ;INCREMENT HIGH BYTE OF 'BGTEST' ONCE MORE  
BCS 3\$ ;BR IF C-BIT SET (SHOULD NOT BE)  
BVC 3\$ ;BR IF V-BIT CLEAR (SHOULD NOT BE)  
BMI 4\$ ;BR IF N-BIT SET (SHOULD BE)  
3\$: PERR35 ;NO - REPORT ERROR AND ABORT TEST  
;COND CODES NOT EQUAL TO 1010

10281  
10282  
10283  
10284 036300 010701  
10285 036302 162701 000036  
10286 036306 010102  
10287 036310 005202

;UPDATE TEST POINTER  
4\$: MOV PC,R1  
5\$: SUB #5\$-BGTEST,R1  
MOV R1,R2  
INC R2

10288  
10289  
10290 036312 010503  
10291 036314 005725  
10292 036316 010504

;UPDATE MOVE REGISTERS  
MOV R5,R3  
TST (R5)+ ;BUMP (SAFELY) BY 2  
MOV R5,R4

10293  
10294  
10295 036320 020537 002516  
10296 036324 001001  
10297 036326 000207

;DONE?  
CMP R5,#LINK1 ;DONE?  
BNE 6\$ ;NO - SKIP  
RETURN ;YES - RETURN

10298  
10299  
10300 036330 014344  
10301 036332 001376  
10302 036334 005011  
10303 036336 000743  
10304 000076  
10305

;MOVE CODE 1 LOCATION  
6\$: MOV -(R3),-(R4)  
BNE 6\$  
CLR (R1) ;CLEAR TEST WORD 'BGTEST'  
BR BRGOBB ;RUN MOVED CODE AGAIN  
GBLENGTH=-MTP033  
.ENABL AMA

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 314  
 MTP033 BRANCH GOBBLE TEST

10307 036340

```

MTP034: SUBTST <<MTP034      SOFT ERROR - BACKGROUND PATTERN TEST>>
:*****
:*SUBTEST      MTP034  SOFT ERROR - BACKGROUND PATTERN TEST
:*****
1$:      MOV      R2,(R0)+      :V177640
          SOB      R1,MTP034    :V177642
          RETURN                     :V177644
2$:      MOV      (R4)+,R1      :V177646
          CMP      R1,R2        :V177650
          BEQ      3$          :V177652
          PERR02                     :V177654
          NOP                      :V177656
3$:      SOB      R3,2$         :V177660
          RETURN                     :V177662
  
```

10308 036340 010220  
 10309 036342 077102  
 10310 036344 000207  
 10311 036346 012401  
 10312 036350 020102  
 10313 036352 001402  
 10314 036354 104430  
 10315 036356 000240  
 10316 036360 077306  
 10317 036362 000207

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 315  
 MTP034 SOFT ERROR - BACKGROUND PATTERN TEST

10319 036364

MTP035:SUBTST &lt;&lt;MTP035 WORST CASE NOISE PARITY TEST&gt;&gt;

\*\*\*\*\*  
 :\*SUBTEST MTP035 WORST CASE NOISE PARITY TEST  
 :\*\*\*\*\*

10320 036364 012737 000003 002074

MOV #3,NOPAR ;SET PARITY TRAPS TO RETURN TO 'PARTHERE'

10321

10322 036372

FOR R0 := #FIRST TO #LAST BY #4000

10323 036376 012737 000005 002146

MOV #BIT2!BIT0,CSR ;SET WRITE WRONG PARITY &amp; PAR. TRAPS INTO CSR

10324 036404 104425

LOADCSR

10325 036406 012737 036442 002300

MOV #1\$,PARTHERE

10326 036414 011010

MOV (R0),(R0) ;WVP TEST LOCATION

10327 036416 005710

TST (R0)

10328 036420 010037 002032

MOV R0,ADDRESS

10329 036424 104050

ERROR +50

10330 036426 004737 057476

CALL PERBNK

10331 036432 032763 002000 002652

BIT #BIT10,CONFIG+2(R3)

10332 036440 001002

BNE 2\$

10333 036442 104426

1\$: READCSR

10334 036444 104512

ERRGEN

10335

10336 036446 104503

2\$: CLR1CSR

10337 036450 011010

MOV (R0),(R0)

;CLEAR WRONG PARITY IN MEMORY

10338 036452 012737 000001 002146

MOV #BIT0,CSR

10339 036460 104425

LOADCSR

10340 036462 012737 036474 002300

MOV #3\$,PARTHERE

10341 036470 005710

TST (R0)

10342 036472 000405

BR 4\$

10343 036474 010037 002032

3\$: MOV R0,ADDRESS

10344 036500 104050

ERROR +50

10345 036502 004737 057476

CALL PERBNK

10346 036506

4\$: END; OF FOR

10347

10348 036520 005037 002074

CLR NOPAR

;RESET PARITY TRAP ACTION

10349 036524 000207

RETURN

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 316  
MTP035 WORST CASE NOISE PARITY TEST

10351 036526

```

MTP036: SUBTST      <<MTP036      CORRECTION CODE TEST>>
:*****
:*SUBTEST      MTP036  CORRECTION CODE TEST
:*****
:
:  THIS TEST CHECKS TO SEE THAT EACH BIT OF A DATA WORD
:  CAN BE CORRECTED INDIVIDUALLY FROM A ZERO TO A ONE AND
:  VISA VERSA.
:
:  CACHOFF
:  CLR PASFLG
:  CBREG
:  REPEAT
:    LET PASFLG :B= PASFLG + #1
:    LET R4 := #1
:    LET BITNO := #0
:    IFB PASFLG EQ #1
:      LET R5 := #1
:    ELSE
:      LET R5 := #177776
:    END
:  REPEAT
:
:    INC BITNO
:    LET R4 := R4 + #1
:    LET R2 :B= PTABLE(R4)
:    ASH #5,R2
:    BIS #BIT2,R2
:    LET CSR := R2
:    LOADCSR
:    LET (R1) := R0
:    TST (R1)
:    IF (R1) NE R5
:      LET ADDRESS := #60000
:      LET CHECK := R2
:      LET TSTDAT := R5
:      LET TSTDAT+2 := (R1)
:      ERROR +52
:    END
:    CLR (R1)
:    IFB PASFLG EQ #1
:      ASL R5
:    ELSE
:      SEC
:      ROL R5
:    END
:  UNTIL BITNO EQ #16.
:  COM R0
:  UNTILB PASFLG EQ #2
:  CLR CSR
:  CACHON
:  RETURN
:
:  :TURN OFF CACHE
:  :CLEAR PASFLG
:  :ENABLE CHECK/SYNDROME BIT REGISTER
:
:  :INCREMENT LOOP COUNTER
:  :INDEX TO SINGLE BIT ERROR TABLE
:  :CLEAR INNER LOOP COUNTER
:  :SELECT DATA TO BE CORRECTED BY PASSNO
:  :DATA=0;BIT TO BE CORRECTED IS A ONE
:
:  :DATA=177776;BIT TO BE CORRECTED IS A ZERO
:
:
:  :INCREMENT BIT POINTER
:  :POINT TO NEXT SET OF CHECK BITS
:  :GET NEXT SET OF CHECK BITS
:  :SHIFT TO LINE UP IN CSR
:  :ENABLE DIAG MODE
:  :GET CHECK BITS TO BE WRITTEN
:  :LOAD CSR WITH DATA
:  :WRITE DATA TO TEST ADDRESS
:  :CORRECT SBE
:  :WAS DATA CORRECTED???
:  :MOV ERROR INFORMATION IN
:
:
:  :NO ERROR
:
:  :CLEAR LUT
:  :SHIFT NEW DATA DEPENDING ON PASFLG
:  :SHIFT BITNO TO THE LEFT
:
:  :SET CARRY BIT AND.....
:  :ROTATE LEFT
:
:  :UNTIL ALL BITS ARE DONE
:  :COMPLEMENT DATA AND REPEAT
:  :UNTIL 2 PASSES ARE COMPLETE!
:  :CLEAR CSR
:  :TURN CACHE
:
:

```

10352  
10353  
10354  
10355  
10356 036526 104424  
10357 036530 105037 002262  
10358 036534 104513  
10359 036536  
10360 036536  
10361 036542  
10362 036546  
10363 036552  
10364 036562  
10365 036566  
10366 036570  
10367 036574  
10368 036574  
10369  
10370 036574 005237 002320  
10371 036600  
10372 036602  
10373 036606 072227 000005  
10374 036612 052702 000004  
10375 036616  
10376 036622 104425  
10377 036624  
10378 036626 005711  
10379 036630  
10380 036634  
10381 036642  
10382 036646  
10383 036652  
10384 036656 104052  
10385 036660  
10386 036660 005011  
10387 036662  
10388 036672 006305  
10389 036674  
10390 036676 000261  
10391 036700 006105  
10392 036702  
10393 036702  
10394 036712 005100  
10395 036714  
10396 036724 104503  
10397 036726 104423  
10398 036730 000207  
10399

10400  
10401  
10402 036732 002 007 037  
036735 031 032 025  
036740 026 020 057

MS11-P SINGLE BIT ERROR CHECK BIT TABLE  
PTABLE: .BYTE 2,7,37,31,32,25,26,20,57,51,52,45,46,40,75,70



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 316-1  
MTP036 CORRECTION CODE TEST

036743	051	052	045
036746	046	040	075
036751	070		

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 317  
 MTP036 CORRECTION CODE TEST

10404 036752

```

MTP037: SUBTST <<MTP037      CHECK ECC DISABLE TEST>>
:*****
:*SUBTEST      MTP037 CHECK ECC DISABLE TEST
:*****
:
: THIS TEST CHECKS THAT ECC CAN BE DISABLED AND THAT
: NO CORRECTION TAKES PLACE WITH ECC DISABLED.
:
: CACHOFF          :TURN OFF CACHE
LET GOOD := #0      :GOOD DATA FOR ERROR PRINT OUT
LET CHECK := #0      :CLEAR CHECK BIT FIELD
CB1CSR            :ENABLE SYNDROME/CHECK BIT REGISTER
LET CHECK := #100    :SBE CHECK BITS
CB1CSR            :WRITE CHECK BITS TO CB REGISTER
LET (R1) := #0       :WRITE CHECK BITS TO MEMORY
IF (R1) NE #0        :WAS CORRECTION MADE???
    LET BAD := (R1)  :YES IT WAS.....ERROR
    LET ADDRESS := #60000
    ERROR +37
END
CACHON            :TURN ON CACHE
RETURN

```

10405  
 10406  
 10407  
 10408  
 10409 036752 104424  
 10410 036754  
 10411 036760  
 10412 036764 104475  
 10413 036766  
 10414 036774 104475  
 10415 036776  
 10416 037000  
 10417 037004  
 10418 037010  
 10419 037016 104037  
 10420 037020  
 10421 037020 104423  
 10422 037022 000207

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 319  
MTP037 CHECK ECC DISABLE TEST

10425 037024

10426  
10427  
10428  
10429  
10430 037024  
10431 037030 072427 000011  
10432 037034 042704 170037  
10433 037040  
10434 037044  
10435 037054  
10436 037060 005037 002310  
10437 037064 104475  
10438 037066  
10439 037066 105237 002262  
10440 037072  
10441 037076  
10442 037102  
10443 037110 104475  
10444 037112  
10445 037114 104503  
10446 037116 005711  
10447 037120 104426  
10448 037122  
10449 037126 042705 170037  
10450 037132  
10451 037134 060402  
10452 037136 000240  
10453 037140  
10454 037144  
10455 037150  
10456 037154 104455  
10457 037156  
10458 037156  
10459 037160 104475  
10460 037162  
10461 037172 104503  
10462 037174 000207  
10463

```

MTP041: SUBTST <<MTP041 ADDRESS TO CSR ON DOUBLE BIT ERROR TEST>>
*****
*SUBTEST MTP041 ADDRESS TO CSR ON DOUBLE BIT ERROR TEST
*****
:
: THIS TEST CHECKS TO SEE IF THE CORRECT ADDRESS APPEARS
: IN CSR BITS 5-11 ON A DOUBLE ERROR.
:
LET R4 := BANK ;GET STARTING BANK NUMBER
ASH #9, R4 ;SHIFT INTO POSTION TO MATCH ADDRESS IN CSR
BIC #^C7740, R4 ;CLEAR OFF EXTRANEIOUS BITS
LET R0 := #-40 ;INIT CSR ADDRESS TO 0 - 1K (BIT 5 = 1K ADD.)
LET R1 := #FIRST - #4000 ;GET LOW ADDRESS IN BANK
LET PASFLG := B= #0 ;INIT PASFLG
CLR CHECK ;CLEAR CHECK BIT FIELD TO BE LOADED
CB1CSR ;ENABLE CHECK/SYNDROME BIT REGISTER
REPEAT
INCB PASFLG ;INC LOOP COUNTER
LET R0 := R0 + #40 ;INC CSR ADDRESS TO BE EXPECTED
LET R1 := R1 + #4000
LET CHECK := #1340 ;DOUBLE ERROR CHECK BITS
CB1CSR ;WRITE DOUBLE ERROR CHECK BITS
LET (R1) := #0 ;WRITE DATA AND D.E. CHK BITS AT A=0
CLR1CSR ;CLEAR CSR
TST (R1) ;READ ADDRESS TO GET DOUBLE ERROR
READCSR ;READ CSR FOR CORRECT ADDRESS
LET R5 := CSR
BIC #^C7740, R5
LET R2 := R0
ADD R4, R2
NOP
IF R2 NE R5
LET BAD := R2
LET GOOD := R5
PERR31
END
LET (R1) := #0
CB1CSR
UNTILB PASFLG EQ #16.
CLR1CSR
RETURN
:
: GET CORRECT ADDRESS
: ADD STARTING BANK TO DOUBLE BIT ADDRESS
: DEBUG AIDE
: DO ADDRESSES AGREE?
:
: NO ERROR
:
: ENABLE CHECK/SYNDROME BIT REGISTER
: DO 16K AT A TIME
:

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 321  
MTP041 ADDRESS TO CSR ON DOUBLE BIT ERROR TEST

10466 037176

MTP042: SUBTST <<MTP042 EXTENDED ADDRESS TO CSR ON ERROR TEST>>  
:\*\*\*\*\*  
:SUBTEST MTP042 EXTENDED ADDRESS TO CSR ON ERROR TEST  
:\*\*\*\*\*

: THIS TESTS THE EXTENDED UNIBUS ADDRESS IN THE  
: CSR BY CAUSING A SINGLE ERROR, ENABLING BIT # 14, THEN CHECKING  
: FOR THE PROPER ADDRESS IN THE CSR.

10467  
10468  
10469  
10470  
10471  
10472 037176 104424  
10473 037200  
10474 037204 042704 177607  
10475 037210 072427 000002  
10476 037214 052704 040000  
10477 037220 062737 000400 172352  
10478 037226  
10479 037232  
10480 037236 042705 177770  
10481 037242 072527 000011  
10482 037246 052705 000020  
10483 037252 104513  
10484 037254  
10485 037254 105237 002262  
10486 037260  
10487 037266 104425  
10488 037270  
10489 037272 104503  
10490 037274 005711  
10491 037276 104426  
10492 037300 042737 020000 002146  
10493 037306  
10494 037314  
10495 037322  
10496 037326 104023  
10497 037330  
10498 037330  
10499 037336 104425  
10500 037340 104426  
10501 037342 042737 020000 002146  
10502 037350  
10503 037356  
10504 037364  
10505 037370  
10506 037376 104023  
10507 037400  
10508 037400  
10509 037402  
10510 037406 062705 000740  
10511 037412 104513  
10512 037414  
10513 037424 104503  
10514 037426 104423  
10515 037430 000207  
10516  
10517

```

CACHOFF                                ;TURN OFF CACHE MEMORY
LET R4 := BANK                        ;GET BANK NUMBER TO FIGURE OUT EXTENDED ADDRESS
BIC #^C170,R4                        ;CLEAR OFF LOWER BITS
ASH #2,R4                             ;SHIFT TO LINE UP WITH CSR
BIS #BIT14,R4                         ;SET EXTENDED ADDRESS BIT
ADD #400,KIPAR5                       ;SET UP PAR TO POINT TO TOP OF A BANK
LET PASFLG := #0                     ;INIT LOOP COUNTER
LET R5 := BANK                        ;R5 GETS THE BANK NUMBER
BIC #^C7,R5                           ;CLEAR ALL BUT THE LOWER BITS
ASH #9,R5                             ;ROTATE INTO POSITION
BIS #BIT4,R5                          ;SET UP SBE INDICATOR ;:DATA TO BE EXPECTED
CBREG                                ;ENABLE CHECK/SYNDPOME BIT REGISTER
REPEAT
  INCB PASFLG                         ;INCR LOOP COUNTER
  LET CSR := #104                    ;WRITE CHECK BITS TO CSR WITH DIAG MODE
  LOADCSR                             ;LOAD CSR WITH DATA
  LET (R1) := #0                     ;WRT ZEROS AT A=0 AND SINGLE ERROR BITS
  CLR1CSR                             ;CLEAR CSR
  TST (R1)                            ;READ A=0;DATA BIT 0 SHOULD BE CORRECTED TO A 1
  READCSR                             ;READ CSR FOR DATA
  BIC #BIT13,CSR                      ;CLEAR POSSIBLE INHIBIT MODE IN DATA 'CSR'
  IF CSR NE R5 THEN                   ;HAS SINGLE ERROR BITS SET IN CSR?
    LET BAD := CSR
    LET GOOD := R5
    ERROR +23
  END
  LET CSR := #40000                  ;WRITE EUB BIT TO CSR
  LOADCSR                             ;READ FOR CORRECT EXTENDED UNIBUS ADDRESS
  READCSR                             ;CLEAR INHIBIT MODE POINTER IN DATA
  BIC #BIT13,CSR                      ;READ EUB ADDRESS
  IF CSR NE R4 THEN
    LET BAD := CSR
    LET GOOD := R4
    SET HEADER
    ERROR +23
  END
  LET (R1) := #0                     ;CLEAR LUT
  LET R1 := #137776                 ;SET UP NEW ADDRESS
  ADD #740,R5                       ;ADD TO GET NEW ADDRESS
  CBREG                              ;ENABLE CHECK/SYNDROME BIT REGISTER
  UNTILB PASFLG EQ #2               ;LOOP 2 TIMES
  CLR1CSR                           ;CLEAR CSR
  CACHON                             ;TURN ON CACHE
  RETURN

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 322  
MTP042 EXTENDED ADDRESS TO CSR ON ERROR TEST

10519 037432

MTP043: SUBTST &lt;&lt;MTP043 WRITE BYTE CLEARS SINGLE BIT ERROR TEST&gt;&gt;

\*\*\*\*\*  
:SUBTEST MTP043 WRITE BYTE CLEARS SINGLE BIT ERROR TEST  
\*\*\*\*\*

10520  
10521  
10522  
10523  
10524  
10525 037432 104424  
10526 037434 104513  
10527 037436 105037 002262  
10528 037442  
10529 037446  
10530 037452  
10531 037452 105237 002262  
10532 037456  
10533 037464 104425  
10534 037466  
10535 037470 104503  
10536 037472  
10537 037476 104426  
10538 037500 042737 177757 002146  
10539 037506  
10540 037516  
10541 037524  
10542 037532 104060  
10543 037534  
10544 037534 104513  
10545 037536 005711  
10546 037540 104426  
10547 037542 042737 174037 002146  
10548 037550  
10549 037560  
10550 037566  
10551 037574  
10552 037602 104061  
10553 037604  
10554 037604 005302  
10555 037606  
10556 037612  
10557 037622 104423  
10558 037624 000207

: THIS TEST CHECKS TO SEE IF A SINGLE BIT ERROR WILL BE CORRECTED DURING  
: THE READ PORTION OF A WRITE BYTE AND THAT THE CORRECT CHECK BITS WILL  
: BE GENERATED ON A WRITE.  
:  
: CACHOFF :TURN OFF CACHE  
: CBREG :ENABLE CHECK/SYNDROME BIT REGISTER  
: CLR PASFLG :CLEAR LOOP COUNTER  
: LET R2 := R1 + #1 :R2 POINTS TO HIGH BYTE  
: LET R4 := #1 :INITIAL DATA = 1  
: REPEAT  
: INCB PASFLG :INCREMENT LOOP COUNTER  
: LET CSR := #604 :WRITE CHECK BITS CORRESPONDING TO DATA OF 0  
: LOADCSR :WRITE CSR  
: LET (R1) := R4 :WRITE DATA OF 1 CREATING A SINGLE BIT ERROR  
: CLR1CSR :WRITE CSR TO NORMAL MODE  
: LET (R2) := #377 :WRITE BYTE OF WORD  
: READCSR :READ CSR  
: BIC #C20,CSR :SEE IF SBE INDICATOR IS SET  
: IF CSR NE #20 :IS SBE SET? ???  
: LET GOOD := #20  
: LET BAD := CSR  
: ERROR +60  
: END  
: CBREG :WRITE CSR TO DIAG MODE  
: TST (R1) :READ SA0 FOR CORRECT CHECK BITS  
: READCSR :READ CSR  
: BIC #C3740,CSR :MASK OUT CHECK BIT FIELD  
: IF CSR NE #300 :WERE CORRECT CHECK BITS GENERATED????  
: SET HEADER  
: LET GOOD := #300  
: LET BAD := CSR  
: ERROR +61  
: END  
: DEC R2 :POINT TO HIGH BYTE AND REPEAT  
: LET R4 := #400 :BIT 0 OF HIGH BYTE  
: UNTILB PASFLG EQ #2 :DO HIGH AND LOW BYTE  
: CACHON :TURN ON CACHE  
: RETURN  
:

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 323  
MTP043 WRITE BYTE CLEARS SINGLE BIT ERROR TEST

10560 037626

MTP044: SUBTST <<MTP044 SHIFTING CHECK BITS THROUGH THE CSR TEST>>

\*\*\*\*\*  
:SUBTEST MTP044 SHIFTING CHECK BITS THROUGH THE CSR TEST  
\*\*\*\*\*

: THIS TEST CHECKS THE ABILITY TO READ AND WRITE CHECKBITS INTO MEMORY  
: BY SHIFTING A ONE BIT THROUGH A FIELD OF ZEROS. THE CSR IS READ FOR THE  
: CORRECT PATTERNS. THE TEST IS THEN REPEATED ON A ZERO BIT THROUGH A  
: FIELD OF ALL ONES.

10561  
10562  
10563  
10564  
10565  
10566  
10567 037626 104424  
10568 037630  
10569 037634  
10570 037640 104475  
10571 037642  
10572 037646  
10573 037646  
10574  
10575  
10576 037652  
10577 037656  
10578 037656  
10579 037662  
10580 037664  
10581 037670 104425  
10582 037672  
10583 037674 005105  
10584 037676 010546  
10585 037700 040416  
10586 037702 040504  
10587 037704 052604  
10588 037706  
10589 037712 104425  
10590 037714 104426  
10591 037716  
10592 037722 042703 020000  
10593 037726  
10594 037732  
10595 037740  
10596 037744  
10597 037750  
10598 037756 104053  
10599 037760  
10600 037760 005105  
10601 037762 005711  
10602 037764 000240  
10603 037766 104426  
10604 037770 040537 002146  
10605 037774  
10606 037776 040504  
10607 040000  
10608 040006  
10609 040012  
10610 040020  
10611 040026  
10612 040034 104054  
10613 040036

```

CACHOFF
LET PASFLG :B= #0
LET R5 := #174037
CB1CSR
LET R2 := #46
REPEAT
    LET PASFLG :B= PASFLG + #1
    LET PASSNO := #0
    REPEAT
        LET PASSNO := PASSNO + #1
        LET R4 := R2
        LET CSR := R2
        LOADCSR
        LET (R1) := #0
        COM R5
        MOV R5, -(SP)
        BIC R4, (SP)
        BIC R5, R4
        BIS (SP), R4
        LET CSR := R4
        LOADCSR
        READCSR
        LET R3 := CSR
        BIC #BIT13, R3
        IF R3 NE R4 THEN
            LET ADDRESS := #FIRST
            LET GOOD := R4
            LET BAD := R3
            SET HEADER
            ERROR +53
        END
        COM R5
        TST (R1)
        NOP
        READCSR
        BIC R5, CSR
        LET R4 := R2
        BIC R5, R4
        IF R4 NE CSR
            LET GOOD := R4
            LET BAD := CSR
            LET ADDRESS := #FIRST
            SET HEADER
            ERROR +54
        END
    END
    :TURN OFF CACHE
    :INIT PASFLG
    :CHECK BIT MASK FOR CSR
    :ENABLE CHECK/SYNDROME BIT REGISTER
    :SET UP INITIAL CSR DATA
    :INC LOOP COUNTER
    :CHK BITS = 1
    :DISABLE ECC/DIAG CHK SET
    :INIT PASSNO(INNER LOOP COUNTER)
    :INC LOOP COUNTER
    :COPY R2 TO R4
    :GET CSR DATA TO BE WRITTEN
    :WRITE SBE CHECK BITS TO CSR
    :WRITE DATA AND CHECK BITS AT A=0
    :COMPLEMENT MASK
    :SAVE R5 ON STACK
    :CREATE AN XOR FUNCTION
    :
    :LOAD CSR WITH COMPLEMENT CHECK BITS
    :READ CSR FOR COMPLEMENT CHECK BITS
    :COPY CSR DATA TO R3
    :CLEAR ANY POSSIBLE INHIBIT MODE POINTER
    :READ CSR FOR PROPER CHECK BITS
    :
    :ERROR CALL
    :COMPLEMENT MASK
    :READ CHECK BITS AT A=0 INTO CSR
    :
    :READ CSR FOR CORRECT CHECK BITS
    :MASK OUT CHECK BIT FIELD
    :GET CHECK BITS THAT WERE WRITTEN
    :MASK OUT CHECK BIT FIELD
    :ARE CHECK BITS THE SAME?
    :
    :ERROR CALL

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 323-1  
 MTP044 SHIFTING CHECK BITS THROUGH THE CSR TEST

10614 040036 040502  
 10615 040040  
 10616  
 10617 040050 006302  
 10618 040052  
 10619 040054 005105  
 10620 040056 010546  
 10621 040060 040216  
 10622 040062 040502  
 10623 040064 052602  
 10624 040066 006302  
 10625 040070 010546  
 10626 040072 040216  
 10627 040074 040502  
 10628 040076 052602  
 10629 040100 005105  
 10630 040102  
 10631 040102  
 10632 040106  
 10633 040116  
 10634 040122  
 10635 040132 104503  
 10636 040134 005011  
 10637 040136 104423  
 10638 040140 000207

BIC R5,R2  
 IFB PASFLG EQ #1  
  
 ASL R2  
 ELSE  
 COM R5  
 MOV R5,-(SP)  
 BIC R2,(SP)  
 BIC R5,R2  
 BIS (SP)+,R2  
 ASL R2  
 MOV R5,-(SP)  
 BIC R2,(SP)  
 BIC R5,R2  
 BIS (SP)+,R2  
 COM R5  
 END  
 LET R2 := R2 + #6  
 UNTILB PASSNO EQ #6  
 LET R2 := #3706  
 UNTILB PASFLG EQ #2  
 CLR1CSR  
 CLR (R1)  
 CACHON  
 RETURN

;SHIFT CHECK BITS AND CREATE NEW DATA FOR CSR  
 ;SELECT FUNCTION  
 ;DO A FIELD OF ZEROS--->ONES  
 ;SHIFT CHECK BITS  
 ;DO A FIELD OF ONES --->ZEROS  
  
 ;TAKE OUT CHECK BIT FIELD  
  
  
 ;SHIFT CHECK BITS  
 ;PUT BACK CHECK BIT FIELD  
  
  
 ;COMPLEMENT DATA PATTERN  
  
 ;ADD 6 SO THAT WRITE ON CSR WILL ENABLE DIAG MODE  
 ;DO ALL CHECK BITS  
 ;REPEAT WITH FIELD OF ONES  
  
  
 ;TURN ON CACHE  
 ;

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 324  
MTP044 SHIFTING CHECK BITS THROUGH THE CSR TEST

10640 040142

MTP045: SUBTST &lt;&lt;MTP045 SYNDROMES TO CSR ON DOUBLE BIT ERROR TEST&gt;&gt;

\*\*\*\*\*  
:SUBTEST MTP045 SYNDROMES TO CSR ON DOUBLE BIT ERROR TEST  
\*\*\*\*\*

10641  
10642  
10643  
10644  
10645  
10646 040142 104424  
10647 040144 104513  
10648 040146  
10649 040152  
10650 040160  
10651 040166  
10652 040166 005237 002264  
10653 040172 104425  
10654 040174  
10655 040176 104503  
10656 040200 005711  
10657 040202 104426  
10658 040204  
10659 040214  
10660 040222  
10661 040230 104063  
10662 040232  
10663 040232 104513  
10664 040234 104426  
10665 040236 000240  
10666 040240 042737 174033 002146  
10667 040246  
10668 040256  
10669 040264  
10670 040272 104042  
10671 040274  
10672 040274 005011  
10673 040276  
10674 040304  
10675 040312  
10676 040322 104503  
10677 040324 104423  
10678 040326 000207

```

:
: THIS TEST CHECKS TO SEE IF THE DOUBLE BIT ERROR INDICATOR IS SET
: ON A DOUBLE BIT ERROR AND THE CORRECT SYNDROMES ARE LATCHED INTO THE
: CSR. THIS TEST IS THEN REPEATED WITH MULTIPLE ERROR CHECK/SYNDROME BITS
:
: CACHOFF ;TURN OFF CACHE
: CBREG ;ENABLE CHECK/SYNDROME BIT REGISTER
: LET PASSNO := #0 ;CLEAR LOOP COUNTER
: LET GOOD := #3744 ;GOOD DATA
: LET CSR := #3144 ;DBE CHECK BITS FOR CSR
: REPEAT
:   INC PASSNO
:   LOADCSR ;WRITE DBE CHECK BITS TO CSR
:   LET (R1) := #0 ;WRITE ZEROS AND DBL ERROR CHK BITS A=0
:   CLR1CSR ;CLEAR CSR OUT
:   TST (R1) ;READ A=0 TO GET DOUBLE BIT ERROR
:   READCSR ;WAS UNCORRECCABLE ERROR BIT SET???
:   IF #BIT15 OFF.IN CSR
:     SET HEADER
:     LET BAD := CSR
:     ERROR +63 ;BIT NOT SET
:   END
:   CBREG ;ENABLE SYNDROME BIT REGISTER
:   READCSR ;READ CSR FOR CORRECT SYNDROME BITS
:   NOP ;DEBUG AIDE
:   BIC #^C3744,CSR ;MASK SYNDROMES OUT
:   IF CSR NE GOOD THEN ;CHECK IF DOUBLE ERROR BIT IS SET
:     LET BAD := CSR
:     SET HEADER
:     ERROR +42 ;BAD DATA
:   END
:   CLR (R1) ;CLEAR LUT
:   LET GOOD := #3604 ;REPEAT WITH MULTIPLE ERROR SYNDROMES
:   LET CSR := #3004 ;MULTIPLE ERROR CHECK BITS
: UNTIL PASSNO EQ #2
: CLR1CSR
: CACHON
: RETURN

```



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 325  
 MTP045 SYNDROMES TO CSR ON DOUBLE BIT ERROR TEST

10680 040330

MTP046: SUBTST &lt;&lt;MTP046 CHECK SINGLE BIT ERRORS WITH ECC DISABLED&gt;&gt;

\*\*\*\*\*  
 \*SUBTEST MTP046 CHECK SINGLE BIT ERRORS WITH ECC DISABLED  
 \*\*\*\*\*

10681  
 10682  
 10683  
 10684  
 10685  
 10686 040330 005037 002264  
 10687 040334 104424  
 10688 040336  
 10689 040336  
 10690 040342 005000  
 10691 040344 105037 002262  
 10692 040350 104513  
 10693 040352  
 10694 040352  
 10695 040356  
 10696 040362  
 10697 040370  
 10698 040374  
 10699 040404  
 10700 040410  
 10701 040412  
 10702 040416  
 10703 040416  
 10704 040416  
 10705 040422 005237 002320  
 10706 040426  
 10707 040430  
 10708 040434 072227 000005  
 10709 040440 052702 000006  
 10710 040444  
 10711 040450 104425  
 10712 040452  
 10713 040454  
 10714 040464 104471  
 10715 040466  
 10716 040470 104507  
 10717 040472  
 10718 040472 005711  
 10719 040474 004737 040614  
 10720 040500 104426  
 10721 040502  
 10722 040512  
 10723 040520  
 10724 040526 104045  
 10725 040530  
 10726 040530 104503  
 10727 040532 005011  
 10728 040534  
 10729 040544 006305  
 10730 040546  
 10731 040550 000261  
 10732 040552 006105  
 10733 040554

```

: THIS TEST CHECKS TO SEE THAT FOR EACH BIT OF A DATA WORD THAT A SBE
: IS TREATED LIKE A UNCORRECTABLE ERROR WITH ECC DISABLED AND TRAPS
: ARE DETECTED.
:
CLR PASSNO                                :CLEAR OUTER LOOP COUNTER
CACHOFF                                  :TURN OFF CACHE
REPEAT
  LET PASSNO := PASSNO + #1
  CLR R0                                  :CLEAR DATA
  CLRB PASFLG                             :CLEAR PASFLG
  CBREG                                   :ENABLE CHECK/SYNDROME BIT REGISTER
  REPEAT
    LET PASFLG := PASFLG + #1             :INCREMENT LOOP COUNTER
    LET R4 := #1                          :INDEX TO SINGLE BIT ERROR TABLE
    LET NOPAR := #1                       :ENABLE PARITY ACTION
    LET BITNO := #0                       :CLEAR INNER LOOP COUNTER
    IFB PASFLG EQ #1                      :SELECT DATA TO BE CORRECTED BY PASSNO
      LET R5 := #1                        :DATA=0;BIT TO BE CORRECTED IS A ONE
    ELSE
      LET R5 := #177776                  :DATA=177776;BIT TO BE CORRECTED IS A ZERO
    END
    REPEAT
      LET PARCNT := #0                   :CLEAR PARITY COUNTER
      INC BITNO                           :INCREMENT BIT POINTER
      LET R4 := R4 + #1                   :POINT TO NEXT SET OF CHECK BITS
      LET R2 := PTABLE(R4)                :GET NEXT SET OF CHECK BITS
      ASH #5,R2                           :SHIFT TO LINE UP IN CSR
      BIS #BIT2:BIT1,R2                   :ENABLE DIAG MODE
      LET CSR := R2                       :GET CHECK BITS TO BE WRITTEN
      LOADCSR                             :LOAD CSR WITH DATA
      LET (R1) := R0                      :WRITE DATA TO TEST ADDRESS
      IF PASSNO EQ #1                     :WRITE CSR
        ECC1DIS                           :FIRST PASS ;ECC DISABLE,NO PBL
      ELSE
        ENA1SBE                           :SECOND PASS ;ECC DISABLE,PBL ENABLED
      END
      TST (R1)                             :CORRECT SBE
      CALL CHKTRP                          :CHECK FOR CORRECT TRAP
      READCSR                             :READ THE CSR FOR UNCORRECTABLE ERROR
      IF #BIT15 OFF.IN CSR                 :IS UNCORRECTABLE ERROR BIT SET???
        LET BAD := CSR
        SET HEADER
        ERROR +45
      END
      CLR1CSR
      CLR (R1)
      IFB PASFLG EQ #1
        ASL R5
      ELSE
        SEC
        ROL R5
      END
    END
  END
  :CLEAR LUT
  :SHIFT NEW DATA DEPENDING ON PASFLG
  :SHIFT BITNO TO THE LEFT
  :
  :SET CARRY BIT AND.....
  :ROTATE LEFT
  :

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 325-1  
 MTP046 CHECK SINGLE BIT ERRORS WITH ECC DISABLED

10734	040554		UNTIL BITNO EQ #16.	:UNTIL ALL BITS ARE DONE
10735	040564	005100	COM R0	:COMPLEMENT DATA AND REPEAT
10736	040566		UNTILB PASFLG EQ #2	:UNTIL 2 PASSES ARE COMPLETE!
10737	040576		UNTIL PASSNO EQ #2	:
10738	040606	104503	CLR1CSR	:
10739	040610	104423	CACHON	:TURN CACHE
10740	040612	000207	RETURN	:
10741				
10742				
10743	040614		CHKTRP: IF PASSNO EQ #1	:PASS 1 CHECK FOR NO TRAP
10744	040624		IF PARCNT EQ #1	:
10745	040634		SET HEADER	:
10746	040642	104057	ERROR +57	:
10747	040644		END	:
10748	040644		ELSE	:
10749	040646		IF PARCNT NE #1	:
10750	040656		SET HEADER	:
10751	040664	104062	ERROR +62	:
10752	040666		END	:
10753	040666		END	:
10754	040666	000207	RETURN	:

10756 040670

```

MTP047: SUBTST  <<MTP047          NO CSR UPDATE ON SBE WITH EXSISTING DBE>>
:*****
:*SUBTEST      MTP047  NO CSR UPDATE ON SBE WITH EXSISTING DBE
:*****

```

```

: THIS TEST CHECKS TO SEE THAT THE CSR CONTENTS WILL NOT CHANGE
: WITH A SINGLE BIT ERROR WHEN A DOUBLE BIT ERROR ALREADY
: EXISTS.

```

```

CACHOFF
LET R4 := BANK
ASH #9,R4
BIC #0x7740,R4
BIS #BIT15,R4
CBREG
LET CSR := #3144

```

```

      LET CSR := #3144

```

LOADCSR

```
LET (R1) := #0
```

```
LET CSR := #104
LOADCSR
```

```
LOADCSR
LET (R2) := #0
```

```
LET (R2) := NO
CLR1CSR
```

TST (R1)

READCSR

BIC #BIT13.CSR  
IS 668-NE 8/

```
IF CSR NE R4
    LET RAD := CS
```

```
LET BAD := CS
LET GOOD := R
```

```

SET HEADER

```

SET HEADER  
ERROR +63

END

BIS #20, R4

TST (R2)  
READCSB

READ CSR  
BIC #B1113 CSR

BIC WB1113, CSR  
IF CSR NE R4

```

17  CSK NE R4
    LET BAD := CS

```

LET GOOD := R

SET HEADER  
ERROR: 51

ERROR +51  
END

END  
CLR1CSB

CLR (R1)

CLR (R2)

**CACHON**

## RETURN

```

:TURN OFF CACHE
:GET BANK NUMBER
:SHIFT INTO PLACE
:MASK OUT UNWANTED BITS
:SET UP GOOD DATA
:ENABLE CHECK/SYNDROME BIT REGISTER
:CHECK BITS FOR DOUBLE BIT ERROR
:
:WRITE DBE CHECK BITS
:WRITE SBE CHECK BITS
:
:WRITE SBE CHECK BITS AT ADDRESS + 4K
:CLEAR CSR
:READ DBE LOCATION
:READ FOR CSR DBE INDICATOR
:CLEAR INHIBIT MODE POINTER

```

```

:SET BIT IN GOOD DATA
:READ SBE
:READ CSR FOR NO CHANGE
:CLEAR INHIBIT MODE POINTER

```

```

: CLEAR 1 CSR
:
: TURN ON CACHE

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 327  
MISC SUBROUTINES

```

10800                      .SBTTL  MISC SUBROUTINES
10801
10802 041064              REGCOPY:SUBTST <<SUBR  COPY R0 TO R4,R1 TO R3, & R2 TO R5>>
                          :*****
                          :*SUBTEST      SUBR      COPY R0 TO R4,R1 TO R3, & R2 TO R5
                          :*****
10803 041064 010004      MOV      R0,R4
10804 041066 010103      MOV      R1,R3
10805 041070 010205      MOV      R2,R5
10806 041072 000207      RETURN
10807
10808 041074              FLIPWARN:SUBTST <<FLIP WARNING CONSTANTS IN WORST CASE NOISE TESTS>>
                          :*****
                          :*SUBTEST      FLIP WARNING CONSTANTS IN WORST CASE NOISE TESTS
                          :*****
10809 041074              PUSH     R0
10810 041076 005237 002602      INC      FLIPLOC
10811 041102 042737 177774 002602      BIC      #^C3,FLIPLOC
10812 041110 022737 000001 002602      CMP      #1,FLIPLOC
10813 041116 001414              BEQ      1$
10814 041120 022737 000002 002602      CMP      #2,FLIPLOC
10815 041126 001413              BEQ      2$
10816 041130 022737 000003 002602      CMP      #3,FLIPLOC
10817 041136 001414              BEQ      3$
10818 041140 005000              CLR      R0
10819 041142 013704 002600      MOV      ONES,R4
10820 041146 000414              BR      4$
10821 041150              1$:      CLEAR  R0,R4
10822 041154 000411              BR      4$
10823 041156 012700 000401      2$:      MOV      #401,R0
10824 041162 013704 002600      MOV      ONES,R4
10825 041166 000404              BR      4$
10826 041170 012700 000401      3$:      MOV      #401,R0
10827 041174 012704 000401      MOV      #401,R4
10828 041200 010037 027522      4$:      MOV      R0,WARN2
10829 041204 010037 027536      MOV      R0,WARN3
10830 041210 010037 027562      MOV      R0,WARN4
10831 041214 010037 027576      MOV      R0,WARN5
10832 041220              POP      R0
10833 041222 000207      RETURN

```

10835 041224

```

BACKGND:SUBTST  (<SUBR  WRITE BACKGROUND>>
:*****
:*SUBTEST      SUBR  WRITE BACKGROUND
:*****

```

PC	OPCODE	OPERAND1	OPERAND2	OPERAND3	OPERAND4	INSTR	COMMENT
10836						:WRITES DATA FROM R2	
10837	041224	104415				SAVREG	
10838	041226	012700	060000			MOV #FIRST,R0	
10839	041232	012701	040000			MOV #SIZE,R1	
10840	041236	022737	000001	003752		CMP #1,PROTYP	
10841	041244	001415				BEQ WARN6B	
10842	041246	012737	000207	027404	WARN6A:	MOV #207,MTP000+4	:WARNING PUTTING 'RETURN' AFTER WRITE
10843	041254	012737	027400	002260		MOV #MTP000,SUPDOADD	
10844	041262	004737	027206			CALL SUPDO3	
10845	041266	012737	000240	027404		MOV #240,MTP000+4	:RESTORE 'NOP' AFTER WRITE
10846	041274	104416				RESREG	
10847	041276	000207				RETURN	
10848	041300				WARN6B:	BMOV MTP000	
10849	041306	012737	000207	177644	WARN6:	MOV #207,UIPAR2	:WARNING PUTTING 'RETURN' INSTRUCTION AFTER WRITE
10850	041314	004737	027030			CALL SUPDO1	
10851	041320	104416				RESREG	
10852	041322	000207				RETURN	

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 329  
 SUBR WRITE BACKGROUND

10854 041324

```

GETCSR: SUBTST <<SUBR GET CSR INFORMATION FROM CONFIGURATION TABLE>>
:*****
:*SUBTEST      SUBR      GET CSR INFORMATION FROM CONFIGURATION TABLE
:*****
:INPUTS : NONE
:OUTPUT : CSRNO = CSR NUMBER
:
MOV      BANKINDEX,R2      ;GET INDEX INTO CONFIG TABLE
MOV      CONFIG(R2),R3     ;MOV IT INTO R3
SWAB     R3
:
ASL      R3
:
BIC      #*C36,R3          ;CLEAR OFF SOME BITS
MOV      R3,CSRNO          ;SAVE CSR NUMBER
RETURN
  
```

10855  
 10856  
 10857  
 10858  
 10859 041324 013702 002102  
 10860 041330 016203 002650  
 10861 041334 000303  
 10862 041336 006303  
 10863 041340 042703 177741  
 10864 041344 010337 002150  
 10865 041350 000207

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 331  
 SUBR GET CSR INFORMATION FROM CONFIGURATION TABLE

10868 041352

PCONFIG:SUBTST &lt;&lt;SUBR PRINT CONFIGURATION MAP&gt;&gt;

\*\*\*\*\*  
 :\*SUBTEST SUBR PRINT CONFIGURATION MAP  
 :\*\*\*\*\*

10869 041352				PUSH	TKVEC,TKVEC+2,R0	
10870 041364	010637	041652		MOV	SF,PCONFS	;SAVE LAST GOOD SP
10871 041370	012737	041620	000060	MOV	#PCONF2,TKVEC	
10872 041376	012737	000340	000062	MOV	#340,TKVEC+2	
10873 041404	017700	141220		MOV	@\$TKB,R0	;KILL ANY OLD INTERRUPT
10874 041410	042737	000200	177776	BIC	#BIT7,PSW	;LOWER CPU PRIORITY TO 140
10875 041416	052777	000100	141202	BIS	#BIT6,@\$TKS	;ENABLE KEYBOARD INTERRUPTS
10876						
10877 041424				TYPE	MSG001	
10878 041430				TYPE	MSG002	
10879 041434				TYPE	MSG003	
10880 041440	022737	000060	002552	CMP	#60,LASTBANK	
10881 041446	002006			BGE	NOOJ	
10882				;IF FAT PAPER ON TERMINAL GOTO 1\$		
10883 041450				IF #SW4	SET.IN @SWR THEN JUMPTO PCONF1	
10884 041464	012700	000074		NOOJ:	MOV #60.,R0	
10885 041470	010004			MOV	R0,R4	
10886 041472				CLEAR	R1,R3	
10887 041476				TYPE	MSG004	
10888 041502	004737	041654		CALL	TCONFIG	;GO TYPE CONFIGURATION (1ST HALF)
10889 041506	022737	000060	002552	CMP	#60,LASTBANK	
10890 041514	002041			BGE	PCONF2	
10891 041516				TYPE	\$CRLF	
10892 041522				TYPE	MSG017	;PRINT SPACE(S)
10893 041526				TYPE	MSG011	
10894 041532				TYPE	\$CRLF	
10895 041536				TYPE	MSG017	;PRINT SPACE(S)
10896 041542				TYPE	MSG012	
10897 041546	012701	000360		MOV	#60.*2*2,R1	
10898 041552	010103			MOV	R1,R3	
10899 041554	004737	041654		CALL	TCONFIG	
10900 041560	000417			BR	PCONF2	
10901						
10902 041562	012700	000170		PCONF1:	MOV #120.,R0	
10903 041566	010004			MOV	R0,R4	
10904 041570				CLEAR	R1,R3	
10905 041574				TYPE	MSG014	;SPACE
10906 041600				TYPE	MSG011	
10907 041604				TYPE	MSG004	
10908 041610				TYPE	MSG012	
10909 041614	004737	041654		CALL	TCONFIG	
10910						
10911 041620	013706	041652		PCONF2:	MOV PCONFS,SP	;RESTORE STACK
10912 041624	042777	000100	140774	BIC	#BIT6,@\$TKS	
10913 041632	117700	140772		MOVB	@\$TKB,R0	;READ CHAR TO KILL FLAG
10914 041636				POP	R0,TKVEC+2,TKVEC	
10915 041650	000207			RETURN		
10916						
10917 041652	000000			PCONFS:	0	;STACK SAVED HERE!

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 333  
 SUBR PRINT CONFIGURATION MAP

10920 041654

```

SUBTST <<SUBR TYPE CONFIGURATION>>
*****
*SUBTEST SUBR TYPE CONFIGURATION
*****
:CALL: MOV #N,R0 ;N=NUMBER OF CHARACTERS
:      MOV R0,R4 ;BACKUP
:      MOV #K,R1 ;INDEX CONSTANT
:      MOV R1,R3 ;BACKUP
:      CALL TCONFIG ;ACTUAL CALL
:      RETURN ;ONLY RETURN
*****

```

10921  
 10922  
 10923  
 10924  
 10925  
 10926  
 10927  
 10928

10929  
 10930  
 10931  
 10932

```

*****
** ERROR **
*****
10933 041654 012737 000340 177776 TCONFIG:MOV #340,PSW ;DISABLE INTERRUPTS
10934 041662 TYPE MSG005
10935 041666 032761 000001 002650 1$: BIT #BIT0,CONFIG(R1) ;ERROR ON THIS BANK?
10936 041674 001403 BEQ 2$ ;NO - SKIP
10937 041676 TYPE MSG013 ;PRINT 'X'
10938 041702 000402 BR 3$
10939 041704 2$: TYPE MSG014 ;PRINT SPACE
10940 041710 062701 000004 3$: ADD #4,R1 ;BUMP POINTER
10941 041714 077014 SOB R0,1$ ;LOOP UNTIL DONE
10942 041716 010400 MOV R4,R0
10943 041720 010301 MOV R3,R1

```



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 335  
 SUBR TYPE CONFIGURATION

```

10946
10947
10948
10949 041722
10950
10951 041726 012737 000340 177776 TCFIG1:
10952 041734 032761 010000 002652
10953 041742 001014
10954 041744 032761 000002 002650
10955 041752 001004
10956 041754 112737 000040 074704
10957 041762 000424
10958 041764 112737 000055 074704 18$:
10959 041772 000420
10960 041774 016105 002650 1$:
10961 042000 042705 007777
10962 042004 000305
10963 042006 072527 177774
10964 042012 022705 000011
10965 042016 100002
10966 042020 062705 000007
10967 042024 062705 000060 2$:
10968 042030 110537 074704
10969 042034 16$:
10970 042040
10971 042050 062701 000004
10972 042054 077054
10973 042056 010400
10974 042060 010301
10975
10976
10977
10978
10979
10980 042062
10981 042066 033761 002104 002650 TCFIG2:
10982 042074 001432
10983 042076 016105 002652
10984 042102 000305
10985 042104 042705 177770
10986 042110 020527 000003
10987 042114 003022
10988 042116
10989 042124
10990 042132 112737 000120 074704
10991 042140
10992 042142 112737 000115 074704
10993 042150
10994 042150
10995 042152 112737 000114 074704
10996 042160
10997 042160 000403
10998 042162 112737 000040 074704 17$:
10999 042170 8$:
11000 042174
11001 042204 062701 000004
11002 042210 077052

*****
** INTERLEAVE **
*****
TYPE MSG007
:THIS IS AN ENTRY POINT FROM ERROR REPORTS
:DISABLE INTERUPTS
MOV #340,PSW
BIT #BIT12,CONFIG+2(R1)
BNE 1$
BIT #BIT1,CONFIG(R1)
BNE 18$
:IS THERE ANY MEMORY HERE?
:BRANCH IF MEMORY PRESENT.
MOVB #' ,MSG015
:MOVE A BLANK IN TO BE PRINTED
BR 16$
:BRANCH TO TYPE ROUTINE
MOVB #'- ,MSG015
BR 16$
MOV CONFIG(R1),R5
BIC #^C170000,R5
:GET CSR INTERLEAVE
SWAB R5
ASH #-4,R5
CMP #9,R5
BPL 2$
ADD #7,R5
ADD #60,R5
:MAKE ASCII
:PLUG INTO MEMORY
MOVB R5,MSG015
TYPE MSG015
IF NOTAB NE #0 THEN $RETURN
ADD #4,R1
SOB R0,TCFIG1
:LOOP UNTIL DONE
MOV R4,R0
MOV R3,R1

*****
** MEMORY TYPE **
*****
ENABL LSB
TYPE MSG009
BIT CPUBIT,CONFIG(R1)
BEQ 17$
MOV CONFIG+2(R1),R5
SWAB R5
BIC #^C7,R5
CMP R5,#3
BGT 17$
:GET MEMORY TYPE
:CLEAR NON INTERESTING BITS
:IS IT A LEGAL MEMORY TYPE
:IF IF SO BRANCH!!!!!!
IF #BIT0 SET.IN R5
:IS IT AN ECC MEMORY???
IF #BIT1 SET.IN R5
:IS IT A MS11-P OR A MS11-M???
MOVB #'P,MSG015
:IT IS A MS11-P
ELSE
MOVB #'M,MSG015
:IT IS A MS11-M
END
ELSE
MOVB #'L,MSG015
:IT IS A MS11-L
END
B? 8$
:
MOVB #' ,MSG015
TYPE MSG015
IF NOTAB NE #0 THEN $RETURN
ADD #4,R1
SOB R0,TCFIG2
:BUMP POINTER
:LOOP UNTIL DONE

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 335-1  
SUBR TYPE CONFIGURATION

11003	042212	010400			MOV	R4,R0	
11004	042214	010301			MOV	R3,R1	
11005					.DSABL	LSB	
11006							
11007					*****		
11008					** CSR **		
11009					*****		
11010	042216				TYPE	MSG016	
11011	042222	112737	000040	074704	TCFIG3: MOV	#',MSG015	
11012	042230	016105	002650		MOV	CONFIG(R1),R5	
11013	042234	032705	000002		BIT	#BIT1,R5	
11014	042240	001414			BEQ	16\$	
11015	042242	042705	170377		BIC	#^C7400,R5	
11016	042246	000305			SWAB	R5	
11017	042250	022705	000011		CMP	#9.,R5	
11018	042254	100002			BPL	10\$	
11019	042256	062705	000007		ADD	#7,R5	
11020	042262	062705	000060		ADD	#60,R5	;MAKE ASCII
11021	042266	110537	074704		MOVB	R5,MSG015	;PLUG INTO MEMORY
11022	042272				TYPE	MSG015	
11023	042276				IF NOTAB NE #0 THEN \$RETURN		
11024	042306	062701	000004		ADD	#4,R1	;BUMP POINTER
11025	042312	077035			SOB	R0,TCFIG3	
11026	042314	010400			MOV	R4,R0	
11027	042316	010301			MOV	R3,R1	
11028							
11029					*****		
11030					** PROTECTED **		
11031					*****		
11032	042320				TYPE	MSG010	
11033	042324	105761	002650		TSTB	CONFIG(R1)	;BANK PROTECTED?
11034	042330	100004			BPL	12\$	;NO - SKIP
11035	042332	112737	000120	074704	MOVB	#'P,MSG015	
11036	042340	000407			BR	13\$	
11037	042342	032761	000100	002650	12\$: BIT	#BIT6,CONFIG(R1)	;PROTECTED REGION OF ECC?
11038	042350	001406			BEQ	14\$	;NO - SKIP
11039	042352	112737	000111	074704	MOVB	#'I,MSG015	
11040	042360				TYPE	MSG015	
11041	042364	000402			BR	15\$	
11042	042366				14\$: TYPE	MSG014	;PRINT SPACE
11043	042372	062701	000004		ADD	#4,R1	;BUMP POINTER
11044	042376	077026			SOB	R0,11\$	;LOOP UNTIL DONE
11045	042400	010400			MOV	R4,R0	
11046	042402	010301			MOV	R3,R1	
11047	042404	000207			RETURN		

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 337  
 TRAP PARITY ERROR HANDLER

```

11050 .SBTTL TRAP PARITY ERROR HANDLER
11051 *****
11052 :VECTOR TO HERE FROM TRAPS TO 114
11053 :IGNORE ERRORS BUT COUNT IF NOPAR FLAG = 1.
11054 *****
11055
11056 CODE ACTION
11057 --0- PRINT-UNEXPECTED PARITY TRAP
11058 1 COUNT ERROR
11059 2 SET 'ABORT' / SETUP 'BADPC' / RETURN VIA PCBUMP
11060 3 RETURN VIA 'PARTHERE'
11061
11062
11063 042406 022737 000001 002074 PARITY: CMP #1,NOPAR ;COUNTING PARITY ERRORS?
11064 042414 001003 BNE 1$ ;NO - SKIP
11065 042416 005237 002070 INC PARCNT ;PARITY ERROR COUNTER + 1
11066 042422 000002 RTI
11067 042424 022737 000002 002074 1$: CMP #2,NOPAR ;ACTION CODE = 2 ?
11068 042432 001013 BNE 2$ ;NO - SKIP
11069 042434 SET ABORTFLAG ;YES
11070 042442 004737 042614 CALL BADSTACK ;FIND BAD SP,PC,PSW OFF STACK
11071 042446 063716 002322 ADD PCBUMP,(SP) ;UPDATE RETURN PC
11072 042452 042766 000004 000002 BIC #BIT2,2(SP) ;SHOW FAILURE BY .NE.
11073 042460 000002 RTI
11074 042462 022737 000003 002074 2$: CMP #3,NOPAR ;ACTION CODE = 3 ?
11075 042470 001003 BNE 3$ ;NO - SKIP
11076 042472 013716 002300 MOV PARTHERE,(SP)
11077 042476 000002 RTI
11078 042500 004737 042614 3$: CALL BADSTACK ;FIND BAD SP,PC,PSW OFF STACK
11079 042504 FATAL 32

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 339  
 TRAP NON-EXISTANT MEMORY (HOLES) HANDLER

```

11082 .SBTTL TRAP NON-EXISTANT MEMORY (HOLES) HANDLER
11083 :*****
11084 :VECTOR TO HERE (SOMETIMES) FROM TRAPS TO 4
11085 :CODE IN NONEM DETERMINES ACTION AS FOLLOWS:
11086 : 1) IGNORE ERRORS BUT COUNT IF NONEM (NO NON-EXISTANT MEMORY) FLAG = 1.
11087 : 2) TO EXIT PATTERN 0 DURING SIZING IF NON-EXIST MEM ERROR
11088 :*****
11089
11090 042512 022737 000001 002076 NONEXIST: CMP #1, NONEM ;COUNTING NON-EXISTANT MEMORY ERRORS?
11091 042520 001011 BNE 2$ ;NO - SKIP
11092 042522 005237 002066 INC NEMCNT ;BUMP NON-EXISTANT MEMORY COUNTER
11093 042526 022737 000001 002066 CMP #1, NEMCNT ;FIRST ERROR?
11094 042534 001002 BNE 1$ ;NO - SKIP
11095 042536 010037 002032 MOV R0, ADDRESS ;ASSUME R0 CONTAINS THE ADDRESS ACCESSED
11096 042542 000002 1$: RTI
11097 042544 005237 002066 2$: INC NEMCNT ;BUMP NON-EXISTANT MEMORY COUNTER
11098 042550 012701 000001 MOV #1, R1 ;DUMMY UP R1 FOR A FORCED SOB EXIT
11099 042554 000002 RTI
11100
11101 :*****
11102 .SBTTL TRAP TIMEOUT (TRAP TO 4) HANDLER
11103 042556 004737 042614 TIMEOUT: CALL BADSTACK ;FIND BAD SP, PC, PSW OFF STACK
11104 042562 FATAL 6
11105 :*****
11106 .SBTTL TRAP MEMORY MANAGEMENT (TRAP TO 250) HANDLER
11107 042570 004737 042614 MMTRAP: CALL BADSTACK ;FIND BAD SP, PC, PSW OFF STACK
11108 042574 FATAL 7
11109 .SBTTL TRAP RESERVED INSTRUCTION HANDLER
11110 042602 004737 042614 PDP1105: CALL BADSTACK ;FIND BAD SP, PC, PSW OFF STACK
11111 042606 FATAL 5
11112
11113
11114
11115
11116
11117
11118
11119 042614 BADSTACK: SUBTST <<FIND BAD SP, PC, & PSW FROM STACK>>
:*****
: *SUBTEST FIND BAD SP, PC, & PSW FROM STACK
:*****
11120 042614 010637 002024 MOV SP, BADSP
11121 042620 062737 000002 002024 ADD #2, BADSP
11122 042626 016637 000002 002020 MOV 2(SP), BADPC
11123 042634 016637 000004 002030 MOV 4(SP), BADPSW
11124 042642 000207 RETURN

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 341  
 TRAP KERNEL TRAP HANDLER

```

11127          .SBTTL TRAP    KERNEL TRAP HANDLER
11128          ;*****
11129          ;KERNEL IS A TRAP THAT COMES HERE
11130          ;*****
11131
11132 042644 042766 140000 000002 $KERNEL:      BIC      #140000,2(SP)
11133 042652 000002          RTI
11134          ;*****
11135          .SBTTL TRAP    ENERGIZE TRAP HANDLER
11136 042654 052737 000001 177572 $ENERGIZE:BIS  #BIT0,MMR0
11137 042662 000002          RTI
11138          ;*****
11139          .SBTTL TRAP    DEENERGIZE TRAP HANDLER
11140 042664 042737 000001 177572 $DEENERGIZE:BIC #BIT0,MMR0
11141 042672 000002          RTI
11142          ;*****
11143          .SBTTL TRAP    CACHON TRAP HANDLER
11144 042674 005737 002540 $CACHN: TST    CACHKN          ;IS THERE A CACHE
11145 042700 001406          BEQ     1$          ;NO - RETURN
11146 042702 013737 002540 177746          MOV    CACHKN,CONTRL ;SETUP CACHE AS PER CONSTANT (USUALLY 1 = FULLY ON)
11147 042710 052737 000001 177746          BIS    #BIT0,CONTRL ;DISABLE TRAPS (BUT NOT ABORTS)
11148 042716 000002          1$:      RTI
11149          ;*****
11150          .SBTTL TRAP    CACHOFF TRAP HANDLER
11151 042720 005737 002540 $CACHF: TST    CACHKN          ;IS THERE A CACHE?
11152 042724 001403          BEQ     1$          ;NO - RETURN
11153          ;DISABLE TRAPS (NOT ABORTS), FORCE MISSES, FLUSH, BYPASS
11154 042726 053737 002544 177746          BIS    CACHKF,CONTRL
11155 042734 000002          1$:      RTI

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 343  
 TRAP LOAD CSR TRAP HANDLER

```

11158 .SBTTL TRAP LOAD CSR TRAP HANDLER
11159 ;LOAD CORRECT CSR WITH DATA IN CSR
11160 ;PROGRAM CSR'S ASSERT INHIBIT MODE POINTER WHEN LOADED
11161 042736 $LOADC: PUSH R0,R1 ;SAVE REGISTERS
11162 042742 013700 002150 MOV CSRNO,R0 ;CREATE CSR ADDRESS
11163 042746 IF INHECC IS TRUE THEN GOTO 3$ ;DON'T WANT INH. MODE POINTER ON
11164 042754 005737 002526 TST PGMCSR ;PROGRAM IN INTERLEAVED SPACE?
11165 042760 100007 BPL 1$ ;BRANCH IF NOT
11166 042762 113701 002527 MOV PGMCSR+1,R1 ;CHECK SECOND CSR
11167 042766 042701 177740 BIC #^C37,R1 ;CLEAR UNNECESSARY BITS
11168 042772 020137 002150 CMP R1,CSRNO ;IS THIS THE CURRENT CSR?
11169 042776 001404 BEQ 2$ ;BRANCH IF IT IS
11170 043000 123737 002526 002150 1$: CMPB PGMCSR,CSRNO ;IS THIS THE CURRENT CSR?
11171 043006 001003 BNE 3$ ;BRANCH IF NOT
11172 043010 052737 020000 002146 2$: BIS #BIT13,CSR ;SET THE INHIBIT MODE POINTER TO 1ST 16K
11173 043016 013760 002146 172100 3$: MOV CSR,CSRADD(R0) ;LOAD THE CSR
11174 043024 POP R1,R0 ;RESTORE REGISTERS
11175 043030 U00002 RTI
11176
11177 .SBTTL TRAP READ CSR TRAP HANDLER
11178 ;READ THE CORRECT CSR INTO LOCATIONS CSR
11179 043032 $READC: PUSH R0
11180 043034 013700 002150 MOV CSRNO,R0
11181 043040 016037 172100 002146 MOV CSRADD(R0),CSR ;READ IT
11182 043046 POP R0
11183 043050 000002 RTI

```

11185					.SBTTL	TRAP TEST (R1) & READ CSR CAREFULLY	
11186	043052				\$TSTRD: PUSH	R0,R2,R3	
11187	043060	012700	172100		MOV	#CSRADD,R0	:CREATE CSR ADDRESS
11188	043064	063700	002150		ADD	CSRNO,R0	
11189	043070	005002			CLR	R2	
11190	043072	005737	002526		TST	PGMCSR	
11191	043076	100007			BPL	1\$	
11192	043100	113703	002527		MOVB	PGMCSR+1,R3	
11193	043104	042703	000200		BIC	#BIT7,R3	
11194	043110	020337	002150		CMP	R3,CSRNO	
11195	043114	001404			BEQ	2\$	
11196	043116	123737	002526	002150	CMPSB	PGMCSR,CSRNO	
11197	043124	001002			BNE	3\$	
11198	043126	012702	020000		MOV	#BIT13,R2	
11199	043132	022737	000001	003752	CMP	#1,PROTYP	:IS THIS AN 11/44?
11200	043140	001403			BEQ	4\$	:BRANCH IF IT IS
11201	043142	004737	043230		CALL	TSTRD1	
11202	043146	000405			BR	5\$	
11203	043150				BMOV	TSTRD1	
11204	043156	004737	177640		CALL	FASTCITY	:CALL TO THE USER INSTRUCTION PAR'S
11205					:IF SINGLE BIT ERROR ONLY - SET CARRY BIT		
11206	043162				POP	R3,R2,R0	
11207	043170				IF #BIT4 SET.IN CSR AND #BIT15 OFF.IN CSR		
11208	043210	052766	000001	000002	BIS	#BIT0,2(SP)	
11209	043216				ELSE		
11210	043220	042766	000001	000002	BIC	#BIT0,2(SP)	
11211	043226				END ;OF IF #BIT4		
11212	043226	000002			RTI		
11213							
11214	043230	010210			TSTRD1: MOV	R2,(R0)	:V177640
11215	043232				TESTAREA		:V177642 :ENTER SUPERVISOR MODE
11216	043240	105711			TSTB	(R1)	:V177646
11217	043242	042737	140000	177776	BIC	#BIT15!BIT14,PSW	:V177650
11218	043250	011037	002146		MOV	(R0),CSR	:V177656
11219	043254	000207			RETURN		:V177662

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 346  
 TRAP ECC DISABLE ALL CSR'S TRAP HANDLER

11222					.SBTTL	TRAP	ECC DISABLE ALL CSR'S TRAP HANDLER
11223	043256	012737	000002	002146	SECCDIS:MOV	#BIT1,CSR	
11224	043264	004737	044002		CALL	CSR0UT	
11225	043270	000002			RTI		
11226					.SBTTL	TRAP	ECC DISABLE OF 1 SELECTED CSR TRAP HANDLER
11227	043272	012737	000002	002146	SECC1DIS:MOV	#BIT1,CSR	
11228	043300	104425			LOADCSR		
11229	043302	000002			RTI		
11230					.SBTTL	TRAP	INITIALIZE ALL CSR'S TRAP HANDLER
11231	043304	012737	000001	002146	SECCINIT:MOV	#BIT0,CSR	
11232	043312	004737	044002		CALL	CSR0UT	
11233	043316	000002			RTI		
11234					.SBTTL	TRAP	INITIALIZE 1 SELECTED CSR TRAP HANDLER
11235	043320	012737	000001	002146	SECC1INIT:MOV	#BIT0,CSR	
11236	043326	104425			LOADCSR		
11237	043330	000002			RTI		
11238					.SBTTL	TRAP	ENABLE SBE PARITY TRAPS ON ALL CSR'S
11239	043332	012737	000003	002146	SENASBE:MOV	#BIT0!BIT1,CSR	
11240	043340	004737	044002		CALL	CSR0UT	
11241	043344	000002			RTI		
11242					.SBTTL	TRAP	ENABLE SBE PARITY TRAPS ON 1 SELECTED CSR
11243	043346	012737	000003	002146	SENA1SBE:MOV	#BIT0!BIT1,CSR	
11244	043354	104425			LOADCSR		
11245	043356	000002			RTI		
11246					.SBTTL	TRAP	WRITE CHECKBITS THRU ALL CSR'S TRAP HANDLER
11247	043360	013737	002310	002146	SCBCSR:MOV	CHECK,CSR	;BITS 11-5
11248	043366	052737	000006	002146	BIS	#BIT1!BIT2,CSR	;CHECK MODE
11249	043374	004737	044002		CALL	CSR0UT	
11250	043400	000002			RTI		
11251					.SBTTL	TRAP	WRITE CHECKBITS THRU 1 SELECTED CSR TRAP HANDLER
11252	043402	013737	002310	002146	SCB1CSR:MOV	CHECK,CSR	;BITS 11-5
11253	043410	052737	000006	002146	BIS	#BIT1!BIT2,CSR	;CHECK MODE
11254	043416	104425			LOADCSR		
11255	043420	000002			RTI		



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 348  
 TRAP WAS THERE A SBE ON ANY CSR TRAP HANDLER

```

11258 .SBTTL TRAP WAS THERE A SBE ON ANY CSR TRAP HANDLER
11259 043422 $WASSBE: PUSH R1,R4
11260 043426 013701 002222 MOV TOTCSRS,R1 ;GET CSR'S BYTE
11261 043432 005004 CLR R4
11262 043434 BEGIN LWSBE
11263 043434 FOR CSRNO := #0 TO #36 BY #2
11264 043440 006301 ASL R1
11265 043442 ON.ERROR
11266 043444 104426 READCSR
11267 043446 IF #BIT4 SET.IN CSR
11268 043456 SET R4
11269 043462 LEAVE LWSBE
11270 043464 END :OF IF #BIT4
11271 043464 END :OF ON.ERROR
11272 043464 IF R1 EQ #0 THEN LEAVE LWSBE
11273 043470 END :OF FOR CSRNO
11274 043506 END LWSBE
11275 043506 006004 ROR R4 ;SET C BIT FOR ERROR
11276 043510 POP R4,R1
11277 043514 ON.ERROR
11278 043516 052766 000001 000002 BIS #BIT0,2(SP)
11279 043524 ELSE
11280 043526 042766 000001 000002 BIC #BIT0,2(SP)
11281 043534 END :OF ON.ERROR
11282 043534 000002 RTI
11283 .SBTTL TRAP WAS THERE A SBE IN 1 SELECTED CSR TRAP HANDLER
11284 ;ON RETURN IF CARRY IS SET THERE WAS A SBE
11285 043536 104426 $WAS1SBE: READCSR
11286 043540 042766 000001 000002 BIC #BIT0,2(SP) ;CLR C BIT ON STACK
11287 043546 032737 000020 002146 BIT #BIT4,CSR
11288 043554 001403 BEQ 1$
11289 043556 052766 000001 000002 BIS #BIT0,2(SP) ;SET C BIT ON STACK
11290 043564 000002 1$: RTI

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 350  
 TRAP WAS THERE A DBE ON ANY CSR TRAP HANDLER

11293					.SBTTL TRAP WAS THERE A DBE ON ANY CSR TRAP HANDLER
11294	043566				\$WASDBE: PUSH R1,R4
11295	043572	013701	002222		MOV TOTCSRS,R1 ;GET CSR'S BYTE
11296	043576	005004			CLR R4
11297	043600				BEGIN LWDBE
11298	043600				FOR CSRNO := #0 TO #36 BY #2
11299	043604	006301			ASL R1
11300	043606				ON.ERROR
11301	043610	104426			READCSR
11302	043612				IF #BIT15 SET.IN CSR
11303	043622				SET R4
11304	043626				LEAVE LWDBE
11305	043630				END ;OF IF #BIT4
11306	043630				END ;OF ON.ERROR
11307	043630				IF R1 EQ #0 THEN LEAVE LWDBE
11308	043634				END ;OF FOR CSRNO
11309	043652				END LWDBE
11310	043652	006004			ROR R4 ;SET C BIT FOR ERROR
11311	043654				POP R4,R1
11312	043660				ON.ERROR
11313	043662	052766	000001	000002	BIS #BIT0,2(SP)
11314	043670				ELSE
11315	043672	042766	000001	000002	BIC #BIT0,2(SP)
11316	043700				END ;OF ON.ERROR
11317	043700	000002			RTI
11318					.SBTTL TRAP WAS THERE A DBE ON 1 SELECTED CSR TRAP HANDLER
11319					;ON RETURN IF CARRY IS SET THERE WAS A DBE
11320	043702	104426			\$WAS1DBE: READCSR
11321	043704	005737	002146		TST CSR ;DBE?
11322	043710	100004			BPL 3\$ ;NO - SKIP
11323	043712	052766	000001	000002	BIS #BIT0,2(SP) ;SET C BIT ON STACK
11324	043720	000002			RTI
11325	043722	042766	000001	000002	BIC #BIT0,2(SP) ;CLR C BIT ON STACK
11326	043730	000002			RTI

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 352  
 TRAP CLEAR ALL ECC CSR'S TRAP HANDLER

11329						.SBTTL	TRAP	CLEAR ALL ECC CSR'S TRAP HANDLER
11330	043732					\$CLRCSR: CLEAR	CSR	
11331	043736	004737	044002			CALL	CSR OUT	
11332	043742	000002				RTI		
11333						.SBTTL	TRAP	CLEAR 1 SELECTED CSR TRAP HANDLER
11334	043744					\$CLR1CSR: CLEAR	CSR	
11335	043750	104425				LOADCSR		
11336	043752	000002				RTI		
11337						.SBTTL	TRAP	ECC DISABLE, CHECK MODE, & WRITE CHECKBITS IN ALL CSR'S TRAP HANDLER
11338						;CHECKBITS	ALREADY IN LOC 'CSR'	
11339	043754	052737	000006	002146	\$CHKDIS: BIS	#BIT1!BIT2, CSR		;ECC DISABLE & DIAG CHECK MODE
11340	043762	004737	044002			CALL	CSR OUT	
11341	043766	000002				RTI		
11342						.SBTTL	TRAP	ECC DISABLE, CHECK MODE, & WRITE CHECKBITS IN 1 SELECTED CSR
11343						;CHECKBITS	ALREADY IN LOC 'CSR'	
11344	043770	052737	000006	002146	\$CHK1DIS: BIS	#BIT1!BIT2, CSR		;ECC DISABLE & DIAG CHECK MODE
11345	043776	104425				LOADCSR		
11346	044000	000002				RTI		

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 354  
 TRAP ECC DISABLE, CHECK MODE, & WRITE CHECKBITS IN 1 SELECTED

11349 044002

CSR0UT: SUBTST <<SUBR WRITE IN ALL CSR'S>>

\*\*\*\*\*  
 :\*SUBTEST SUBR WRITE IN ALL CSR'S  
 \*\*\*\*\*

11350 044002

11351 044004 013701 002222

11352 044010

11353 044010

11354 044014 006301

11355 044016

11356 044020 104425

11357 044022

11358 044022

11359 044026

11360 044044

11361 044044

11362 044046 000207

11363

11364 044050

PUSH R1  
 MOV TOTCSRS,R1 ;GET CSR'S BYTE  
 BEGIN LCSROUT  
 FOR CSRNO := #0 TO #36 BY #2  
 ASL R1  
 ON.ERROR  
 LOADCSR  
 END ;OF ON.ERROR  
 IF R1 EQ #0 THEN LEAVE LCSROUT  
 END ;OF FOR CSRNO  
 END LCSROUT  
 POP R1  
 RETURN

\$INVALID: SUBTST <<TRAP INVALIDATE BACKGROUND PATTERN>>

\*\*\*\*\*  
 :\*SUBTEST TRAP INVALIDATE BACKGROUND PATTERN  
 \*\*\*\*\*

11365 044050

11366 044054 013701 002100

11367 044060 006301

11368 044062 006301

11369 044064 042761 020000 002652

11370 044072

11371 044076 000002

PUSH R0,R1  
 MOV BANK,R1  
 ASL R1  
 ASL R1  
 BIC #BIT13,CONFIG+2(R1)  
 POP R1,R0  
 RTI

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 355  
 TRAP INVALDATE BACKGROUND PATTERN

11373 044100

```

$ERRGEN:      SUBTST<<TRAP  GENERATE AND TEST ERROR ADDRESS>>
:*****
:*SUBTEST     TRAP  GENERATE AND TEST ERROR ADDRESS
:*****
11374 044100   PUSH    R0,R1,R2,R3
11375 044110   013703 002102   MOV    BANKINDEX,R3
11376 044114   005737 002452   TST    NOSJPER
11377 044120   001003          BNE     6$
11378 044122   013700 172246   MOV    S1PAR3,R0          ;GENERATE WHAT ERROR ADDR SHOULD BE
11379 044126   000402          BR     7$
11380 044130   013700 177646   6$:   MOV    U1PAR3,R0
11381 044134   072027 177773   7$:   ASH    #-5,R0
11382 044140   005737 002130   TST    EUFLAG
11383 044144   001002          BNE     1$
11384 044146   042700 177600   BIC    #^C177,R0
11385 044152   000301          1$:   SWAB   R1          ;GET CURRENT ADDRESS BITS 11 AND 12
11386 044154   006201          ASR    R1
11387 044156   006201          ASR    R1
11388 044160   006201          ASR    R1
11389 044162   042701 177775   BIC    #^C2,R1
11390 044166   060100          ADD    R1,R0          ;ADD THEM TO THE ADJUSTED PAR VALUE
11391          ;GET ERROR ADDRESS FROM CSR UNDER TEST
11392 044170   013701 002146   MOV    CSR,R1
11393 044174   072127 177773   ASH    #-5,R1
11394 044200   042701 177600   BIC    #^C177,R1
11395 044204   005737 002450   TST    N222BIT          ;IS THIS AN 11/44 OR 11/24?
11396 044210   001024          BNE     2$          ;BRANCH IF NOT NECESSARY
11397 044212   005737 002130   TST    EUFLAG          ;IS IT EUB?
11398 044216   001421          BEQ     2$          ;BRANCH IF NOT
11399 044220          PUSH   R0          ;SAVE GENERATED ERROR ADDRESS
11400 044222   013702 002150   MOV    CSRNO,R2          ;GET CSR NUMBER
11401 044226   052762 040000 172100   BIS    #BIT14,CSRADD(R2) ;TURN ON EUB BIT CAREFULLY
11402 044234   016200 172100   MOV    CSRADD(R2),R0     ;GET CSR CONTENTS
11403 044240   042762 040000 172100   BIC    #BIT14,CSRADD(R2) ;TURN OFF EUB BIT CAREFULLY
11404 044246   042700 177037   BIC    #^C740,R0        ;CLEAR EVERYTHING BUT ERROR ADDR
11405 044252   006300          ASL    R0
11406 044254   006300          ASL    R0          ;SHIFT ADDR BITS 18-21 INTO POSITION
11407 044256   060001          ADD    R0,R1          ;ADD TO CURRENT ERROR ADDRESS
11408 044260          POP    R0
11409 044262   020001          2$:   CMP    R0,R1          ;COMPARE REAL AND GENERATED ERR. ADDR.
11410 044264   001420          BEQ     5$          ;BRANCH IF THEY ARE THE SAME
11411 044266   005737 002134   TST    INTFLAG          ;INTERLEAVED?
11412 044272   001411          BEQ     3$          ;NO - WE HAVE AN ERROR
11413 044274   062700 000100   ADD    #100,R0
11414 044300   005737 002136   TST    INT64K          ;64K INTERLEAVED MEMORY?
11415 044304   001002          BNE     4$
11416 044306   062700 000100   ADD    #100,R0
11417 044312   020001          4$:   CMP    R0,R1
11418 044314   001404          BEQ     5$
11419 044316   005737 002064   3$:   TST    SKPERR
11420 044322   001001          BNE     5$
11421 044324   104462          PERR36
11422 044326   010137 002454   5$:   MOV    R1,ERRADD
11423 044332   005037 002064   CLR    SKPERR
11424 044336          POP    R3,R2,R1,R0
11425 044346   000002          RTI
11426

```

```

;ARE WE SUPPOSED TO SKIP ERROR P.O.?
;YES - SKIP ERROR PRINTOUT
;ELSE PRINT ERROR ADDRESS ERROR
;SAVE CSR'S ERROR ADDRESS
;ENABLE THE ERROR PRINTOUT AGAIN
;RESTORE REGISTERS

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 355-1  
 TRAP GENERATE AND TEST ERROR ADDRESS

11427 044350

\$CBREG: SUBTST <<TRAP ENABLE CHECK/SYNDROME BIT REGISTER>>  
 :\*\*\*\*\*  
 :\*SUBTEST TRAP ENABLE CHECK/SYNDROME BIT REGISTER  
 :\*\*\*\*\*

11428 044350 005037 002146  
 11429 044354 052737 000004 002146  
 11430 044362 104425  
 11431 044364 000002

CLR CSR  
 BIS #BIT2,CSR  
 LOADCSR  
 RTI  
 :  
 :ENABLE DIAGNOSTIC MODE  
 :LOAD CSR REGISTER  
 :

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 357  
 TRAP ENABLE CHECK/SYNDROME BIT REGISTER

11434 044366

```

CHKGEN: SUBTST<<SUBR    GENERATE CHECK BITS>>
:*****
:*SUBTEST    SUBR    GENERATE CHECK BITS
:*****
:CHECK BIT GENERATOR ROUTINE
:CALLING SEQUENCE IS:
:      MOV    #WORD1,SOURCE    ;SOURCE = ADDRESS OF DATA
:      CALL   CHKGEN
:
:CHECK BITS RETURNED IN BITS 11-5 OF LOCATION CHECK
:
:      PUSH   R0,R1,R2,R3,R4,R5
:      MOV    #77,R2            ;DEFAULT CHECKBITS FOR DOUBLE WORD OF ZEROS
:      MOV    #CHKTAB,R3        ;ADDRESS OF CHECKBIT TABLE
:      MOV    SOURCE,R5         ;GET SOURCE ADDRESS
:      MOV    (R5)+,R1          ;GET LSB'S
:      MOV    (R5),R0           ;GET MSB'S
:
1$:      SXT    R4                ;EXTEND SIGN OF DOUBLE WORD TO R4
:      BICB   (R3)+,R4          ;ELIMINATE BITS THAT DON'T COUNT
:      XOR    R4,R2             ;COMPLEMENT MASKED BITS IN CHECKBITS
:      ASHC   #1,R0             ;DOUBLE PRECISION LEFT SHIFT R0,,R1
:      BNE    1$               ;LOOP TILL ALL BITS ARE CHECKED
:
:      BIC    #^C177,R2         ;KILL ALL JUNK BITS
:      SWAB   R2                ;POSITION CHECKBITS IN BITS 11-5
:      ASR    R2
:      ASR    R2
:      ASR    R2
:      MOV    R2,CHECK
:      POP    R5,R4,R3,R2,R1,R0
:      RETURN

```

```

11435
11436
11437
11438
11439
11440
11441
11442 044366
11443 044402 012702 000077
11444 044406 012703 044474
11445 044412 013705 002306
11446 044416 012501
11447 044420 011500
11448
11449 044422 006704
11450 044424 142304
11451 044426 074402
11452 044430 073027 000001
11453 044434 001372
11454
11455 044436 042702 177600
11456 044442 000302
11457 044444 006202
11458 044446 006202
11459 044450 006202
11460 044452 010237 002310
11461 044456
11462 044472 000207

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 359  
 SUBR GENERATE CHECK BITS

11465	044474		CHKTAB: .BYTE #3	
11466	044474	200	.BYTE ^C177	:BIT 31
11467	044475	301	.BYTE ^C076	:BIT 30
11468	044476	302	.BYTE ^C075	:BIT 29
11469	044477	203	.BYTE ^C174	:BIT 28
11470	044500	304	.BYTE ^C073	:BIT 27
11471	044501	205	.BYTE ^C172	:BIT 26
11472	044502	206	.BYTE ^C171	:BIT 25
11473	044503	307	.BYTE ^C070	:BIT 24
11474			.BYTE #2	
11475	044504	310	.BYTE ^C067	:BIT 23
11476	044505	211	.BYTE ^C166	:BIT 22
11477	044506	212	.BYTE ^C165	:BIT 21
11478	044507	313	.BYTE ^C064	:BIT 20
11479	044510	214	.BYTE ^C163	:BIT 19
11480	044511	315	.BYTE ^C062	:BIT 18
11481	044512	316	.BYTE ^C061	:BIT 17
11482	044513	217	.BYTE ^C160	:BIT 16
11483			.BYTE #1	
11484	044514	320	.BYTE ^C057	:BIT 15
11485	044515	221	.BYTE ^C156	:BIT 14
11486	044516	222	.BYTE ^C155	:BIT 13
11487	044517	323	.BYTE ^C054	:BIT 12
11488	044520	224	.BYTE ^C153	:BIT 11
11489	044521	325	.BYTE ^C052	:BIT 10
11490	044522	6	.BYTE ^C051	:BIT 9
11491	044523	227	.BYTE ^C150	:BIT 8
11492			.BYTE #0	
11493	044524	340	.BYTE ^C037	:BIT 7
11494	044525	241	.BYTE ^C136	:BIT 6
11495	044526	242	.BYTE ^C135	:BIT 5
11496	044527	343	.BYTE ^C034	:BIT 4
11497	044530	244	.BYTE ^C133	:BIT 3
11498	044531	345	.BYTE ^C032	:BIT 2
11499	044532	346	.BYTE ^C031	:BIT 1
11500	044533	247	.BYTE ^C130	:BIT 0



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 361  
SUBR GENERATE CHECK BITS

11503 044534

11504  
11505  
11506  
11507  
11508  
11509  
11510  
11511  
11512  
11513  
11514 044534  
11515 044546 012700 172340  
11516 044552 012701 172240  
11517 044556 012704 172200  
11518 044562 005737 002452  
11519 044566 001404  
11520 044570 012701 177640  
11521 044574 012704 177600  
11522 044600 012702 077406  
11523 044604 012705 000010  
11524 044610 012021  
11525 044612 010224  
11526 044614 077503  
11527 044616 012741 177600  
11528  
11529  
11530 044622 022703 000170  
11531 044626 001516  
11532 044630 072327 000011  
11533  
11534 044634 012701 172246  
11535 044640 005737 002452  
11536 044644 001402  
11537 044646 012701 177646  
11538 044652 012702 000004  
11539 044656 010321  
11540 044660 062703 000200  
11541 044664 077204  
11542 044666 005737 002236  
11543 044672 001442  
11544 044674 162701 000010  
11545 044700 010102  
11546 044702 062702 000004  
11547 044706 022737 000001 002236  
11548 044714 001403  
11549 044716 010200  
11550 044720 010102  
11551 044722 010001  
11552 044724 012122  
11553 044726 011112  
11554 044730 013700 002102  
11555 044734 005737 002136  
11556 044740 001403

```

SUBTST<<SUBR  MAPPER>>
*****
*SUBTEST      SUBR  MAPPER
*****
:THIS SUBROUTINE MAPS THE MEMORY BANK (16K WORDS = 1 BANK)
:IN R3 TO THE TEST PATTERN AREA (SUPERVISOR VIRTUAL (60000 - 157777) FOR
:THE 11/44 AND 11/45-55; USER VIRTUAL (60000 - 157777) FOR ALL OTHER
:PDP-11'S).
:CALL      MOV      BANKNO,R3      ;SET UP BANK ARGUMENT
:          CALL     MAPPER          ;ACTUAL CALL
:          RETURN                    ;ONLY RETURN

MAPPER:      ;SET SUPERVISOR/USER UP FOR 1 TO 1 MAP
PUSH        R0,R1,R2,R4,R5
MOV         #KIPAR0,R0             ;FIRST AREA TO MAP TO
MOV         #SIPAR0,R1             ;FIRST ADDRESS REGISTER
MOV         #SIPDR0,R4             ;FIRST DESCRIPTOR REGISTER
TST         NOSUPER                ;CAN WE USE SUPERVISOR MODE?
BEQ         4$                     ;YES, BRANCH
MOV         #UIPAR0,R1             ;FIRST ADDRESS REGISTER
MOV         #UIPDR0,R4             ;FIRST DESCRIPTOR REGISTER
4$: MOV      #77406,R2              ;CONSTANT FOR 4K PAGE, UP, R/W
MOV         #8,R5                  ;COUNTER
1$: MOV      (R0)+,(R1)+            ;PUT IN SUPERVISOR ADDRESS
MOV         R2,(R4)+              ;PUT IN SUPERVISOR DESCRIPTOR
SOB         R5,1$                  ;LOOP TILL DONE
MOV         #177600,-(R1)          ;CORRECT LAST FIELD FOR PERIPHERALS PAGE

:SET UP SUPERVISOR/USER FOR TEST AREA
CMP         #120.,R3               ;MAP NOTHING (1 TO 1)?
BEQ         3$                     ;YES - SKIP
ASH         #9.,R3                 ;BANK 1 STARTS AT 100,000 LESS 6 LSB'S
MOV         #SIPAR3,R1             ;FOR MEMORY MANAGEMENT = 1000
TST         NOSUPER                ;SETUP FOR AUTO INCREMENTING
BEQ         5$                     ;DO WE HAVE SUPERVISOR MODE?
MOV         #UIPAR3,R1             ;YES - BRANCH
MOV         #4,R2                  ;SETUP FOR AUTO INCREMENTING
5$: MOV      R3,(R1)+              ;COUNTER
2$: MOV      #200,R3               ;PLUG IN PAR INFO
ADD         R2,2$                  ;BUMP ADDRESS 4K
SOB         R2,2$                  ;LOOP TILL DONE
TST         SPLTCSR
BEQ         9$
SUB         #10,R1
MOV         R1,R2
ADD         #4,R2
CMP         #1,SPLTCSR
BEQ         10$
MOV         R2,R0
MOV         R1,R2
MOV         R0,R1
10$: MOV      (R1)+,(R2)+          ;
MOV         (R1),(R2)
MOV         BANKINDEX,R0
TST         INT64K
BEQ         11$

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 361-1  
SUBR MAPPER

11557	044742	012700	004000		MOV	#4000,R0	
11558	044746	000402			BR	12\$	
11559	044750	012700	010000	11\$:	MOV	#10000,R0	
11560	044754	005737	002452	12\$:	TST	NOSUPER	
11561	044760	001403			BEQ	13\$	
11562	044762	012701	177652		MOV	#UIPAR5,R1	
11563	044766	000402			BR	14\$	
11564	044770	012701	172252	13\$:	MOV	#SIPAR5,R1	
11565	044774	060021		14\$:	ADD	R0,(R1)+	
11566	044776	060011			ADD	R0,(R1)	
11567					:IF WE ONLY HAVE AN 124K SYSTEM, WE DON'T WANT TO TEST THE		
11568					:LAST 4K, WHERE THE UNIBUS DEVICE PAGE IS. INSTEAD, THE		
11569					:PROGRAM WILL REMAP THE LAST 4K TO 8-12K. ALSO, IF THERE		
11570					:IS A BANK 177 ON AN 11/44, THE PROGRAM WILL REMAP THE LAST		
11571					:4K TO 8-12K FOR THE SAME REASON.		
11572	045000	022737	000007	002552	9\$:	CMP	#7, LASTBANK
11573	045006	001010				BNE	7\$
11574	045010	005737	002450			TST	N022BIT
11575	045014	001423				BEQ	3\$
11576	045016	022737	000007	002100		CMP	#7, BANK
11577	045024	001017				BNE	3\$
11578	045026	000404				BR	8\$
11579	045030	022737	000177	002552	7\$:	CMP	#177, LASTBANK
11580	045036	001012				BNE	3\$
11581	045040	005737	002452		8\$:	TST	NOSUPER
11582	045044	001404				BEQ	6\$
11583	045046	013737	177652	177654		MOV	UIPAR5,UIPAR6
11584	045054	000403				BR	3\$
11585	045056	013737	172252	172254	6\$:	MOV	SIPAR5,SIPAR6
11586	045064				3\$:	POP	R5,R4,R2,R1,R0
11587	045076	000207				RETURN	
11588						.SBTTL	TRAP MAP KERNEL (ALMOST 1 TO 1) TRAP HANDLER
11589	045100				\$KMAP:	PUSH	R0,R1,R2,R3,R4
11590	045112	005000				CLR	R0
11591	045114	012701	172340			MOV	#KIPAR0,R1
11592	045120	012702	077406			MOV	#77406,R2
11593	045124	012703	172300			MOV	#KIPDR0,R3
11594	045130	012704	000010			MOV	#8,R4
11595	045134	010021			1\$:	MOV	R0,(R1)+
11596	045136	010223				MOV	R2,(R3)+
11597	045140	062700	000200			ADD	#200,R0
11598	045144	077405				SOB	R4,1\$
11599	045146	012741	177600			MOV	#177600,-(R1)
11600	045152	012741	177400			MOV	#177400,-(R1)
11607	045156					POP	R4,R3,R2,R1,R0
11608	045170	000002				RTI	

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 363  
 TRAP MAP KERNEL (ALMOST 1 TO 1) TRAP HANDLER

11611 045172

RELOCATE:SUBTST &lt;&lt;RELOCATE PROGRAM&gt;&gt;

```
*****
: *SUBTEST      RELOCATE PROGRAM
: *****
```

```
11612 045172      IF #SW12 SET.IN @SWR THEN $RETURN ERROR
11613 045206      IF APTFLAG IS TRUE OR ACTFLAG IS TRUE
11614 045222      IF $PASS NE #0 THEN $RETURN ERROR
11615 045234      END; OF IF APTFLAG
11616 045234      BEGIN LOADERBANK
11617 045234      FOR BANK := #1 TO LASTBANK
11618 045242      CALL EXBANK
11619 045246      IF ACFLAG IS TRUE AND PFLAG IS FALSE AND BMFLAG IS FALSE
11620 045270      MOV      BANK,R0
11621 045274      MOV      R0,LOADBANK
11622 045300      MOV      LOADHOME,R1
11623 045304      CALL      BANKMOV
11624 045310      CALL      NEWLOAD      ;MAP NEW LOADER BANK IN KERNEL
11625 045314      MOV      BANKINDEX,R1
11626 045320      BIS      #BIT15,CONFIG+2(R1)      ;MARK LOADER
11627 045326      BIC      #BIT13,CONFIG+2(R1)      ;INVALIDATE BACKGROUND PATTERN
11628 045334      LEAVE LOADERBANK
11629 045336      END ;OF IF ACFLAG
11630 045336      END ;OF FOR BANK
11631 045352      IF #SW13 OFF.IN @SWR
11632 045362      TYPE      MSG075      ;RELOCATION NOT POSSIBLE
11633 045366      END ;OF IF #SW13
11634 045366      $RETURN ERROR
11635 045372      END LOADERBANK
11636 045372      BEGIN FINDBANK
11637 045372      MOV      LASTBANK,R2
11638 045376      ASL      R2
11639 045400      ASL      R2      ;R2 <- R2 * 4
11640 045402      FOR R1 := #2*2 TO R2 BY #4
11641 045406      IF #BIT7!BIT0 OFF.IN CONFIG(R1) ;IF NO ERRORS & NOT PROGRAM SPACE
11642 045416      IF #BIT15 OFF.IN CONFIG+2(R1) ;IF NOT LOADER BANK
11643 045426      IF CPUBIT SET.IN CONFIG(R1) ;IF ACCESSABLE
11644 045436      IF #BIT8 OFF.IN CONFIG+2(R1) THEN LEAVE FINDBANK ;IF PARITY
11645 045446      IF #BIT6 SET.IN CONFIG(R1) AND #BIT7 OFF.IN CONFIG(R1)
11646      ;IF 1ST PROTECTABLE ECC BANK
11647 045466      LEAVE FINDBANK
11648 045470      END ;OF IF #BIT6
11649 045470      IF INHECC IS FALSE
11650 045476      SET      INHECC
11651 045504      MOV      R1,INHBANK
11652 045510      END; OF IF INHECC
11653 045510      END ;OF IF CPUBIT
11654 045510      END ;OF IF #BIT15
11655 045510      END ;OF IF #BIT7
11656 045510      END ;OF FOR
11657 045520      IF FULLREL IS FALSE
11658 045526      IF INHECC IS TRUE
11659 045534      MOV      INHBANK,R1
11660 045540      CMP      REALPAT,#30      ;IS THIS PATTERN 30?
11661 045546      BEQ      RELENT1      ;YES - SKIP MESSAGE
11662 045550      BR      RELENT1
11663 045552      END; OF IF INHECC
11664 045552      END; OF IF FULLREL
```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 363-1  
RELOCATE PROGRAM

11665	045552	00503:	002532		CLR INHECC		;MAKE SURE FLAG IS TURNED OFF!
11666	045556				IF #SW13 OFF.IN @SWR		
11667	045566	023727	002274	000030	CMP REALPAT,#30		;IS THIS PATTERN 30?
11668	045574	001402			BEQ SKUB		;YES - SKIP MESSAGE
11669	045576				TYPE MSG075		;RELOCATION NOT POSSIBLE
11670	045602				END ;OF IF #SW13		
11671	045602				\$RETURN ERROR		
11672	045606				END FINDBANK		
11673	045606				CLEAR INHECC		;IF WE RELOCATED PROPERLY, THIS SHOULD BE OFF!
11674	045612	042761	020000	002652	RELENT1: BIC #BIT13,CONFIG+2(R1)		;INVALIDATE BACKGROUND PATTERN
11675	045620	005000			CLR R0		
11676	045622	071027	000004		DIV #4,R0		
11677	045626				RELOC1: LET NEWBANK := R0		
11678	045632	013737	002526	002530	MOV PGMCSR,PGMCSR+2		;SAVE CURRENT PGM. CSR
11679	045640	004737	046556		CALL USERMAP		;MAP NEWBANK TO USER PAR
11680	045644				USER		;ENTER USER MODE
11681	045652				BMOV 0,100000,SIZE		;MOVE PROGRAM
11682	045664	104417			KERNEL		;ENTER KERNEL MODE
11683	045666	022737	000001	003752	CMP #1,PROTYP		;IS THIS AN 11/44 ?
11684	045674	001021			BNE JMPRL1		;JUMP IF NOT
11685	045676	042737	000040	172516	BIC #BIT5,MMR3		;TURN OFF UNIBUS MAP
11686	045704	013700	002304		MOV NEWBANK,R0		
11687	045710	006200			ASR R0		
11688	045712				ON.ERROR		
11689	045714	012737	100000	170200	MOV #BIT15,MAPLO		
11690	045722				END ;OF ON.ERROR		
11691	045722	010037	170202		MOV R0,MAPH0		
11692	045726	004737	046344		CALL LOWMAP		;SETUP LOWER 16K IN UNIBUS MAP
11693	045732	052737	000040	172516	BIS #BIT5,MMR3		;ENERGIZE UNIBUS MAP
11694	045740	042737	000001	177572	JMPRL1: BIC #BIT0,MMR0		;DEENERGIZE MEMORY MANAGEMENT
11695	045746	004737	046640		CALL NEWKERNEL		
11696	045752	013700	002304		MOV NEWBANK,R0		
11697	045756	006300			ASL R0		
11698	045760	006300			ASL R0		;R0 <- R0 * 4
11699	045762	016002	002650		MOV CONFIG(R0),R2		
11700	045766	000302			SWAB R2		
11701	045770	042702	177760		BIC #^C17,R2		
11702	045774	006302			ASL R2		
11703	045776	052737	000001	177572	BIS #BIT0,MMR0		;ENERGIZE MEMORY MANAGEMENT
11704	046004	010237	002526		MOV R2,PGMCSR		;PUT NEW PGM. CSR INTO PGMCSR
11705	046010	032760	010000	002652	BIT #BIT12,CONFIG+2(R0)		;IS THE NEW BANK INTERLEAVED?
11706	046016	001412			BEQ 1\$		;BRANCH IF NOT INTERLEAVED
11707	046020	016002	002650		MOV CONFIG(R0),R2		
11708	046024	042702	007777		BIC #^C170000,R2		
11709	046030	072227	177775		ASH #-3,R2		
11710	046034	052702	100000		BIS #BIT15,R2		
11711	046040	050237	002526		BIS R2,PGMCSR		
11712	046044				1\$: SET RLFLAG		
11713	046052				\$RETURN NOERROR		

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 365  
RELOCATE PROGRAM

11716 046056

```

UNRELOCATE:SUBTST      <<UNRELOCATE PROGRAM>>
:*****
:*SUBTEST      UNRELOCATE PROGRAM
:*****
:RESTORE LOADERS
PUSH      R0
MOV      LOADBANK,R1
MOV      LOADHOME,R0
CALL     BANKMOV
CALL     NEWLOAD      ;MAP NEW LOADER BANK IN KERNEL SPACE
PUSH     BANK
MOV      LOADBANK,BANK
CALL     EXBANK
MOV      BANKINDEX,R1
BIC      #BIT15,CONFIG+2(R1) ;CLEAR LOADER FLAG
MOV      LOADHOME,BANK
CALL     EXBANK
MOV      BANKINDEX,R1
BIC      #BIT13,CONFIG+2(R1) ;INVALIDATE BACKGROUND PATTERN
POP      BANK
CLEAR    INHECC      ;MAKE SURE ECC TESTS ARE NOT INHIBITED!

:RESTORE BANK 0
BIC      #BIT13,CONFIG+2 ;INVALIDATE BACKGROUND PATTERN
LET NEWBANK := #0
CALL     USERMAP      ;MAP NEWBANK TO USER PAR
USER      ;ENTER USER MODE
BMOV     0,100000,SIZE ;MOVE PROGRAM
KERNEL   ;ENTER KERNEL MODE
BIC      #BIT0,MMR0     ;DEENERGIZE MEMORY MANAGEMENT
CALL     NEWKERNEL
MOV      PGMCSR+2,PGMCSR ;RESTORE PREVIOUS PGM. CSR
BIS      #BIT0,MMR0     ;ENERGIZE MEMORY MANAGEMENT
CLR      RLFLAG
CMP      #1,PROTYP      ;IS THIS AN 11/44 ?
BNE      1$
BIC      #BIT5,MMR3      ;TURN OFF UNIBUS MAP
CLEAR    MAPLO,MAPHO
CALL     LOWMAP         ;SETUP LOWER 16K OF UNIBUS MAP
BIS      #BIT5,MMR3      ;ENERGIZE UNIBUS MAP
1$: MOV    #CONFIG+2,R0    ;MOVE 2ND WORD OF CONFIG TO R0
2$: BIC    #BIT13,(R0)    ;CLEAR BACKGROUND VALID BIT
ADD      #4,R0           ;INCREMENT TO NEXT BANK
CMP      R0,#3620        ;DONE?
BLE      2$             ;NO - BRANCH
POP      R0
RETURN

```

```

11717
11718 046056
11719 046060 013701 002426
11720 046064 013700 002562
11721 046070 004737 046410
11722 046074 004737 046742
11723 046100
11724 046104 013737 002426 002100
11725 046112 004737 047020
11726 046116 013701 002102
11727 046122 042761 100000 002652
11728 046130 013737 002562 002100
11729 046136 004737 047020
11730 046142 013701 002102
11731 046146 042761 020000 002652
11732 046154
11733 046160
11734
11735
11736 046164 042737 020000 002652
11737 046172
11738 046176 004737 046556
11739 046202
11740 046210
11741 046222 104417
11742 046224 042737 000001 177572
11743 046232 004737 046640
11744 046236 013737 002530 002526
11745 046244 052737 000001 177572
11746 046252 005037 002124
11747 046256 022737 000001 003752
11748 046264 001014
11749 046266 042737 000040 172516
11750 046274
11751 046304 004737 046344
11752 046310 052737 000040 172516
11753 046316 012700 002652
11754 046322 042710 020000
11755 046326 062700 000004
11756 046332 020027 003620
11757 046336 003771
11758 046340
11759 046342 000207
11760
11761 046344

```

```

LOWMAP: SUBTST <<SETUP LOWER 16K OF UNIBUS MAP>>
:*****
:*SUBTEST      SETUP LOWER 16K OF UNIBUS MAP
:*****
PUSH      R0,R1,R2
MOV      #MAPLO,R0
MOV      #MAPL1,R1
MOV      #3,R2
1$: MOV    (R0)+,(R1)

```

```

11762 046344
11763 046352 012700 170200
11764 046356 012701 170204
11765 046362 012702 000003
11766 046366 012011

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 365-1  
SETUP LOWER 16K OF UNIBUS MAP

11767 046370 062721 020000  
11768 046374 012021  
11769 046376 077205  
11770 046400  
11771 046406 000207

ADD #BIT13,(R1)+  
MOV (R0)+,(R1)+  
SOB R2,1\$  
POP R2,R1,R0  
RETURN

11774 046410

\*\*\*\*\*  
 \*SUBTEST            MOVE BANKS  
 \*\*\*\*\*

```

:MOVE 3/4 OF A BANK
:CALLING SEQUENCE
:R0 = DESTINATION BANK
:R1 = SOURCE BANK
SAVREG
CALL    USERMAP
RESREG
SAVREG
ASH     #9.,R0
ASH     #9.,R1
MOV     #UIPAR4,R2
MOV     #200.R3

MOV     R1,(R2)+           ;MAP 1ST HALF BANK
ADD     R3,R1              ;BUMP BY 4K
MOV     R1,(R2)+
ADD     R3,R1

MOV     R0,(R2)+
ADD     R3,R0
MOV     R0,(R2)+
ADD     R3,R0

USER
BMOV    100000,140000,SIZE/2 ;MOV 1ST HALF BANK
KERNEL ;ENTER KERNEL MODE

MOV     #UIPAR4,R2

MOV     R1,(R2)+           ;MAP 2ND HALF BANK
ADD     R3,R1              ;BUMP BY 4K
MOV     R1,(R2)+
ADD     R3,R1

MOV     R0,(R2)+
ADD     R3,R0
MOV     R0,(R2)+
ADD     R3,R0

USER
BMOV    100000,140000,SIZE/4 ;MOV 3RD FOURTH OF BANK
KERNEL ;ENTER KERNEL MODE

RESREG
RETURN

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 369  
MOVE BANKS

11822 046556

USERMAP:SUBTST &lt;&lt;SUBR MAP USER TO NEW BANK&gt;&gt;

\*\*\*\*\*  
:SUBTEST SUBR MAP USER TO NEW BANK  
\*\*\*\*\*

11823 046556 012701 177640  
11824 046562 012702 172340  
11825 046566 012703 177600  
11826 046572 012704 172300  
11827 046576 012705 000004  
11828 046602 012221  
11829 046604 011423  
11830 046606 077503

MOV #UIPAR0,R1 ;COPY KERNEL PAR'S & PDR'S (0-3)  
MOV #KIPAR0,R2  
MOV #UIPDR0,R3  
MOV #KIPDR0,R4  
MOV #4,R5  
1\$: MOV (R2)+,(R1)+  
MOV (R4),(R3)+  
SOB R5,1\$

11831  
11832 046610 013700 002304  
11833 046614 072027 000011  
11834  
11835 046620 012705 000004  
11836 046624 010021  
11837 046626 062700 000200  
11838 046632 011423  
11839 046634 077505  
11840 046636 000207  
11841

MOV NEWBANK,R0  
ASH #9.,R0 ;BANK 1 STARTS AT 100,000 LESS 6 LSB'S  
;FOR MEMORY MANAGEMENT = 1000  
MOV #4,R5  
2\$: MOV R0,(R1)+ ;SETUP UIPAR(4-7)  
ADD #200,R0 ;BUMP ADDRESS 4K  
MOV (R4),(R3)+ ;SETUP UIPDR(4-7)  
SOB R5,2\$  
RETURN

11842 046640

NEWKERNEL:SUBTST &lt;&lt;SUBR SETUP KERNEL PAR'S FOR NEW BANK&gt;&gt;

\*\*\*\*\*  
:SUBTEST SUBR SETUP KERNEL PAR'S FOR NEW BANK  
\*\*\*\*\*

11843 046640  
11844 046646 012700 172340  
11845 046652 013701 002304  
11846 046656 072127 000011  
11847  
11848 046662 012705 000004  
11849 046666 010120  
11850 046670 062701 000200  
11851 046674 077504  
11852 046676  
11853 046704 000207  
11854  
11855 046706

PUSH R0,R1,R5  
MOV #KIPAR0,R0  
MOV NEWBANK,R1  
ASH #9.,R1 ;BANK 1 STARTS AT 100,000 LESS 6 LSB'S  
;FOR MEMORY MANAGEMENT = 1000  
MOV #4,R5  
1\$: MOV R1,(R0)+ ;SETUP KIPAR(0-3)  
ADD #200,R1  
SOB R5,1\$  
POP R5,R1,R0  
RETURN

MAPKERNAL:SUBTST &lt;&lt;SUBR MAP KERNAL PARS 4 AND 5 TO A BANK&gt;&gt;

\*\*\*\*\*  
:SUBTEST SUBR MAP KERNAL PARS 4 AND 5 TO A BANK  
\*\*\*\*\*

11856  
11857 046706 013705 002100  
11858 046712 072527 000011  
11859 046716 013737 172350  
11860 046724 010537 172350  
11861 046730 062705 000200  
11862 046734 010537 172352  
11863 046740 000207  
11864  
11865 046742

002266

MOV BANK,R5 ;MOV BANK NUMBER TO R5  
ASH #9.,R5 ;R5 ENTERS 100000 LESS SHIFT TO CREATE MAPPING  
MOV KIPAR4,SAVPAR ;SAVE OLD PAR  
MOV R5,KIPAR4 ;GET NEW PAR'S  
ADD #200,R5  
MOV R5,KIPAR5  
RETURN

NEWLOAD:SUBTST &lt;&lt;SUBR SETUP KERNEL PAR'S FOR NEW LOADER BANK&gt;&gt;

\*\*\*\*\*  
:SUBTEST SUBR SETUP KERNEL PAR'S FOR NEW LOADER BANK  
\*\*\*\*\*

11866

;R0 CONTAINS THE DESTINATION BANK



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 369-1  
 SUBR SETUP KERNEL PAR'S FOR NEW LOADER BANK

11867	046742				PUSH	R0,R1	
11868	046746	012701	172350		MOV	#KIPAR4,R1	
11869	046752	072027	000011		ASH	#9,R0	:BANK 1 STARTS AT 100000 LESS 6 LSB'S (1000)
11870	046756	010021			MOV	R0,(R1)+	:SETUP KIPAR4
11871	046760	062700	000200		ADD	#200,R0	
11872	046764	010021			MOV	R0,(R1)+	:SETUP KIPAR5
11873	046766				POP	R1,R0	
11874	046772	000207			RETURN		
11875							
11876	046774				UNMAP: SUBTST	<<SUBR UNMAP KERNAL PAR'S 4 AND 5>>	
					:*****		
					:*SUBTEST	SUBR UNMAP KERNAL PAR'S 4 AND 5	
					:*****		
11877	046774	013737	002266	172350	MOV	SAVPAR,KIPAR4	:RESTORE KIPAR4
11878	047002	062737	000200	002266	ADD	#200,SAVPAR	:ADD 200 FOR NEXT PAR
11879	047010	013737	002266	172352	MOV	SAVPAR,KIPAR5	:RESTORE KIPAR5
11880	047016	000207			RETURN		:

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 371  
 SUBR UNMAP KERNAL PAR'S 4 AND 5

11883 047020

EXBANK: SUBTST &lt;&lt;SUBR EXAMINE BANK&gt;&gt;

\*\*\*\*\*  
 \*SUBTEST SUBR EXAMINE BANK  
 \*\*\*\*\*

DOES THE FOLLOWING:

(1) SETS UP 'BANKINDEX' AND R1 BASED ON VALUE OF 'BANK'.  
 (2) SETS THE 'MKFLAG' IF THE BANK IS ECC.  
 (3) SETS THE 'KPFLAG' IF THE BANK IS THE PROTECTED REGION OF ECC MEMORY.  
 (4) SETS THE 'ACFLAG' IF THE BANK CAN BE ACCESSED BY THIS CPU.  
 (5) SETS THE 'PFLAG' IF THE BANK IS IN PROGRAM SPACE.  
 (6) SETS THE 'RRFLAG' IF RELOCATION IS REQUIRED TO TEST THIS BANK; HOWEVER, IT COMPLEMENTS THIS FLAG IF THE RELOCATION FLAG 'RLFLAG' IS SET (THIS IS NECESSARY FOR THE USE OF THE RECURSIVE 'MODE' SUBROUTINES). THE 'RRFLAG' IS ALWAYS SET TO DISABLE TESTING IF FIELD SERVICE MODE 'SELECTED BANKS' ARE BEING TESTED AND THIS BANK IS NOT SELECTED.  
 (7) SETS THE 'BMFLAG' IF THE BANK IS A BAD MEMORY; HOWEVER, IT COMPLEMENTS THIS FLAG IF THE 'WORST' FLAG IS NOT SET (THIS IS NECESSARY FOR THE USE OF THE RECURSIVE 'MODE' SUBROUTINES).  
 (8) SETS THE 'INTFLAG' IF THE BANK IS INTERLEAVED.  
 (9) SETS THE 'INT64K' FLAG IF THE BANK IS INTERLEAVED ON 64K WORD BOUNDS.  
 (10) SETS THE 'SKIPMK' FLAG IF THIS BANK IS INTERLEAVED, AND HAS ALREADY BEEN TESTED.

(11) SETS THE 'PMEMFLG' IF THE ECC MEMORY UNDER TEST IS A MS11-P

11904 047020				PUSH	R0,R1,R2	
11905 047026				CLEAR	MKFLAG,KPFLAG,PMEMFLAG	
11906 047042				SET	ACFLAG	
11907 047050				CLEAR	PFLAG,RRFLAG,BMFLAG	
11908 047064				CLEAR	INTFLAG,INT64K,SKIPMK	
11909 047100	013701	002100		MOV	BANK,R1	
11910 047104	006301			ASL	R1	
11911 047106	006301			ASL	R1	;R1 <- R1 * 4
11912 047110	010137	002102		MOV	R1,BANKINDEX	
11913 047114	032761	000100	002650	BIT	#BIT6,CONFIG(R1)	;PROTECTED REGION OF ECC MEMORY?
11914 047122	001403			BEQ	1\$	;NO - SKIP
11915 047124				SET	KPFLAG	
11916 047132	012700	000002		MOV	#BIT1,R0	
11917 047136				IF R0 SET,IN CPUBIT AND R0 OFF,IN CONFIG(R1)		
11918 047152	005037	002114		CLR	ACFLAG	
11919 047156				END ;OF IF R0		
11920 047156	005737	002114		TST	ACFLAG	;ACTIVE MEMORY?
11921 047162	001415			BEQ	2\$	;BRANCH IF NOT
11922 047164	016102	002652		MOV	CONFIG+2(R1),R2	
11923 047170	000302			SWAB	R2	
11924 047172	042702	177770		BIC	#^C7,R2	;ISOLATE MEM TYPE BITS
11925 047176	020227	000003		CMP	R2,#3	;IS THIS AN ILLEGAL MEM TYPE?
11926 047202	003405			BLE	2\$	;BRANCH IF NOT
11927 047204				SET	BMFLAG	;SET BAD BANK FLAG
11928 047212	000137	047436		JMP	ENEXBK	;JUMP OVER REST OF FLAG TESTS
11929 047216	032761	000400	002652	BIT	#BIT8,CONFIG+2(R1)	;IS THERE ECC THERE?
11930 047224	001412			BEQ	3\$	;NO - SKIP
11931 047226				SET	MKFLAG	;YES - SET MKFLAG
11932 047234	032761	001000	002652	BIT	#BIT9,CONFIG+2(R1)	;IS IT A MS11-P????
11933 047242	001403			BEQ	3\$	;NO SKIP!!!!
11934 047244				SET	PMEMFLG	;SET MS11-P FLAG
11935 047252	032761	000200	002650	BIT	#BIT7,CONFIG(R1)	;BANK = PROGRAM SPACE?
11936 047260	001406			BEQ	5\$	;NO - SKIP

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 371-1  
 SUBR EXAMINE BANK

11937	047262				SET	PFLAG,RRFLAG	
11938	047276	005737	002124	5\$:	TST	RLFLAG	;IS PROGRAM RELOCATED?
11939	047302	001402			BEQ	6\$	;NO - SKIP
11940	047304	005137	002122		COM	RRFLAG	;YES - COMPLEMENT RELOCATION REQUIRED FLAG
11941	047310	032761	000001	002650	6\$:	BIT	#BIT0,CONFIG(R1)
11942	047316	001403			BEQ	8\$	;ERRORS PRESENT IN THIS BANK?
11943	047320				SET	BMFLAG	;NO - SKIP
11944	047326	005737	002564	8\$:	TST	WORST	;IS THIS A WORST FIRST PASS?
11945	047332	001002			BNE	9\$	;YES - SKIP
11946	047334	005137	002126		COM	BMFLAG	;NO - COMPLEMENT BAD MEMORY FLAG
11947	047340			9\$:	IF SELONLY	IS TRUE AND #BIT14 OFF	;IN CONFIG+2(R1)
11948	047356				SET	RRFLAG	
11949	047364				END ;OF	IF SELONLY	
11950	047364	032761	010000	002652	BIT	#BIT12,CONFIG+2(R1)	;IS THIS BANK INTERLEAVED?
11951	047372	001421			BEQ	ENEXBK	;BRANCH IF IT IS NOT
11952	047374				SET	INTFLAG	
11953	047402	032761	004000	002652	BIT	#BIT11,CONFIG+2(R1)	;IS THIS BANK INTERLEAVED WITH 64K BOARDS?
11954	047410	001403			BEQ	10\$	;BRANCH IF IT IS NOT
11955	047412				SET	INT64K	
11956	047420	032761	000040	002650	10\$:	BIT	#BIT5,CONFIG(R1)
11957	047426	001403			BEQ	ENEXBK	;SHOULD THIS BANK BE TESTED?
11958	047430				SET	SKIPMK	;BRANCH IF IT SHOULD
11959	047436				ENEXBK:	POP	R2,R1,R0
11960	047444	000207			RETURN		;RESTORE REGISTERS

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 373  
 SUBR EXAMINE BANK

11963 047446

BANKOK: SUBTST <<SUBR BANK OK?>>

\*\*\*\*\*  
 :SUBTEST SUBR BANK OK?  
 \*\*\*\*\*

11964  
 11965  
 11966  
 11967 047446 013700 002132  
 11968 047452 005100  
 11969 047454 013701 002116  
 11970 047460 074001  
 11971 047462 000207

:TEST TO INSURE THAT THE TYPE OF MEMORY IN THE PRESENT BANK  
 :IS OF THE TYPE WE ARE TESTING 'TMFLAG'.  
 :RESULT IS RETURNED IN THE CONDITION CODES (OK = (=0)).  
 MOV TMFLAG,R0  
 COM R0  
 MOV MKFLAG,R1  
 XOR R0,R1  
 RETURN ;OK = (=OK)

11972  
 11973 047464  
 11974 047464

INCRPT:

INCPAT: SUBTST <<SUBR INCREMENT PATTERN TESTING >>

\*\*\*\*\*  
 :SUBTEST SUBR INCREMENT PATTERN TESTING  
 \*\*\*\*\*

11975  
 11976  
 11977 047464 005237 002110  
 11978 047470 022737 000030 002110  
 11979 047476 000207  
 11980  
 11981 047500  
 11982 047500

:INCREMENT THE PATTERN & SET UP THE CONDITION CODES  
 :RESULT - Z BIT SET INDICATES OVERFLOW  
 INC PATTERN  
 CMP #30,PATTERN ;SET UP CONDITION CODES  
 RETURN ;NOT EQUAL TO ZERO IS GOOD (NO OVERFLOW)

SETPAT:

HIPAT: SUBTST <<SUBR SET HIGHEST PATTERN TESTING TYPE>>

\*\*\*\*\*  
 :SUBTEST SUBR SET HIGHEST PATTERN TESTING TYPE  
 \*\*\*\*\*

11983 047500 012737 000027 002110  
 11984 047506 000207  
 11985  
 11986 047510

MOV #27,PATTERN ;SET HIGHEST PATTERN  
 RETURN

INCBNK: SUBTST <<SUBR INCREMENT BANK & TEST>>

\*\*\*\*\*  
 :SUBTEST SUBR INCREMENT BANK & TEST  
 \*\*\*\*\*

11987  
 11988 047510 005237 002100  
 11989 047514 023737 002552 002100  
 11990 047522 000207

:RESULTS RETURNED IN CONDITION CODES  
 INC BANK  
 CMP LASTBANK,BANK ;TOO FAR?  
 RETURN

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 375  
 SUBR INCREMENT BANK & TEST

11993 047524

BOOT: SUBTST &lt;&lt;BOOTSTRAP ROUTINE&gt;&gt;

\*\*\*\*\*  
 :\*SUBTEST BOOTSTRAP ROUTINE  
 :\*\*\*\*\*

11994  
 11995  
 11996  
 11997  
 11998  
 11999  
 12000 047524 104472  
 12001 047526  
 12002 047534  
 12003 047546 004737 024566  
 12004 047552 104421  
 12005 047554 005737 002450  
 12006 047560 001003  
 12007 047562 042737 000040 172516  
 12008 047570 005001  
 12009 047572 000005  
 12010 047574 012700 177406  
 12011 047600 010160 000004  
 12012 047604 012710 177400  
 12013 047610 012740 000005  
 12014 047614 105710  
 12015 047616 100376  
 12016 047620 062701 020000  
 12017 047624 005710  
 12018 047626 100761  
 12019 047630 005007

:INITIALIZE ALL CSR'S  
 :UNRELOCATE IF NECESSARY  
 :FLUSH OUT ANY DBE'S  
 :TURN OFF MEMORY MANAGEMENT  
 :TURN OFF THE UNIBUS MAP  
 :BOOT RK0 OR RK1  
 ECCINIT ;TRAP ON DOUBLE BIT ERRORS (NORMAL)  
 SET4 #BOOT1 ;TRAPS TO 4 GOTO BOOT1  
 IF RLFLAG IS TRUE THEN \$CALL UNRELOCATE  
 CALL MT0030 ;FLUSH OUT DBE'S  
 DEENERGIZE ;TURN OFF MEMORY MANAGEMENT  
 TST NO22BIT ;IS THIS AN 11/44 OR 11/24?  
 BNE BOOT1  
 BIC #BIT5,MMR3 ;TURN OFF THE UNIBUS MAP  
 BOOT1: CLR R1  
 1\$: RESET  
 MOV #177406,R0  
 MOV R1,4(R0)  
 MOV #177400,(R0)  
 MOV #5,-(R0)  
 2\$: TSTB (R0)  
 BPL 2\$  
 ADD #BIT13,R1  
 TST (R0)  
 BMI 1\$  
 CLR PC

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 377  
BOOTSTRAP ROUTINE

12022 047632

EXIT: SUBTST &lt;&lt;HALT PROGRAM&gt;&gt;

```
*****
;SUBTEST      HALT PROGRAM
*****
```

12023 047632 004737 047664

```
EXIT2:  CALI.  SHUTUP
        IF APTFLAG IS TRUE OR ACTFLAG IS TRUE
```

12024 047636

12025 047652 000777

12026 047654

12027 047656 000000

12028 047660 000137 003654

12029 047664

12030

12031 047664

SHUTUP: SUBTST &lt;&lt;SHUTDOWN DIAGNOSTIC&gt;&gt;

```
*****
;SUBTEST      SHUTDOWN DIAGNOSTIC
*****
```

12032

12033

12034

12035

12036

12037

12041 047664 104472

12042 047666

12043 047700

12044 047706 004737 024566

12045 047712

12046 047712 012700 000001

12047 047716 013701 002562

12048 047722 004737 046410

12049 047726 104421

12050 047730 005737 002450

12051 047734 001003

12052 047736 042737 000040 172516

12056 047744 000207

12057

12058 047746

```
        ;INITIALIZE ALL CSR'S
        ;UNRELOCATE
        ;FLUSH OUT DBE'S
        ;RESTORE LOADERS
        ;TURN OFF MEMORY MANAGEMENT
        ;UNMAP THE UNIBUS MAP
        ECCINIT      ;TRAP ON DOUBLE BIT ERRORS (NORMAL)
        IF RLFLAG IS TRUE THEN $CALL UNRELOCATE
        IF QUICK IS FALSE
            CALL MT0030      ;FLUSH OUT DBE'S
        END ;OF IF QUICK
        MOV #1,R0      ;DESTINATION BANK
        MOV LOADHOME,R1 ;SOURCE BANK
        CALL BANKMOV
        DEENERGIZE      ;TURN OFF MEMORY MANAGEMENT
        TST NO22BIT      ;DOES THIS PDP-11 HAVE 22-BIT ADDR?
        BNE 1$          ;BRANCH IF NOT
        BIC #BIT5,MMR3    ;TURN OFF UNIBUS MAP
1$:      RETURN
```

APTDOWN: SUBTST &lt;&lt;APT SHUTDOWN SEQUENCE&gt;&gt;

```
*****
;SUBTEST      APT SHUTDOWN SEQUENCE
*****
```

12059 047746

12060 047762

12061 047770 012737 047746 060024

12062 047776 012737 000340 060026

12063 050004 012737 000000 127746

12064 050012 104417

12065 050014 000000

```
        MAP #0      ;MAP SUPERVISOR SPACE (TEST AREA) TO BANK #0
        TESTAREA    ;ENTER TEST MODE
        MOV #APTDOWN,FIRST+24
        MOV #340,FIRST+26
        MOV #0,FIRST+APTDOWN
        KERNEL      ;ENTER KERNEL MODE
APTHLT: HALT
```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 379  
APT SHUTDOWN SEQUENCE

12068 050016

```

SUBTST <<BLOCK MOVE SUBROUTINE>>
*****
*SUBTEST    BLOCK MOVE SUBROUTINE
*****
;BLOCK3 HAS 3 ARGUEMENTS
;BLOCK2 HAS 2 ARGUEMENTS
;BLOCK1 HAS 1 ARGUEMENTS
;
;ALL ARE CALLED BY THE BMOV MACRO
.ENABL    LSB
BLOCK1: PUSH    R0,R1,R2
MOV       #FASTCITY,R2
MOV       #16.,R1
BR        3$

BLOCK2: PUSH    R0,R1,R2
MOV       #16.,R1
BR        2$

BLOCK3: PUSH    R0,R1,R2
MOV       (R5)+,R1
2$: MOV       (R5)+,R2
3$: MOV       (R5)+,R0

1$: MOV       (R0)+,(R2)+
SOB       R1,1$
POP       R2,R1,R0
RTS       R5
.DSABL    LSB

```

12069  
12070  
12071  
12072  
12073  
12074  
12075 050016 012702 177640  
12076 050024 012701 000020  
12077 050030 000413  
12078 050034  
12079  
12080 050036  
12081 050044 012701 000020  
12082 050050 000404  
12083  
12084 050052  
12085 050060 012501  
12086 050062 012502  
12087 050064 012500  
12088  
12089 050066 012022  
12090 050070 077102  
12091 050072  
12092 050100 000205  
12093

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 380  
FIELD SERVICE MODE

12095  
12096  
12097 050102  
  
12098 050102 104415  
12099 050104  
12100  
12101 050110  
12102 050124  
12103 050130 104416  
12104 050132 000207  
12105 050134  
12106 050134 005737 002540  
12107 050140 001402  
12108 050142  
12109 050146  
12110 050156 104424  
12111 050160  
12112 050166  
12113 050172 104414  
12114 050174  
12115 050176 020027 000022  
12116 050202 101403  
12117 050204  
12118 050210 000766  
12119 050212  
12120 050222 050300  
12121 050224 050402  
12122 050226 050512  
12123 050230 050660  
12124 050232 051134  
12125 050234 051454  
12126 050236 052372  
12127 050240 052400  
12128 050242 052672  
12129 050244 053076  
12130 050246 053370  
12131 050250 053416  
12132 050252 053440  
12133 050254 053460  
12134 050256 053502  
12139 050260 053520  
12140 050262 053604  
12141 050264 053646  
12142 050266 053662  
12143 050270  
12144 050276 000733

```
.SBTTL FIELD SERVICE MODE

FIELDSERVICE:SUBTST <<SUBR FIELD SERVICE COMMAND MODE>>
:*****
:*SUBTEST SUBR FIELD SERVICE COMMAND MODE
:*****
: SAVREG
: TYPE MSG020 ;FIELD SERVICE COMMAND MODE

: IF RLFLAG IS TRUE OR NOFSMODE IS TRUE
: TYPE MSG048 ;NOT AVAILABLE NOW - TRY LATER!
: RESREG
: RETURN
: END ;OF IF RLFLAG
: TST CACHKN
: BEQ 1$
: PUSH CONTRL ;SAVE CACHE STATUS
: PUSH CSRNO,KAMIKAZE ;SAVE CSR & KAMIKAZE STATUS
: CACHOFF ;TURN CACHE OFF
: SET KAMIKAZE
: FS1: TYPE MSG026 ;COMMAND:
: RDDEC ;READ A DECIMAL NUMBER
: POP RC ;COMMAND --> R0
: CMP R0,#18.
: BLOS 1$
: TYPE MSG021
: BR FS1
: 1$: CASE R0
: FSCMD0 ;EXIT FIELD SERVICE COMMANDS
: FSCMD1 ;READ CSR
: FSCMD2 ;LOAD CSR
: FSCMD3 ;EXAMINE MEMORY
: FSCMD4 ;MODIFY MEMORY
: FSCMD5 ;SELECT BANK & PATTERN
: FSCMD6 ;TYPE CONFIGURATION MAP
: FSCMD7 ;SOB-A-LONG TEST
: FSCMD8 ;ERROR SUMMARY
: FSCMD9 ;REFRESH TEST
: FCMD10 ;SET FILL COUNT
: FCMD11 ;ENTER KAMIKAZE MODE
: FCMD12 ;EXIT KAMIKAZE MODE
: FCMD13 ;TURN CACHE OFF
: FCMD14 ;TURN CACHE ON
: FCMD15 ;TEST ONLY SELECTED BANKS
: FCMD16 ;RESUME TESTING ALL BANKS
: FCMD17 ;ENABLE TRACE
: FCMD18 ;DISABLE TRACE
: END ;OF CASE
: BR FS1
```



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 382  
 SUBR FIELD SERVICE COMMAND MODE

12147 050300

```
FSCMD0: SUBTST <<COMMAND 0 EXIT>>
:*****
:*SUBTEST COMMAND 0 EXIT
:*****
```

12148 050300  
 12149 050304 062706 000002  
 12150 050310  
 12151 050316 062706 000002  
 12152 050322 005037 002006  
 12153 050326  
 12154 050330  
 12155 050334  
 12156 050334  
 12157 050340 005737 002540  
 12158 050344 001414  
 12159 050346  
 12160 050356 062706 000002  
 12161 050362  
 12162 050364 005737 002540  
 12163 050370 001402  
 12164 050372  
 12165 050376  
 12166 050376 104416  
 12167 050400 000207  
 12168  
 12169 050402

```
TYPE MSG103 ;LEAVING FIELD SERVICE MODE
ADD #2,SP
IF SKIPKAMI IS TRUE
ADD #2,SP ;THROW AWAY OLD KAMIKAZE FLAG
CLR SKIPKAMI
ELSE
POP KAMIKAZE ;RESTORE OLD KAMIKAZE FLAG
END ;OF IF SKIPKAMI
POP CSRNO
TST CACHKN
BEQ RES0
IF CACHKN EQ CACHKF ;IF CACHE IS OFF
ADD #2,SP ;THROW AWAY CACHE STATUS
ELSE
TST CACHKN
BEQ RES0
POP CONTRL ;RESTORE CACHE STATUS
END ;OF IF CACHKN
RES0: RESREG
RETURN
```

```
FSCMD1: SUBTST <<FS COMMAND 1 READ CSR>>
:*****
:*SUBTEST FS COMMAND 1 READ CSR
:*****
```

12170 050402 004737 053674  
 12171 050406 010637 002302  
 12172 050412  
 12173 050420 104426  
 12174 050422  
 12175 050430 104026  
 12176 050432  
 12177 050454 000207  
 12178 050456  
 12179 050462 013706 002302  
 12180 050466  
 12181 050510 000207

```
CALL WHICHCSR
MOV SP,FSSTACK
SET4 #RES1 ;TRAPS TO 4 GOTO RES1
READCSR
SET NOERROR
ERROR +26 ;USE ERROR ROUTINE FOR PRINTOUT
RES4 ;RESET TRAPS TO 4 TO DEFAULT
RETURN
RES1: TYPE MSG025 ;THIS CSR DOES NOT EXIST
MOV FSSTACK,SP
RES4 ;RESET TRAPS TO 4 TO DEFAULT
RETURN
```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 384  
 FS COMMAND 1 READ CSR

12184 050512

```

FSCMD2: SUBTST <<FS      COMMAND 2      LOAD CSR>>
:*****
:*SUBTEST      FS      COMMAND 2      LOAD CSR
:*****
      CALL      WHICHCSR
      MOV       SP,FSSTACK
      SET4      #RES2          ;TRAPS TO 4 GOTO RES2
      READCSR
      TYPE      M^G027
      SET       NOERROR
      ERROR     +26          ;USE ERROR ROUTINE FOR PRINTOUT
      RES4
      TYPE      MSG023        ;RESET TRAPS TO 4 TO DEFAULT
      RDOCT
      POP       CSR          ;FIRST CSR WORD
      LOADCSR
      READCSR
      TYPE      MSG028        ;READ AN OCTAL NUMBER
      SET       NOERROR      ;PUT IN IN LOC "CSR"
      ERROR     +26          ;USE FOR PRINTOUT - NOT AN ERROR
      RETURN
RES2:  TYPE      MSG025        ;THIS CSR DOES NOT EXIST
      MOV       FSSTACK,SP   ;RESET TRAPS TO 4 TO DEFAULT
      RES4
      RETURN
  
```

12185 050512 004737 053674  
 12186 050516 010637 002302  
 12187 050522  
 12188 050530 104426  
 12189 050532  
 12190 050536  
 12191 050544 104026  
 12192 050546  
 12193 050570  
 12194 050574 104413  
 12195 050576  
 12196 050602 104425  
 12197 050604 104426  
 12198 050606  
 12199 050612  
 12200 050620 104026  
 12201 050622 000207  
 12202 050624  
 12203 050630 013706 002302  
 12204 050634  
 12205 050656 000207

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 386  
 FS COMMAND 2 LOAD CSR

12208 050660

```

FSCMD3: SUBTST <<FS COMMAND 3 EXAMINE MEMORY>>
:*****
:*SUBTEST FS COMMAND 3 EXAMINE MEMORY
:*****
12209 050660 PUSH BANK,NOPAR,PARTHERE,4
12210 050700 012737 000002 002074 MOV #2,NOPAR ;INDICATE PARITY ACTION
12211 050706 TYPE MSG029 ;EXAMINE MEMORY
12212 050712 1$: TYPE MSG031 ;PHYSICAL ADDRESS (0-17775776)??
12213 050716 104413 RDOCT ;READ OCTAL NUMBER ONTO STACK & $HIOCT
12214 050720 013737 065140 002100 MOV $HIOCT,BANK ;PUT MSB'S IN BANK
12215 050726 POP RO ;PUT LSB'S IN RO
12216 050730 000241 CLC
12217 050732 006100 ROL RO
12218 050734 006137 002100 ROL BANK
12219 050740 000241 CLC
12220 050742 006000 ROR RO
12221 050744 023737 002100 002552 CMP BANK, LASTBANK ;CHECK FOR BANK TOO HIGH
12222 050752 003357 BGT 1$ ;BRANCH IF TRUE
12223 050754 062700 060000 ADD #FIRST,RO
12224 050760 032700 000001 BIT #BIT0,RO ;CHECK FOR ODD ADDRESS
12225 050764 001352 BNE 1$ ;BRANCH IF ODD ADDRESS
12226 050766 020027 157776 CMP RO,#LAST ;CHECK FOR ADDRESS OVER 16K
12227 050772 101347 BHI 1$ ;BRANCH IF OVER 16K
12228 050774 012737 051046 002300 MOV #3$,PARTHERE ;INCASE OF ABORTS
12229 051002 SET4 #4$ ;TRAPS TO 4 GOTO 4$
12230 051010 MAP BANK ;MAP SUPERVISOR SPACE (TEST AREA) TO BANK
12231 051024 TESTAREA ;ENTER TEST MODE
12232 051032 011001 MOV (RO),R1
12233 051034 104417 KERNEL ;ENTER KERNEL MODE
12234 051036 TYPOCS R1
12235 051044 000410 BR EXCMD3
12236 3$: TYPE MSG032 ;PARITY ABORT
12237 051046 BR EXCMD3
12238 051052 000405
12239 4$: ADD #4,SP ;FIX STACK
12240 051054 062706 000004 TYPE MSG033 ;TIMEOUT TRAP
12241 051060 000400 BR EXCMD3
12242 051064 104417
12243 EXCMD3: KERNEL ;ENTER KERNEL MODE
12244 051066 104417 POP 4,PARTHERE,NOPAR,BANK
12245 051070 RES4 ;RESET TRAPS TO 4 TO DEFAULT
12246 051110 RETURN
12247 051132 000207

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 388  
 FS COMMAND 3 EXAMINE MEMORY

12250 051134

```

FSCMD4: SUBTST <<FS COMMAND 4 MODIFY MEMORY>>
:*****
:*SUBTEST FS COMMAND 4 MODIFY MEMORY
:*****
12251 051134 PUSH BANK,NOPAR,PARTHERE,4
12252 051154 012737 000003 002074 MOV #3,NOPAR ;INDICATE PARITY ACTION
12253 051162 TYPE MSG036 ;MODIFY MEMORY
12254 051166 1$: TYPE MSG031 ;PHYSICAL ADDRESS (0-17775776)??
12255 051172 104413 RDOCT ;READ OCTAL NUMBER ONTO STACK & $HIOCT
12256 051174 013737 065140 002100 MOV $HIOCT,BANK ;PUT MSB'S IN BANK
12257 051202 POP RO ;PUT LSB'S IN RO
12258 051204 000241 CLC
12259 051206 006100 ROL RO
12260 051210 006137 002100 ROL BANK
12261 051214 000241 CLC
12262 051216 006000 ROR RO
12263 051220 IF BANK GT LASTBANK THEN GOTO 1$ ;CHECK FOR BANK TOO HIGH
12264 051230 062700 060000 ADD #FIRST,RO
12265 051234 IF #BIT0 SET.IN RO THEN GOTO 1$ ;CHECK FOR ODD ADDRESS
12266 051242 IF RO HI #LAST THEN GOTO 1$ ;CHECK FOR ADDRESS OVER 16K
12267 051250 012737 051316 002300 MOV #3$,PARTHERE ;INCASE OF ABORTS
12268 051256 SET4 #4$ ;TRAPS TO 4 GOTO 4$
12269 051264 MAP BANK ;MAP SUPERVISOR SPACE (TEST AREA) TO BANK
12270 051300 104511 INVALIDATE
12271 051302 TESTAREA ;ENTER TEST MODE
12272 051310 011001 MOV (RO),R1
12273 ;GETTING HERE MEANS WE GOT LUCKY - NO TRAPS
12274 051312 104417 KERNEL ;ENTER KERNEL MODE
12275 051314 000410 BR 5$
12276
12277 051316 3$: TYPE MSG032 ;PARITY ABORT
12278 051322 000431 BR EXCMD4 ;EXIT
12279
12280 051324 062706 000004 4$: ADD #4,SP ;FIX STACK
12281 051330 TYPE MSG033 ;TIMEOUT TRAP
12282 051334 000424 BR EXCMD4 ;EXIT
12283
12284 051336 5$: TYPE MSG037 ;OLD DATA WAS
12285 051342 TYPOCS R1 ;PRINT IT
12286 051350 TYPE MSG039 ;INPUT NEW DATA
12287 051354 104413 RDOCT ;READ ON OCTAL NUMBER ONTO THE STACK
12288 051356 POP R1 ;GET NEW NUMBER
12289 051360 TESTAREA ;ENTER TEST MODE
12290 051366 010110 MOV R1,(RO) ;PUT IT IN MEMORY
12291 051370 011001 MOV (RO),R1 ;READ IT AGAIN
12292 051372 104417 KERNEL ;ENTER KERNEL MODE
12293 051374 TYPE MSG038 ;DATA IS NOW
12294 051400 TYPOCS R1 ;PRINT IT
12295
12296 051406 104417 EXCMD4: KERNEL ;ENTER KERNEL MODE
12297 051410 POP 4,PARTHERE,NOPAR,BANK
12298 051430 RES4 ;RESET TRAPS TO 4 TO DEFAULT
12299 051452 000207 RETURN

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 390  
 FS COMMAND 4 MODIFY MEMORY

```

12302 051454      FSCMD5: SUBTST <<FS      COMMAND 5      SELECT BANK & PATTERN>>
:*****
:*SUBTEST      FS      COMMAND 5      SELECT BANK & PATTERN
:*****
12303 051454      PUSH      BANK,PATTERN,TESTADD,PCBUMP,TKVEC,TKVEC+2
12304 051504      MOV      SP,FSSTACK      ;SAVE LAST GOOD STACK POINTER
12305 051510      TYPE      MSG040      ;SELECT BANK & PATTERN TEST
12306 051514      1$:      TYPE      MSG030      ;BANK(0-177)?
12307 051520      RDOCT      ;READ AN OCTAL NUMBER ONTO THE STACK
12308 051522      POP      BANK      ;PUT IT IN BANK
12309 051526      IF BANK GT LASTBANK THEN GOTO 1$ ;CHECK FOR BANK TOO HIGH
12310
12311 051536      MOV      BANK,R1
12312 051542      ASL      R1
12313 051544      ASL      R1
12314 051546      IF CPUBIT OFF.IN CONFIG(R1)
12315 051556      TYPE      MSG041      ;BANK NOT ACCESSABLE
12316 051562      GOTO 1$
12317 051564      END ;OF IF
12318
12319 051564      2$:      TYPE      MSG042      ;PATTERN(0-45)?
12320 051570      RDOCT      ;READ AN OCTAL NUMBER ONTO THE STACK
12321 051572      POP      PATTERN      ;PUT IT IN PATTERN
12322 051576      IF PATTERN GT #47 THEN GOTO 2$ ;CHECK FOR PATTERN TO HIGH
12323 051606      IF PATTERN EQ #0
12324 051614      TYPE      MSG043      ;PATTERN 0 DATA IS?
12325 051620      RDOCT      ;READ AN OCTAL NUMBER ONTO THE STACK
12326 051622      POP      R2      ;PUT IT IN R2
12327 051624      END ;OF IF
12328
12329
12330 051624      MAP      BANK      ;MAP SUPERVISOR SPACE (TEST AREA) TO BANK
12331 051640      INVALDATE
12332 051642      CALL      EXBANK      ;SET NEW MARGINS
12333 051646      IF RRFLAG IS TRUE
12334 051654      TYPE      MSG049      ;BANK REQUIRES RELOCATION
12335 051660      JMP CMD5C
12336 051664      END ;OF IF RRFLAG
12337 051664      TYPE      MSG046      ;TO ESCAPE TYPE ANY KEY!
12338 051670      MOV      CSRNO,SAVCSR      ;SAVE OLD CSR NUMBER
12339 051676      MOV      BANK,R2
12340 051702      ASH      #2,R2      ;GENERATE INDEX INTO CONFIGURATION TABLE
12341 051706      MOV      CONFIG(R2),R3      ;R3 = LOW WORD OF CONFIGURATION TABLE FOR THIS BANK
12342 051712      ASH      #-10,R3      ;POSITION CSR CODE IN BITS 0-3
12343 051716      BIC      #^C17,R3      ;CLEAR ALL BUT THE CSR CODE
12344 051722      ASL      R3      ;ADJUST CSR NUMBER
12345 051724      MOV      R3,CSRNO
12346 051730      MOV      #CMD5C,TKVEC
12347 051736      MOV      #340,TKVEC+2
12348 051744      MOV      @STKB,R0
12349 051750      BIC      #BIT7,PSW
12350 051756      BIS      #BIT6,@STKS      ;KILL ANY OLD INTERRUPT
:                                     ;LOWER CPU PRIORITY TO 140
:                                     ;ENABLE KEYBOARD INTERRUPTS
12351
12352
12353 051764      CMD5B: SET      HEADER,MUT
12354 052000      MOV      BANK,R1
12355 052004      ASL      R1

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 390-1  
FS COMMAND 5 SELECT BANK & PATTERN

12356	052006	006301		ASL	R1	
12357	052010	005037	002236	CLR	SPLTCSR	
12358	052014	005037	002262	CLR	PASFLG	
12359	052020	012737	060000	MOV	#FIRST,TESTADD	
12360	052026	012737	060002	MOV	#FIRST+2,TESTADD+2	
12361	052034			IF #BIT12	SET.IN CONFIG+2(R1)	
12362	052044	005237	002236	INC	SPLTCSR	
12363	052050			MAP	BANK	
12364	052064	012737	120000	MOV	#120000,TESTADD+2	
12365	052072			END: OF	IF #BIT12	
12366	052072			IF #SW0	SET.IN @SWR	
12367	052102	104470		ECCDIS		;DISABLE ERROR CORRECTION
12368	052104			ELSE		
12369	052106			PUSH	CSRNO	
12370	052112	104502		CLRCSR		;CLEAR CSRS
12371	052114			POP	CSRNO	
12372	052120			END: OF	IF	
12373	052120	012737	000002	MOV	#2,NOPAR	;PARITY ACTION
12374	052126	012737	000002	MOV	#2,PCBUMP	;TRAPS ADD 2 TO PC
12375	052134	013700	002110	MOV	PATTERN,R0	
12376	052140	006300		ASL	R0	
12377	052142	004770	052154	CALL	@FSPAT(R0)	
12378	052146	005037	002074	CLR	NOPAR	
12379	052152	000712		BR	CMD5B	;LOOP TILL KEYBOARD INTERRUPT
12380						
12381	052154	020700		FSPAT:	MT0000	<1 SEC DATA PATTERN TEST
12382	052156	020760			MT0001	<1 SEC ADDRESS TEST
12383	052160	021100			MT0002	<1 SEC COMPLEMENT ADDRESS TEST
12384	052162	021240			MT0003	1 SEC 3 XOR 9 WORST CASE NOISE TEST
12385	052164	021472			MT0004	1 SEC ROTATING ZEROS TEST
12386	052166	021614			MT0005	1 SEC ROTATING ONES TEST
12387	052170	021750			MT0006	<1 SEC INITIAL DATA TEST
12388	052172	022004			MT0007	<1 SEC ADDRESS BIT TEST
12389	052174	022046			MT0010	<1 SEC BYTE ADDRESSING TEST
12390	052176	022102			MT0011	<2 SEC CREATE SINGLE BIT ERROR TEST
12391	052200	022160			MT0012	<1 SEC WRITE BYTE CLEARS SBE TEST
12392	052202	022264			MT0013	1 SEC CREATE DOUBLE BIT ERROR TEST
12393	052204	022350			MT0014	1 SEC BASIC DOUBLE BIT ERROR TEST
12394	052206	022440			MT0015	1 SEC WRITE INHIBIT OF BYTE WITH DBE
12395	052210	022516			MT0016	<1 SEC WRITE INHIBIT OF WORD WITH DBE
12396	052212	022574			MT0017	<1 SEC HOLDING 1'S & 0'S TEST
12397	052214	022616			MT0020	1 SEC SYNDROMES TO CSR ON SBE TEST
12398	052216	022706			MT0021	1 SEC MARCHING 0'S & 1'S TEST
12399	052220	023160			MT0022	10 SEC REFRESH & SHIFTING DIAGONAL TEST
12400	052222	023212			MT0023	10 SEC SHIFTING DIAGONAL TEST
12401	052224	023256			MT0024	20 SEC FAST GALLOPING PATTERN TEST
12402	052226	023522			MT0025	<1 SEC INTERRUPT ENABLE TEST
12403	052230	023600			MT0026	<1 SEC RANDOM DATA TEST
12404	052232	024102			MT0027	1 SEC UNIQUE BANK TEST
12405	052234	024566			MT0030	1 SEC FLUSH OUT DBE'S TEST
12406	052236	025070			MT0031	3 SEC SOB-A-LONG TEST
12407	052240	025260			MT0032	<1 SEC WRITE RECOVERY TEST
12408	052242	025612			MT0033	35 SEC BRANCH GOBBLE TEST
12409	052244	026000			MT0034	1 SEC SOFT ERROR TEST
12410	052246	026152			MT0035	<1 SEC WORST CASE NOISE PARITY TEST
12411	052250	026264			MT0036	1 SEC CORRECTION CODE TEST
12412	052252	026336			MT0037	<1 SEC ECC DISABLE TEST

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 390-2

FS COMMAND 5 SELECT BANK &amp; PATTERN

12413	052254	026404			MT0040	: 1 SEC	NO WRITE ABORT WITH ECC DISABLED TEST
12414	052256	026406			MT0041	: 1 SEC	ADDRESS TO CSR ON DOUBLE BIT ERROR TEST
12415	052260	026450			MT0042	: <1 SEC	EXTENDED UNIBUS ADDRESS TEST
12416	052262	026514			MT0043	: 1 SEC	WRITE BYTE CLEARS SBE TEST
12417	052264	026554			MT0044	: 1 SEC	SHIFTING 1/0'S THROUGH THE CHECK BITS
12418	052266	026614			MT0045	: 1 SEC	SYNDROMES TO CSR ON DBE TEST
12419	052270	026654			MT0046	: 1 SEC	CHECK SINGLE BIT ERROR WITH ECC DISABLED TEST
12420	052272	026714			MT0047	: <1 SEC	NO CSR UPDATE WITH SBE ON DBE TEST
12421							
12422	052274	013706	002302		CMD5C: MOV	FSSTACK, SP	;RECOVER OLD STACK POINTER
12423	052300	042777	000100	130320	BIC	#BIT6, @STKS	
12424	052306				POP	TKVEC+2, TKVEC	
12425	052316	117700	130306		MOVB	@STKB, R0	;GET CHARACTER TO GET RID OF FLAG
12426	052322				POP	PCBUMP, TESTADD	
12427	052332				POP	PATTERN, BANK	
12428	052342				MAP	BANK	;REMAP OLD BANK
12429	052356	004737	047020		CALL	EXBANK	
12430	052362	013737	002152	002150	MOV	SAVCSR, CSRNO	;RESTORE CSRNO.
12431	052370	000207			RETURN		

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 391  
FS COMMAND 5 SELECT BANK & PATTERN

12433 052372

FSCMD6: SUBTST <<FS COMMAND 6 TYPE CONFIGURATION MAP>>  
:\*\*\*\*\*  
:\*SUBTEST FS COMMAND 6 TYPE CONFIGURATION MAP  
:\*\*\*\*\*  
CALL PCONFIG  
RETURN

12434 052372 004737 041352  
12435 052376 000207  
12436



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 393  
 FS COMMAND 6 TYPE CONFIGURATION MAP

```

12439 052400      FSCMD7: SUBTST <<FS      COMMAND 7      SOB-A-LONG TEST>>
:*****
:*SUBTEST      FS      COMMAND 7      SOB-A-LONG TEST
:*****
12440 052400      PUSH      BANK,PATTERN,TKVEC,TKVEC+2,NOPAR
12441 052424      MOV      SP,FSSTACK      ;SAVE LAST GOOD STACK POINTER
12442 052430      TYPE      MSG055      ;SOB-A-LONG TEST
12443
12444 052434      IF #SWO SET.IN @SWR
12445 052444      ECCDIS      ;DISABLE ERROR CORRECTION
12446 052446      ELSE
12447 052450      CLRCR      ;CLEAR CSRS
12448 052452      END :OF IF
12449 052452      TYPE      MSG056      ;BELL = EACH PASS COMPLETE
12450
12451 052456      TYPE      MSG046      ;TO ESCAPE TYPE ANY KEY!
12452 052462      MOV      #CMD7C,TKVEC
12453 052470      MOV      #340,TKVEC+2
12454 052476      MOV      @STKB,R0      ;KILL ANY OLD INTERRUPT
12455 052502      BIC      #BIT7,PSW      ;LOWER CPU PRIORITY TO 140
12456 052510      BIS      #BIT6,@STKS      ;ENABLE KEYBOARD INTERRUPTS
12457
12458
12459 052516      SET      HEADER,MUT
12460
12461 052532      CMD7B: FOR BANK := #0 TO LASTBANK
12462 052536      CALL EXBANK
12463 052542      IF ACFLAG IS TRUE AND RRFLAG IS FALSE
12464 052556      INVALIDATE
12465 052560      CALL MT0031
12466 052564      END :OF IF ACFLAG
12467 052564      END :OF FOR BANK
12468 052600      TYPE      $BELL      ;RING BELL
12469 052604      GOTO      CMD7B
12470
12471 052606      013706 002302      CMD7C: MOV      FSSTACK,SP      ;RECOVER OLD STACK POINTER
12472 052612      042777 000100 130006 BIC      #BIT6,@STKS
12473 052620      117700 130004      MOV      @STKB,R0      ;READ CHAR TO KILL FLAG
12474 052624      POP      NOPAR,TKVEC+2,TKVEC,PATTERN,BANK
12475 052650      MAP      BANK      ;MAP SUPERVISOR SPACE (TEST AREA) TO BANK
12476 052664      004737 047020      CALL      EXBANK
12477 052670      000207      RETURN
  
```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 395  
 FS COMMAND 7 SOB-A-LONG TEST

12480 052672

```

FSCMD8: SUBTST <<FS COMMAND 8 ERROR SUMMARY>>
:*****
:*SUBTEST FS COMMAND 8 ERROR SUMMARY
:*****
PUSH R0,R2,R3,BANK
MOV $PASS,TEMP
DEC TEMP
TYPDEC TEMP
TYPE MSG125 ;PASSES COMPLETED
TYPDEC $ERTTL
TYPE MSG079 ;ERROR(S) DETECTED
IF $ERTTL NE #0
CLR SUCCESS
FOR BANK := #0 TO LASTBANK
MOV BANK,R3
MUL #4,R3
IFB CONFIG+2(R3) NE #0
IF SUCCESS IS FALSE
TYPE MSG076 ;BANK ERRORS
SET SUCCESS
END :OF IF SUCCESS
TYPDCS BANK,3
MOVB CONFIG+2(R3),R0
BIC #^C377,R0
TYPDEC R0
TYPE $CRLF
END :OF IFB CONFIG(R3)
END :OF FOR BANK
END :OF IF $ERTTL
POP BANK,R3,R2,R0
RETURN
  
```

12481 052672  
 12482 052704 013737 065646 002430  
 12483 052712 005337 002430  
 12484 052716  
 12485 052724  
 12486 052730  
 12487 052736  
 12488 052742  
 12489 052750 005037 002330  
 12490 052754  
 12491 052760 013703 002100  
 12492 052764 070327 000004  
 12493 052770  
 12494 052776  
 12495 053004  
 12496 053010  
 12497 053016  
 12498 053016  
 12499 053026 116300 002652  
 12500 053032 042700 177400  
 12501 053036  
 12502 053042  
 12503 053046  
 12504 053046  
 12505 053062  
 12506 053062  
 12507 053074 000207

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 397  
 FS COMMAND 8 ERROR SUMMARY

12510 053076

FSCMD9: SUBTST <<FS COMMAND 9 REFRESH TEST>>  
 :\*\*\*\*\*  
 :\*SUBTEST FS COMMAND 9 REFRESH TEST  
 :\*\*\*\*\*

12511 053076

12512 053122 010637 002302

12513 053126

12514

12515 053132

12516 053142 104470

12517 053144

12518 053146 104502

12519 053150

12520 053150

12521

12522 053154

12523 053160 012737 053304 000060

12524 053166 012737 000340 000062

12525 053174 017700 127430

12526 053200 042737 000200 177776

12527 053206 052777 000100 127412

12528

12529 053214

12530

12531 053230

12532 053234 004737 047020

12533 053240

12534 053254 104511

12535 053256 004737 023160

12536 053262

12537 053262

12538 053276

12539 053302

12540

12541 053304 013706 002302

12542 053310 042777 000100 127310

12543 053316 117700 127306

12544 053322

12545 053346

12546 053362 004737 047020

12547 053366 000207

12548

PUSH BANK,PATTERN,TKVEC,TKVEC+2,NOPAR  
 MOV SP,FSSTACK ;SAVE LAST GOOD STACK POINTER  
 TYPE MSG073 ;REFRESH TEST

IF #SWO SET.IN @SWR  
 ECCDIS ;DISABLE ERROR CORRECTION

ELSE  
 CLRCSR ;CLEAR CSRS

END :OF IF  
 TYPE MSG056 ;BELL = EACH PASS COMPLETE

TYPE MSG046 ;TO ESCAPE TYPE ANY KEY!

MOV #CMD9C,TKVEC

MOV #340,TKVEC+2

MOV @STKB,R0

BIC #BIT7,PSW ;KILL ANY OLD INTERRUPT

BIS #BIT6,@STKS ;LOWER CPU PRIORITY TO 140

SET HEADER,MUT ;ENABLE KEYBOARD INTERRUPTS

CMD9B: FOR BANK := #0 TO LASTBANK  
 CALL EXBANK  
 IF ACFLAG IS TRUE AND RRFLAG IS FALSE  
 INVALIDATE  
 CALL MT0022  
 END :OF IF ACFLAG  
 END :OF FOR BANK  
 TYPE \$BELL ;RING BELL  
 GOTO CMD9B

CMD9C: MOV FSSTACK,SP ;RECOVER OLD STACK POINTER  
 BIC #BIT6,@STKS  
 MOV @STKB,R0 ;PEAD CHAR TO KILL FLAG  
 POP NOPAR,TKVEC+2,TKVEC,PATTERN,BANK  
 MAP BANK ;MAP SUPERVISOR SPACE (TEST AREA) TO BANK  
 CALL EXBANK  
 RETURN

CZMSPA0 1S11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 399  
 FS COMMAND 9 REFRESH TEST

12551 053370

```
FCMD10: SUBTST <<FS COMMAND 10 SET FILL COUNT>>
:*****
:*SUBTEST FS COMMAND 10 SET FILL COUNT
:*****
```

12552 053370

12553 053372

12554 053376 104413

12555 053400

12556 053402 042700 177760

12557 053406 110037 002353

12558 053412

12559 053414 000207

12560

12561 053416

```

PUSH RO
TYPE MSG085 ;FILL COUNT(OCTAL)?
RDOCT
POP RO
BIC #^C17,RO
MOVB RO,$FILLS
POP RO
RETURN
```

```
FCMD11: SUBTST <<FS COMMAND 11 ENTER KAMIKAZE MODE>>
:*****
:*SUBTEST FS COMMAND 11 ENTER KAMIKAZE MODE
:*****
TYPE MSG101 ;ENTERING KAMIKAZE MODE
SET KAMIKAZE,SKIPKAMI
RETURN
```

12562 053416

12563 053422

12564 053436 000207

12565

12566 053440

```
FCMD12: SUBTST <<FS COMMAND 12 EXIT KAMIKAZE MODE>>
:*****
:*SUBTEST FS COMMAND 12 EXIT KAMIKAZE MODE
:*****
TYPE MSG102 ;LEAVING KAMIKAZE MODE
CLR KAMIKAZE
SET SKIPKAMI
RETURN
```

12567 053440

12568 053444 005037 002004

12569 053450

12570 053456 000207

12571

12572 053460

```
FCMD13: SUBTST <<FS COMMAND 13 TURN CACHE OFF>>
:*****
:*SUBTEST FS COMMAND 13 TURN CACHE OFF
:*****
TYPE MSG106 ;CACHE IS OFF
CACHOFF ;TURN CACHE OFF
MOV CACHKN,CACHKN+2 ;SAVE OLD CACHE ON STATE
CLR CACHKN ;KEEP CACHE OFF
RETURN
```

12573 053460

12574 053464 104424

12575 053466 013737 002540 002542

12576 053474 005037 002540

12577 053500 000207

12578

12579 053502

```
FCMD14: SUBTST <<FS COMMAND 14 TURN CACHE ON>>
:*****
:*SUBTEST FS COMMAND 14 TURN CACHE ON
:*****
TYPE MSG107 ;CACHE IS ON (EXCEPT DURING ACTUAL PATTERNS)
MOV CACHKN+2,CACHKN ;RESTORE OLD CACHE ON STATE
CACHON ;TURN CACHE ON
RETURN
```

12580 053502

12581 053506 013737 002542 002540

12582 053514 104423

12583 053516 000207

12584

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 400  
 FS COMMAND 14 TURN CACHE ON

12597  
 12598 053520

```
FCMD15: SUBTST <<FS COMMAND 15 TEST ONLY SELECTED BANKS>>
:*****
:*SUBTEST FS COMMAND 15 TEST ONLY SELECTED BANKS
:*****
TYPE MSG105 ;ENTER BANKS IN OCTAL - USE NUMBER OUTSIDE RANGE TO TERMINAT
CALL CMD16A ;ERASE OLD SELECTIONS
BEGIN CMD16LOOP
REPEAT
TYPE MSG030 ;BANK(0-177)?
RDOCT ;READ AN OCTAL NUMBER ONTO THE STACK
POP R1 ;PUT IT IN R1
IF R1 GT #177 OR R1 LT #0
LEAVE CMD16LOOP
END ;OF IF R1
ASL R1
ASL R1 ;R1 <- R1 * 4
BIS #BIT14,CONFIG+2(R1)
END ;OF REPEAT
END CMD16LOOP
TYPE MSG110 ;ONLY SELECTED BANKS WILL BE TESTED
SET SELONLY
RETURN
```

12599 053520  
 12600 053524 004737 053614  
 12601 053530  
 12602 053530  
 12603 053530  
 12604 053534 104413  
 12605 053536  
 12606 053540  
 12607 053552  
 12608 053554  
 12609 053554 006301  
 12610 053556 006301  
 12611 053560 052761 040000 002652  
 12612 053566  
 12613 053570  
 12614 053570  
 12615 053574  
 12616 053602 000207  
 12617  
 12618 053604

```
FCMD16: SUBTST <<FS COMMAND 16 RESUME TESTING ALL BANKS>>
:*****
:*SUBTEST FS COMMAND 16 RESUME TESTING ALL BANKS
:*****
TYPE MSG111 ;ALL BANKS WILL BE TESTED
CLR SELONLY
```

12619 053604  
 12620 053610 005037 002000  
 12621  
 12622  
 12623 053614 013702 002552  
 12624 053620 006302  
 12625 053622 006302  
 12626 053624  
 12627 053626 042761 040000 002652  
 12628 053634  
 12629 053644 000207

```
;ENTRY POINT FROM CMD15
CMD16A: MOV LASTBANK,R2
ASL R2
ASL R2
FOR R1 := #0 TO R2 BY #4
BIC #BIT14,CONFIG+2(R1)
END ;OF FOR R1
RETURN
```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 402  
FS COMMAND 16 RESUME TESTING ALL BANKS

12632 053646

FCMD17: SUBTST <<FS COMMAND 17 ENABLE TRACE>>  
:\*\*\*\*\*  
:\*SUBTEST FS COMMAND 17 ENABLE TRACE  
:\*\*\*\*\*

12633 053646

12634 053652 012737 177777 006204  
12635 053660 000207

TYPE MSG127  
MOV #-1,TRACE  
RETURN

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 404  
FS COMMAND 17 ENABLE TRACE

12638 053662

FCMD18: SUBTST <<FS COMMAND 18 DISABLE TRACE>>  
:\*\*\*\*\*  
:\*SUBTEST FS COMMAND 18 DISABLE TRACE  
:\*\*\*\*\*

12639 053662  
12640 053666 005037 006204  
12641 053672 000207

TYPE MSG128  
CLR TRACE  
RETURN

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 406  
 FS COMMAND 18 DISABLE TRACE

12644 053674

WHICHCSR:SUBTST &lt;&lt;SUBR DETERMINE CORRECT CSR&gt;&gt;

\*\*\*\*\*  
 :SUBTEST SUBR DETERMINE CORRECT CSR  
 \*\*\*\*\*

12645 053674 013700 002222  
 12646 053700 022700 100000  
 12647 053704 001003  
 12648 053706 005037 002150  
 12649 053712 000207  
 12650  
 12651 053714  
 12652 053720 104412  
 12653 053722  
 12654 053724 011000  
 12655 053726 020027 000106  
 12656 053732 101370  
 12657 053734 022700 000101  
 12658 053740 103002  
 12659 053742 162700 000007  
 12660 053746 162700 000060  
 12661 053752 006300  
 12662 053754 010037 002150  
 12663 053760 000207

MOV TOTCSRS,R0 ;GET CSR'S FLAG  
 CMP #BIT15,R0 ;CSR 0?  
 BNE 1\$ ;NO - SKIP  
 CLR CSRNO ;YES - SET IT UP  
 RETURN  
 1\$: TYPE MSG022 ;WHICH CSR(0-F)  
 RDLIN ;GET CHARACTER  
 POP R0 ;PUT IN R0  
 MOV (R0),R0 ;PUT CHAR IN R0  
 CMP R0,#106 ;CHECK LIMIT  
 BHI 1\$ ;IF BAD LOOP TILL HE TYPES IT RIGHT  
 CMP #'A,R0  
 BHIS 2\$  
 SUB #7,R0  
 2\$: SUB #60,R0  
 ASL R0  
 MOV R0,CSRNO  
 RETURN



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 421  
 ERROR DATA (SUPERVISOR) SETUP STUFF

13212  
 13213 053762  
 13214 053774  
 13215 054002  
 13216 054010  
 13217 054016 104417  
 13218 054020  
 13219 054020  
 13220 054026  
 13221 054032  
 13222 054034  
 13223 054040  
 13224 054040 000137 057242  
 13225  
 13226 054044  
  
 13227 054044  
 13228 054056  
 13229 054060 005037 002146  
 13230 054064 104505  
 13231 054066  
 13232 054074 005711  
 13233 054076 104417  
 13234 054100 104426  
 13235 054102 013700 002146  
 13236 054106 104503  
 13237 054110 072027 177773  
 13238 054114 042700 177600  
 13239 054120  
 13240 054124 005037 002042  
 13241 054130  
 13242 054136 011137 002050  
 13243 054142 011437 002052  
 13244 054146 104417  
 13245 054150 110037 002054  
 13246 054154 105037 002055  
 13247 054160 004737 057476  
 13248 054164 104033  
 13249 054166  
 13250 054170  
 13251 054200 104506  
 13252 054202  
 13253 054204 104472  
 13254 054206  
 13255 054206 000002

.SBTTL ERROR DATA (SUPERVISOR) SETUP STUFF  
 \$PER25: LET ADDRESS := R1 - #2  
 IF ABORTFLAG IS FALSE  
 TESTAREA ;ENTER TEST MODE  
 LET BAD := -2(R1)  
 KERNEL ;ENTER KERNEL MODE  
 END ;OF IF ABORTFLAG  
 IF 177654 EQ #0  
 LET GOOD := R2  
 ELSE  
 LET GOOD := R3  
 END ;OF IF  
 JMP PERRAW

PERRA3: SUBTST <<DATA WAS 3 WORDS>>

\*\*\*\*\*  
 :\*SUBTEST DATA WAS 3 WORDS  
 \*\*\*\*\*  
 IF BADPC EQ #0 THEN SCALL BADSTACK  
 PUSH R0  
 CLR CSR ;MAKE SURE CSR BIT HOLDER IS CLEAR  
 CHK1DIS ;DISABLE ECC & WRITE CHECKBITS FROM 1 SELECTED CSR  
 TESTAREA  
 TST (R1) ;READ LOCATION TO READ CHECKBITS INTO CSR  
 KERNEL  
 READCSR ;GET CSR CONTENTS  
 MOV CSR,R0 ;SAVE CSR CONTENTS IN R0  
 CLR1CSR ;RETURN CSR TO NORMAL MODE  
 ASH #-5,R0 ;MOVE CHECK BITS TO BOTTOM OF WORD  
 BIC #^C177,R0 ;CLEAR OFF EXTRANEIOUS GARBAGE  
 LET ADDRESS := R1 ;SAVE VIRTUAL ADDRESS FOR PRINTOUT  
 CLR GOOD ;FIRST TEST WORD WRITTEN SHOULD ALWAYS BE ZERO  
 TESTAREA ;ENTER TEST MODE  
 MOV (R1),BAD ;GET BAD DATA FROM MUT - FIRST WORD  
 MOV (R4),BAD2 ;AND SECOND WORD  
 KERNEL ;ENTER KERNEL MODE  
 MOVB R0,BAD3 ;MOVE BAD CHECKBITS FOR PRINTOUT  
 CLRB BAD3+1 ;CLEAR OFF THE OTHER UNUSED BITS  
 CALL PERBNK ;MARK BANK AS BAD IN CONFIG TABLE  
 ERROR +33  
 POP R0 ;PESTORE R0  
 IF #SW0 SET.IN @SWR  
 ENASBE ;TRAP ON SINGLE BIT ERRORS  
 ELSE  
 ECCINIT ;TRAP ON UNCORRECTABLE ERRORS  
 END; OF IF #SW0  
 RTI

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 423  
DATA WAS 3 WORDS

13258 054210  
13259 054214  
13260 054226  
13261 054234  
13262 054242  
13263 054250 104417  
13264 054252  
13265 054252 000137 057242  
13266  
13267 054256

\$PER30: LET GOOD := R1  
LET ADDRESS := (SP) - 16  
IF ABORTFLAG IS FALSE  
TESTAREA ;ENTER TEST MODE  
LET BAD := @ADDRESS  
KERNEL ;ENTER KERNEL MODE  
END :OF IF ABORTFLAG  
JMP PERRAW

GETDATA:SUBTST <<GET DATA FROM ABORTED AREA IF POSSIBLE>>  
:\*\*\*\*\*  
:\*SUBTEST GET DATA FROM ABORTED AREA IF POSSIBLE  
:\*\*\*\*\*

13268 054256  
13269 054270 010637 054354  
13270 054274 012737 054334 000004  
13271 054302 012737 054334 000114  
13272 054310 013700 002032  
13273 054314  
13274 054322 011037 002050  
13275 054326 104417  
13276 054330 005037 002142  
13277 054334 013706 054354  
13278 054340  
13279 054352 000207  
13280 054354 000000

PUSH R0,4,114  
MOV SP,GETDA1  
MOV #1\$,4  
MOV #1\$,114  
MOV ADDRESS,R0  
TESTAREA  
MOV (R0),BAD  
KERNEL  
CLR ABORTFLAG  
1\$: MOV GETDA1,SP ;RESTORE KNOWN GOOD STACK POINTER  
POP 114,4,R0  
RETURN  
GETDA1: 0

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 425  
POWER FAIL AUTO RESTART

13283						.SBTTL POWER FAIL AUTO RESTART
13284						.SBTTL ROUTINE POWER DOWN AND UP
13285						*****
13286						:POWER DOWN ROUTINE
13287	054356					\$PWRDN:
13295						:SAVE CACHE STATUS
13296	054356	005737	002540			TST CACHKN
13297	054362	001403				BEQ 5\$
13298	054364					PUSH CONTRL
13299	054370	104423				CACHON ;TURN CACHE ON
13300	054372	012737	055330	000024	5\$:	MOV #5ILLUP,PWRVEC ;:SET FOR FAST UP
13301	054400	012737	000340	000026		MOV #340,PWRVEC+2 ;:PRIO:7
13302	054406					PUSH R0,R1,R2,R3,R4,R5,CSRNO
13303						:SAVE USER PAR'S & PDR7
13304	054426	012700	177700			MOV #177700,R0
13305	054432	012701	000021			MOV #17,R1
13306	054436				1\$:	PUSH -(R0)
13307	054440	077102				SOB R1,1\$
13308						:SAVE SUPERVISOR PAR'S
13309	054442	005737	002452			TST NOSUPER
13310	054446	001013				BNE PD1
13311	054450	012700	172300			MOV #172300,R0
13312	054454	012701	000020			MOV #16,R1
13313	054460				2\$:	PUSH -(R0)
13314	054462	077102				SOB R1,2\$
13315	054464					IF RLFLAG IS TRUE THEN \$CALL WOOPS
13316						:COPY KERNEL MAP TO USER & SUPERVISOR
13317	054476	012700	172300		PD1:	MOV #KIPDR0,R0
13318	054502	012701	177600			MOV #UIPDRO,R1
13319	054506	012702	172200			MOV #SIPDR0,R2
13320	054512	012703	000040			MOV #32,R3
13321	054516	011021			3\$:	MOV (R0),(R1)+
13322	054520	012022				MOV (R0)+,(R2)+
13323	054522	077303				SOB R3,3\$

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 426  
ROUTINE POWER DOWN AND UP

13325						;SAVE USER & SUPERVISOR STACK POINTERS
13326	054524					USER
13327	054532	010600				MOV USP,R0
13328	054534	104417				KERNEL ;ENTER KERNEL MODE
13329	054536					PUSH R0
13330	054540	005737	002452			TST NOSUPER
13331	054544	001006				BNE 7\$
13332	054546					SUPERVISOR ;ENTER SUPERVISOR MODE
13333	054554	010600				MOV SSP,R0
13334	054556	104417				KERNEL ;ENTER KERNEL MODE
13335	054560					PUSH R0
13336						;SAVE ECC REGISTERS
13337	054562	013701	002222	7\$:		MOV TOTCSRS,R1 ;GET CSR'S
13338	054566					BEGIN LCSRSAVE
13339	054566					FOR CSRNO := #0 TO #36 BY #2
13340	054572	006301				ASL R1
13341	054574					ON.ERROR
13342	054576	104426				READCSR
13343	054600					PUSH CSR
13344	054604					END ;OF ON.ERROR
13345	054604					IF R1 EQ #0 THEN LEAVE LCSRSAVE
13346	054610					END ;OF FOR CSRNO
13347	054626					END LCSRSAVE
13348						;SAVE MMRO,1,2,3
13349	054626					PUSH MMRO,MMR1,MMR2
13350	054642	005737	002452			TST NOSUPER
13351	054646	001002				BNE 8\$
13352	054650					PUSH MMR3
13353						;SAVE KERNEL PAR'S
13354	054654	012700	172400	8\$:		MOV #172400,R0
13355	054660	012701	000020			MOV #16,R1
13356	054664			4\$:		PUSH -(R0)
13357	054666	077102				SOB R1,4\$
13358						;SAVE UNIBUS MAP REGISTERS
13359	054670	022737	000001	003752		CMP #1,PROTYP ;IS THIS AN 11/44?
13360	054676	001004				BNE 9\$ ;BRANCH IF NOT
13361	054700					PUSH MAPH0,MAPL0
13362						;SAVE POSSIBLE SOFTWARE SWITCH REGISTER
13363	054710			9\$:		PUSH @SWR
13364						;SAVE STACK POINTER
13365	054714	010637	055334			MOV SP,\$SAVR6 ;;SAVE SP
13366						;NOW SET UP REAL VECTOR
13367	054720	012737	054732	000024		MOV #SPWRUP,PWRVEC ;;SET UP VECTOR
13368	054726	000000				HALT
13369	054730	000776		\$DOWN:		BR \$DOWN ;;HANG UP

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 428  
ROUTINE POWER DOWN AND UP

```

13372 *****
13373 :POWER UP ROUTINE
13374 054732 $PWRUP:
13378 054732 012737 055330 000024 MOV    #SILLUP,PWRVEC ;;SET FOR FAST DOWN
13379 :RESTORE STACK POINTER
13380 054740 013706 055334 MOV    $SAVR6,SP      ;;GET SP
13381 054744 005037 055334 CLR    $SAVR6        ;;WAIT LOOP FOR THE TTY
13382 054750 005237 055334 1$: INC    $SAVR6        ;;WAIT FOR THE INC
13383 054754 001375 BNE     1$          ;;OF A WORD
13384 :RESTORE POSSIBLE SOFTWARE SWITCH REGISTER
13385 054756 POP     @SWR
13386 :RESTORE UNIBUS MAP
13387 054762 022737 000001 003752 CMP    #1,PROTYP      ;IS THIS AN 11/44?
13388 054770 001006 BNE     10$
13389 054772 POP     MAPLO,MAPHO
13390 055002 004737 046344 CALL    LOWMAP        ;SETUP LOWER 16K OF UNIBUS MAP
13391 :RESTORE KERNEL PAR'S & PDR'S
13392 055006 012700 172340 10$: MOV    #172340,R0
13393 055012 012702 172300 MOV    #KIPDR0,R2
13394 055016 012701 000020 MOV    #16.,R1
13395 055022 6$: POP     (R0)+
13396 055024 012722 077406 MOV    #77406,(R2)+
13397 055030 077104 SOB     R1,6$
13398 :RESTORE MMR3,2,1,0
13399 055032 005737 002452 TST     NOSUPER
13400 055036 001002 BNE     11$
13401 055040 POP     MMR3
13402 055044 11$: POP     MMR2,MMR1,MMR0
13403 :RESTORE ECC REGISTERS
13404 055060 013701 002222 MOV    TOTCSRS,R1      ;GET CSR'S
13405 055064 042701 177400 BIC     #177400,R1
13406 055070 BEGIN LCSRRESTORE
13407 055070 FOR CSRNO := #36 DOWNT0 #0 BY #2
13408 055076 006201 ASR     R1
13409 055100 ON.ERROR
13410 055102 POP     CSR
13411 055106 104425 LOADCSR
13412 055110 END ;OF ON.ERROR
13413 055110 IF R1 EQ #0 THEN LEAVE LCSRRESTORE
13414 055114 END ;OF FOR CSRNO
13415 055132 END LCSRRESTORE
13416 :COPY KERNEL MAP TO USER & SUPERVISOR
13417 055132 012700 172300 MOV    #KIPDR0,R0
13418 055136 012701 177600 MOV    #UIPDR0,R1
13419 055142 012702 172200 MOV    #SIPDR0,R2
13420 055146 012703 000040 MOV    #32.,R3
13421 055152 011021 3$: MOV    (R0),(R1)+
13422 055154 012022 MOV    (R0)+,(R2)+
13423 055156 077303 SOB     R3,3$

```

CZI:SPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 429  
ROUTINE POWER DOWN AND UP

13425					:RESTORE SUPERVISOR & USER STACK POINTERS	
13426	055160	005737	002452		TST NOSUPER	
13427	055164	001006			BNE 13\$	
13428	055166				POP R0	
13429	055170				SUPERV'SOR	:ENTER SUPERVISOR MODE
13430	055176	010006			MOV R0,SSP	
13431	055200	104417			KERNEL	:ENTER KERNEL MODE
13432	055202			13\$:	POP R0	
13433	055204				USER	
13434	055212	010006			MOV R0,USP	
13435	055214	104417			KERNEL	:ENTER KERNEL MODE
13436					:RESTORE SUPERVISOR PAR'S	
13437	055216	012700	172240		MOV #172240,R0	
13438	055222	012701	000020		MOV #16.,R1	
13439	055226			7\$:	POP (R0)+	
13440	055230	077102			SUB R1,7\$	
13441					:RESTORE USER PAR'S & PDR7	
13442	055232	012700	177636		MOV #177636,R0	
13443	055236	012701	000021		MOV #17.,R1	
13444	055242			8\$:	POP (R0)+	
13445	055244	077102			SUB R1,8\$	
13446					:RESTORE POSSIBLE SOFTWARE DISPLAY REGISTER	
13447	055246	013777	002010 125350		MOV \$PATMAR,@DISPLAY	
13448	055254	013737	002010 000174		MOV \$PATMAR,DISPREG	
13449	055262				POP CSRNO,R5,R4,R3,R2,R1,R0	
13450	055302	012737	054356 000024		MOV \$SPWRDN,PWRVEC	::SET UP THE POWER DOWN VECTOR
13451	055310				TYP MSG051	:REPORT THE POWER FAILURE
13452					:RESTORE CACHE STATUS	
13453	055314	005737	002540		T CACHKN	
13454	055320	001402			L:Q 9\$	
13455	055322				FOP CONTRL	
13456	055326	000002		9\$:	RTI	
13457	055330	000000		\$ILLUP:	HALT	::THE POWER UP SEQUENCE WAS STARTED
13458	055332	000776			BR \$ILLUP	::BEFORE THE POWER DOWN WAS COMPLETE
13459	055334	000000		\$SAVR6:	0	::PUT THE SP HERE
13460					.EVEN	

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 430  
ROUTINE POWER DOWN AND UP

13472 055336

WOOPS: SUBTST &lt;&lt;POWER FAIL WHILE RELOCATED&gt;&gt;

\*\*\*\*\*  
:SUBTEST POWER FAIL WHILE RELOCATED  
\*\*\*\*\*

13473 055336			PUSH BANK	
13474 055342	005037	002100	CLR BANK	
13475 055346			MAP BANK	;MAP SUPERVISOR SPACE (TEST AREA) TO BANK
13476 055362			SUPERVISOR	;ENTER SUPERVISOR MODE
13477 055370	013737	060024	MOV FIRST+PWRVEC,WOOPSAV	
13478 055376	013737	060026	MOV FIRST+PWRVEC+2,WOOPSAV+2	
13479 055404			BMOV FIRST+WOOPUP,WOOPSAV+4,WOOPEND-WOOPUP/2+12.	
13480 055416	012737	055522	MOV #WOOPUP,FIRST+PWRVEC	
13481 055424	012737	000340	MOV #340,FIRST+PWRVEC+2	
13482 055432			BMOV WOOPUP,FIRST+WOOPUP,WOOPEND-WOOPUP/2	
13483 055444	012700	172340	MOV #KIPAR0,R0	
13484 055450	012701	135704	MOV #FIRST+WOOPEND,R1	
13485 055454	012702	000010	MOV #8,R2	
13486 055460	012021		1\$: MOV (R0)+,(R1)+	
13487 055462	077202		SOB R2,1\$	
13488 055464	005737	002452	TST NOSUPER	
13489 055470	001002		BNE 2\$	
13490 055472	013721	172516	MOV MMR3,(R1)+	
13491 055476	013721	177576	2\$: MOV MMR2,(R1)+	
13492 055502	013721	177574	MOV MMR1,(R1)+	
13493 055506	013721	177572	MOV MMR0,(R1)+	
13494 055512	104417		KERNEL	;ENTER KERNEL MODE
13495 055514			POP BANK	
13496 055520	000207		RETURN	

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 432  
POWER FAIL WHILE RELOCATED

13499 055522

WOOPUP: SUBTST &lt;&lt;POWER UP FROM BANK 0 TO RELOCATION&gt;&gt;

\*\*\*\*\*  
:SUBTEST POWER UP FROM BANK 0 TO RELOCATION  
\*\*\*\*\*

13500	055522	012700	055704		MOV	#WOOPEND,R0	
13501	055526	012701	172340		MOV	#KIPAR0,R1	
13502	055532	012703	172300		MOV	#KIPDR0,R3	
13503	055536	012702	000010		MOV	#8,R2	
13504	055542	012021		1\$:	MOV	(R0)+,(R1)+	
13505	055544	012723	077406		MOV	#77406,(R3)+	
13506	055550	077204			SOS	R2,1\$	
13507	055552	005737	002452		TST	NOSUPER	
13508	055556	001002			BNE	3\$	
13509	055560	012037	172516		MOV	(R0)+,MMR3	
13510	055564	012037	177576	3\$:	MOV	(R0)+,MMR2	
13511	055570	012037	177574		MOV	(R0)+,MMR1	
13512	055574	012037	177572		MOV	(R0)+,MMR0	
13513	055600	013706	055334		MOV	\$SAVR6,SP	
13514	055604				PUSH	BANK	
13515	055610	005037	002100		CLR	BANK	
13516	055614				MAP	BANK	
13517	055630				SUPERVISOR		;MAP SUPERVISOR SPACE (TEST AREA) TO BANK
13518	055636	013737	055734	060024	MOV	WOOPSAV,FIRST+PWRVEC	;ENTER SUPERVISOR MODE
13519	055644	013737	055736	060026	MOV	WOOPSAV+2,FIRST+PWRVEC+2	
13520							;SIMULATE THE FOLLOWING BLOCK MOV BUT WITH NO STACK ACCESSES
13521					:BMOV	WOOPSAV+4,FIRST+WOOPUP,WOOPEND-WOOPUP/2+12.	
13522	055652	012700	055740		MOV	#WOOPSAV+4,R0	
13523	055656	012701	000105		MOV	#WOOPEND-WOOPUP/2+12.,R1	
13524	055662	012702	135522		MOV	#FIRST+WOOPUP,R2	
13525	055666	012022		2\$:	MOV	(R0)+,(R2)+	
13526	055670	077102			SOS	R1,2\$	
13527							
13528	055672	104417			KERNEL		;ENTER KERNEL MODE
13529	055674				POP	BANK	
13530	055700	000137	054732		JMP	\$PWRUP	
13531	055704	000014			WOOPEND: .REPT	12.	
13534	055734	000107			WOOPSAV: .REPT	WOOPEND-WOOPUP/2+12.+2	



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 434  
IO SUBROUTINES

```

13539          .SBTTL IO SUBROUTINES
13540
13541          .SBTTL ROUTINE TYPE
13542
13543          *****
13544          *ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
13545          *THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
13546          *NOTE1:      $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
13547          *NOTE2:      $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
13548          *NOTE3:      $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
13549          *
13550          *CALL:
13551          *1) USING A TRAP INSTRUCTION
13552          *      TYPE      MESADR      ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
13553          *OR
13554          *      TYPE
13555          *      MESADR
13556          *
13557
13558 056152 105737 002554 $TYPE: TSTB $TPFLG      ;;IS THERE A TERMINAL?
13559 056156 100407      BMI 6$      ;;BR IF NO
13560 056160 010046      MOV RO,-(SP)      ;;SAVE RO
13561 056162 017600 000002      MOV @2(SP),RO      ;;GET ADDRESS OF ASCIZ STRING
13562 056166 112046      MOVB (RO)+,-(SP)      ;;PUSH CHARACTER TO BE TYPED ONTO STACK
13563 056170 001005      BNE 7$      ;;BR IF IT ISN'T THE TERMINATOR
13564 056172 005726      TST (SP)+      ;;IF TERMINATOR POP IT OFF THE STACK
13565 056174 012600      MOV (SP)+,RO      ;;RESTORE RO
13566 056176 062716 000002      ADD #2,(SP)      ;;ADJUST RETURN PC
13567 056202 000002      RTI      ;;RETURN
13568 056204 122716 000011      7$: CMPB #HT,(SP)      ;;BRANCH IF NOT <HT>
13569 056210 001002      BNE 11$      11$:
13570 056212 112716 000040      MOVB #'',(SP)      ;;REPLACE TAB WITH SPACE
13571 056216 122716 000200      CMPB #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
13572 056222 001006      BNE 8$      8$:
13573 056224 005726      TST (SP)+      ;;POP <CR><LF> EQUIV
13574 056226      TYPE      ;;TYPE A CR AND LF
13575 056230 002644      $CRLF
13576 056232 105037 056464      CLRB $CHARCNT      ;;CLEAR CHARACTER COUNT
13577 056236 000753      BR 4$      ;;GET NEXT CHARACTER
13578 056240 004737 056300      8$: CALL $TYPEC      ;;GO TYPE THIS CHARACTER
13579 056244 123726 002636      9$: CMPB $FILLC,(SP)+      ;;IS IT TIME FOR FILLER CHARS.?
13580 056250 001346      BNE 4$      ;;IF NO GO GET NEXT CHAR.
13581 056252 013746 002352      MOV $NULL,-(SP)      ;;GET # OF FILLER CHARS. NEEDED
13582      ;;AND THE NULL CHAR.
13583 056256 105366 000001      10$: DECB 1(SP)      ;;DOES A NULL NEED TO BE TYPED?
13584 056262 002770      BLT 9$      ;;BR IF NO--GO POP THE NULL OFF OF STACK
13585 056264 004737 056300      CALL $TYPEC      ;;GO TYPE A NULL
13586 056270 105337 056464      DECB $CHARCNT      ;;DO NOT COUNT AS A COUNT
13587 056274 000770      BR 10$      ;;LOOP
13588 056276 000000      XCHAR: .WORD 0
13589 056300      $TYPEC: PUSH R1
13590 056302 116601 000004      MOVB 4(SP),R1
13591 056306 005737 002540      TST CACHKN
13592 056312 001402      BEQ 2$      2$:
13593 056314      PUSH CONTRL
13594 056320      PUSH RO
13595 056322 104424      CACHOFF      ;TURN CACHE OFF

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 434-1  
ROUTINE TYPE

13620	056324	105777	124302		3\$:	TSTB	@STPS	::WAIT UNTIL PRINTER IS READY
13621	056330	100375				BPL	3\$	
13622	056332	005037	056276			CLR	XOCHAR	
13623	056336	105777	124264			TSTB	@STKS	::CHECK FOR XOFF
13624	056342	100032				BPL	NC	::SKIP IF NO CHARACTER
13625	056344	117737	124260	056276		MOVB	@STKB,XOCHAR	::SAVE THE CHARACTER
13626	056352	042737	177600	056276		BIC	#^C177,XOCHAR	::STRIP OFF ASCII
13627	056360	023727	056276	000023		CMP	XOCHAR,#023	::WAS IT A CONTROL S?
13628	056366	001020				BNE	NC	::BRANCH IF NOT
13629	056370	105777	124232		CONTS3:	TSTB	@STKS	::WAIT FOR CHARACTER
13630	056374	100375				BPL	CONTS3	
13631	056376	117737	124226	056276		MOVB	@STKB,XOCHAR	::GET CHARACTER
13632	056404	042737	177600	056276		BIC	#^C177,XOCHAR	::STRIP OFF ASCII
13633	056412					IF XOCHAR EQ #21		:: IF IT IS A ^Q
13634	056422	000402				BR	NC	
13635	056424					ELSE		
13636	056426	000760				BR	CONTS3	
13637	056430					END ;OF	IF XOCHAR	
13638	056430	110177	124200		NC:	MOVB	R1,@STPB	::LOAD CHAR TO BE TYPED INTO DATA REG.
13642	056434	122766	000015	000002		CMPB	#CR,2(SP)	::IS CHARACTER A CARRIAGE RETURN?
13643	056442	001003				BNE	1\$	::BRANCH IF NO
13644	056444	105037	056464			CLRB	\$CHARCNT	::YES--CLEAR CHARACTER COUNT
13645	056450	000406				BR	\$TYPEX	::EXIT
13646	056452	122766	000012	000002	1\$:	CMPB	#LF,2(SP)	::IS CHARACTER A LINE FEED?
13647	056460	001402				BEQ	\$TYPEX	::BRANCH IF YES
13648	056462	105227				INCB	(PC)+	::COUNT THE CHARACTER
13649	056464	000000			\$CHARCNT:	.WORD	0	::CHARACTER COUNT STORAGE
13650	056466				\$TYPEX:	POP	R0	
13651	056470	005737	002540			TST	CACHKN	::IS THERE A CACHE?
13652	056474	001402				BEQ	2\$	::BRANCH IF NOT
13653	056476					POP	CONTRL	::POP CACHE STATUS
13654	056502				2\$:	POP	R1	
13655	056504	000207				RETURN		
13656	056506				SUPLIMIT:	;	!!!!!!THIS IS THE LIMIT ON SUPERVISOR MAPPED TO MJT SPACE	

CZMSPAO MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 449  
 ERROR DATA SETUP

	.SBTTL	ERROR DATA SETUP
14334		
14335		
14336	USE THIS	IF THIS CONDITION DISCRIBES THE ERROR
14337		
14338	PERR01	TRAP
14339		BAD DATA IN R0 UNLESS ABORTED
14340		THEN BAD DATA IS POINTED TO BY -(R4)
14341		GOOD DATA IN R5
14342		
14343	PERR02	TRAP
14344		BAD DATA IN R1 UNLESS ABORTED
14345		THEN BAD DATA IS POINTED TO BY -(R4)
14346		GOOD DATA IN R2
14347		
14348	PERR03	TRAP
14349		BAD DATA IS POINTED TO BY -(R1)
14350		GOOD DATA IN R4
14351		
14352	PERR04	TRAP
14353		BAD DATA IN R4 UNLESS ABORTED
14354		THEN BAD DATA IS POINTED TO BY -2(R0)
14355		GOOD DATA IN R2
14356		
14357	PERR05	JSR PC
14358		BAD DATA IS POINTED TO BY -(R0)
14359		GOOD DATA IN R2
14360		RETURN AFTER SETTING UP GOOD,BAD,ADDRESS
14361		
14362	PERR06	JSR PC
14363		BAD DATA IS POINTED TO BY -(R0)
14364		GOOD DATA IS ZERO
14365		RETURN AFTER SETTING UP GOOD,BAD,ADDRESS
14366		
14367	PERR07	TRAP
14368		BAD DATA IN R2 UNLESS ABORTED
14369		THEN BAD DATA IS POINTED TO BY (R1)
14370		GOOD DATA IN DATBUF
14371		
14372	PERR10	TRAP
14373		BAD DATA IN R2 UNLESS ABORTED
14374		THEN BAD DATA IS POINTED TO BY 2(R1)
14375		GOOD DATA IN DATBUF+2
14376		
14377	PERR11	TRAP
14378		BYTE TEST
14379		BAD DATA IN RIGHT BYTE OF R0 UNLESS ABORTED
14380		THEN BAD DATA IS POINTED TO BY (R1)
14381		GOOD DATA IS A ZERO BYTE
14382		
14383	PERR12	TRAP
14384		BYTE TEST
14385		BAD DATA IN RIGHT BYTE OF R0 UNLESS ABORTED
14386		THEN BAD DATA IS POINTED TO BY (R1)
14387		GOOD DATA IS A BYTE OF ONES
14388		
14389	PERR13	TRAP
14390		BAD DATA IN R0 UNLESS ABORTED

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 449-1  
ERROR DATA SETUP

14391	:		THEN BAD DATA IS POINTED TO BY (R1)
14392	:		GOOD DATA IS ZERO
14393	:		
14394	:	PERR14	TRAP
14395	:		BAD DATA IN R0 UNLESS ABORTED
14396	:		THEN BAD DATA IS POINTED TO BY (R1)
14397	:		GOOD DATA IS ONES
14398	:		
14399	:	PERR15	TRAP
14400	:		BAD DATA IN R0 UNLESS ABORTED
14401	:		THEN BAD DATA IS POINTED TO BY (R1)
14402	:		GOOD DATA IN TSTDAT
14403	:		
14404	:	PERR16	TRAP
14405	:		BAD DATA IN R0 UNLESS ABORTED
14406	:		THEN BAD DATA IS POINTED TO BY (R1)
14407	:		GOOD DATA IN TSTDAT+2
14408	:		
14409	:	PERR17	TRAP
14410	:		BAD DATA IN R0 UNLESS ABORTED
14411	:		THEN BAD DATA IS POINTED TO BY (R1)
14412	:		GOOD DATA IN R2
14413	:		
14414	:	PERR20	TRAP
14415	:		BAD DATA IN R0 UNLESS ABORTED
14416	:		THEN BAD DATA IS POINTED TO BY (R1)
14417	:		GOOD DATA IN R3
14418	:		
14419	:	PERR21	TRAP
14420	:		7 BIT BYTE TEST
14421	:		BAD DATA IN RIGHT BYTE OF R0 UNLESS ABORTED
14422	:		THEN BAD DATA IS POINTED TO BY (R1)
14423	:		GOOD DATA IS A 7 BIT BYTE ON ONES
14424	:		
14425	:	PERR22	TRAP
14426	:		BAD DATA IN R2 UNLESS ABORTED
14427	:		THEN BAD DATA IS POINTED TO BY (R1)
14428	:		GOOD DATA IN R0
14429	:		
14430	:	PERR23	TRAP
14431	:		BAD DATA IN R0 UNLESS ABORTED
14432	:		THEN BAD DATA IS POINTED TO BY (R1)
14433	:		GOOD DATA IN R4
14434	:		
14435	:	PERR24	TRAP
14436	:		BAD DATA IN R0 UNLESS ABORTED
14437	:		THEN BAD DATA IS POINTED TO BY (R2)
14438	:		GOOD DATA IN R3
14439	:		
14440	:	PERR25	TRAP
14441	:		BAD DATA POINTED TO BY -(R1)
14442	:		GOOD DATA IN R2 UNLESS LOC V177654 IS SET
14443	:		THEN GOOD DATA IS IN R3
14444	:		
14445	:	PERR26	TRAP
14446	:		BAD DATA IS DOUBLE WORD POINTED TO BY R1 AND IN LOW 7 BITS OF R0
14447	:		GOOD DATA IS 000000,,100000,,100

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 449-2  
 ERROR DATA SETUP

14448	:		
14449	:	PERR27	TRAP
14450	:		BAD DATA IS DOUBLE WORD POINTED TO BY R1 AND IN LOW 7 BITS OF R0
14451	:		GOOD DATA IS 000000,,000000,,077
14452	:		
14453	:	PERR30	TRAP
14454	:		BAD DATA IS POINTED TO BY -16(SP)
14455	:		GOOD DATA IS IN R1
14456	:		
14457	:	PERR31	TRAP
14458	:		SPECIAL ECC FAILURE HANDLER
14459	:		
14460	:	PERR32	TRAP
14461	:		SPECIAL ECC FAILURE HANDLER
14462	:		
14463	:	PERR33	TRAP
14464	:		SPECIAL ECC FAILURE HANDLER
14465	:		
14466	:	PERR34	TRAP
14467	:		SPECIAL ECC FAILURE HANDLER
14468	:		
14469	:	PERR35	TRAP
14470	:		SPECIAL BRANCH GOBBLE FAILURE HANDLER
14471	:		
14472	:	CALLING	SEQUENCE FOR TRAP TYPES
14473	:	BEQ	2\$ ;NO - ERROR,BRANCH FOR CARD
14474	:	PERRXX	;TRAP TO ERROR ROUTINE
14475	:	2\$: NEXT	INSTRUCTION ;CONTINUE TESTING

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 451  
ERROR DATA SETUP

14478	056506	010437	002032		\$PER01: MOV	R4,ADDRESS	
14479	056512	162737	000002	002032	SUB	#2,ADDRESS	
14480	056520	010037	002050		MOV	R0,BAD	
14481	056524	010537	002042		MOV	R5,GOOD	
14482	056530	000137	057242		JMP	PERRAW	
14483							
14484	056534	010437	002032		\$PER02: MOV	R4,ADDRESS	
14485	056540	162737	000002	002032	SUB	#2,ADDRESS	
14486	056546	010137	002050		MOV	R1,BAD	
14487	056552	010237	002042		MOV	R2,GOOD	
14488	056556	000137	057242		JMP	PERRAW	
14489							
14490	056562	010137	002032		\$PER03: MOV	R1,ADDRESS	
14491	056566	162737	000002	002032	SUB	#2,ADDRESS	
14492	056574	010437	002042		MOV	R4,GOOD	
14493	056600	016137	177776	002050	MOV	-2(R1),BAD	
14494	056606	000137	057242		JMP	PERRAW	
14495							
14496	056612	010037	002032		\$PER04: MOV	R0,ADDRESS	
14497	056616	162737	000002	002032	SUB	#2,ADDRESS	
14498	056624	010437	002050		MOV	R4,BAD	
14499	056630	010237	002042		MOV	R2,GOOD	
14500	056634	000137	057242		JMP	PERRAW	
14501							
14502	056640	010237	002042		PERR05: MOV	R2,GOOD	
14503	056644	014037	002050		PERR05: MOV	-(R0),BAD	
14504	056650	010037	002032		MOV	R0,ADDRESS	
14505	056654	062700	000002		ADD	#2,R0	;RESTORE R0
14506	056660	004737	042614		CALL	BADSTACK	
14507	056664	000207			RETURN		
14508							
14509	056666	005037	002042		PERR06: CLR	GOOD	
14510	056672	000764			BR	PERR05	
14511							
14512	056674	010137	002032		\$PER07: MOV	R1,ADDRESS	
14513	056700	010237	002050		MOV	R2,BAD	
14514	056704	013737	002240	002042	MOV	DATBUF,GOOD	
14515	056712	000137	057242		JMP	PERRAW	
14516							
14517	056716				\$PER10: LET	ADDRESS := R1 + #2	
14518	056730				LET	BAD := R2	
14519	056734				LET	GOOD := DATBUF+2	
14520	056742	000137	057242		JMP	PERRAW	
14521							
14522	056746				\$PER11: LET	ADDRESS := R1	
14523	056752				LET	BAD := R0	
14524	056756				LET	GOOD := #0	
14525	056762	000137	057314		JMP	PERRAB	
14526							
14527	056766				\$PER12: LET	ADDRESS := R1	
14528	056772				LET	BAD := R0	
14529	056776				LET	GOOD := #377	
14530	057004	000137	057314		JMP	PERRAB	

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 453  
ERROR DATA SETUP

14533 057010		\$PER13: LET ADDRESS := R1
14534 057014		LET BAD := R0
14535 057020		LET GOOD := #0
14536 057024	000137 057242	JMP PERRAW
14537		
14538 057030		\$PER14: LET ADDRESS := R1
14539 057034		LET BAD := R0
14540 057040		LET GOOD := ONES
14541 057046	000137 057242	JMP PERRAW
14542		
14543 057052		\$PER15: LET ADDRESS := R1
14544 057056		LET BAD := R0
14545 057062		LET GOOD := TSTDAT
14546 057070	000137 057242	JMP PERRAW
14547		
14548 057074		\$PER16: LET ADDRESS := R1
14549 057100		LET BAD := R0
14550 057104		LET GOOD := TSTDAT+2
14551 057112	000453	BR PERRAW
14552		
14553 057114		\$PER17: LET ADDRESS := R1
14554 057120		LET BAD := R0
14555 057124		LET GOOD := R2
14556 057130	000444	BR PERRAW
14557		
14558 057132		\$PER20: LET ADDRESS := R1
14559 057136		LET BAD := R0
14560 057142		LET GOOD := R3
14561 057146	000435	BR PERRAW
14562		
14563 057150		\$PER21: LET ADDRESS := R1
14564 057154		LET BAD := R0
14565 057160		LET GOOD := #177
14566 057166	000477	BR PERRA7
14567		
14568 057170		\$PER22: LET ADDRESS := R1
14569 057174		LET BAD := R2
14570 057200		LET GOOD := R0
14571 057204	000416	BR PERRAW
14572		
14573 057206		\$PER23: LET ADDRESS := R1
14574 057212		LET BAD := R0
14575 057216		LET GOOD := R4
14576 057222	000407	BR PERRAW
14577		
14578 057224		\$PER24: LET ADDRESS := R2
14579 057230		LET BAD := R0
14580 057234		LET GOOD := R3
14581 057240	000400	BR PERRAW

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 454  
 ERROR DATA SETUP

14583 057242

PERRAW: SUBTST <<DATA WAS A WORD>>

\*\*\*\*\*  
 :\*SUBTEST DATA WAS A WORD  
 :\*\*\*\*\*

14584 057242 004737 057476

14585 057246

14586 057260

14587 057272 004737 057452

14588 057276

14589 057304 104011

14590 057306

14591 057310 104012

14592 057312

14593 057312 000002

14594

14595 057314

CALL PERBNK  
 IF ABORTFLAG IS TRUE THEN SCALL GETDATA  
 IF BADPC EQ #0 THEN SCALL BADSTACK  
 CALL PERXOR  
 IF ABORTFLAG IS FALSE  
 ERROR +11  
 ELSE  
 ERROR +12  
 END ;OF IF ABORTFLAG  
 RTI

PERRAB: SUBTST <<DATA WAS A BYTE>>

\*\*\*\*\*  
 :\*SUBTEST DATA WAS A BYTE  
 :\*\*\*\*\*

14596 057314 004737 057476

14597 057320

14598 057332

14599 057344 004737 057452

14600 057350

14601 057356 104014

14602 057360

14603 057362 104015

14604 057364

14605 057364 000002

CALL PERBNK  
 IF ABORTFLAG IS TRUE THEN SCALL GETDATA  
 IF BADPC EQ #0 THEN SCALL BADSTACK  
 CALL PERXOR  
 IF ABORTFLAG IS FALSE  
 ERROR +14  
 ELSE  
 ERROR +15  
 END ;OF IF ABORTFLAG  
 RTI



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 456  
DATA WAS A BYTE

14608 057366

PERRA7: SUBTST <<DATA WAS A 7 BIT BYTE>>

\*\*\*\*\*  
:~SUBTEST DATA WAS A 7 BIT BYTE  
\*\*\*\*\*

14609 057366  
14610 057400 004737 057452  
14611 057404 004737 057476  
14612 057410 104022  
14613 057412 000002

IF BADPC EQ #0 THEN \$CALL BADSTACK  
CALL PERXCR  
CALL PERBNK  
ERROR +22  
RTI

14614  
14615 057414  
14616 057422  
14617 057430 000137 054044  
14618  
14619 057434 005037 002044  
14620 057440  
14621 057446 000137 054044  
14622  
14623 057452

\$PER26: LET GOOD2 := #100000  
LET GOOD3 := #100  
JMP PERRA3

\$PER27: CLR GOOD2  
LET GOOD3 := #077  
JMP PERRA3

PERXOR: SUBTST <<DETERMINE XOR OF GOOD & BAD>>

\*\*\*\*\*  
:~SUBTEST DETERMINE XOR OF GOOD & BAD  
\*\*\*\*\*

14624 057452  
14625 057454 013700 002042  
14626 057460 013737 002050 002056  
14627 057466 074037 002056  
14628 057472  
14629 057474 000207

PUSH R0  
MOV GOOD,R0  
MOV BAD,BAD XOR  
XOR R0,BAD XOR  
POP R0  
RETURN

14632 057476

```

*****
*SUBTEST          LOG ERROR ON BAD BANK
*****

```

```

PUSH    R0,R1
MOV     BANK,R1
ASL     R1
ASL     R1
BIS     #BIT0,CONFIG(R1)
INCB    CONFIG+2(R1)           ;BUMP BANK COUNTER
BNE     12$                   ;NO OVERFLOW - SKIP
DECB    CONFIG+2(R1)           ;SET BACK TO 255.
12$:    CMPB    CONFIG+2(R1),ERRMAX ;IS IT PAST MAX?
BLOS    11$                   ;NO - SKIP
SET     TOOMANY                ;YES
11$:    POP     R1,R0
        RETURN

```

```
PERECC:  MOV      R0,BAD
          IF ADDRESS EQ TESTADD
            MOV TSTDAT,GOOD
          ELSE
            MOV     TSTDAT+2,GOOD
          END ;OF IF (R1)
          CALL     PERXOR
          SET      HEADER
          RETURN
```

```

$PER31: IF REALPAT EQ #41
        ERROR +23
        END
        IF BADPC EQ #0 THEN $CALL BADSTACK
        CALL PERECC
        IF REALPAT EQ #11
            ERROR +37
        END ;OF IF REALPAT
        IF REALPAT EQ #15
            ERROR +43
        END ;OF IF REALPAT
        IF REALPAT EQ #16
            ERROR +44
        END ;OF IF REALPAT
        SET HEADER
        RTI

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 460  
LOG ERROR ON BAD BANK

14676	057722			\$PER32: IF BADPC EQ #0 THEN \$CALL BADSTACK
14677	057734	010137	002032	MOV R1,ADDRESS
14678	057740	010037	002050	MOV R0,BAD
14679	057744	010237	002042	MOV R2,GOOD
14680	057750			SET HEADER
14681	057756	104040		ERROR +40
14682	057760			SET HEADER
14683	057766	000002		RTI
14684				
14685	057770			\$PER33: IF BADPC EQ #0 THEN \$CALL BADSTACK
14686	060002	010137	002032	MOV R1,ADDRESS
14687	060006	010037	002050	MOV R0,BAD
14688	060012	105037	002051	CLRB BAD+1
14689	060016	012737	000377	MOV #377,GOOD
14690	060024	004737	057452	CALL PERXOR
14691	060030			SET HEADER
14692	060036	104041		ERROR +41
14693	060040			SET HEADER
14694	060046	000002		RTI
14695				
14696	060050			\$PER34: IF BADPC EQ #0 THEN \$CALL BADSTACK
14697	060062			IF #BIT15!BIT4 OFF.IN CSR
14698	060072	104016		ERROR +16 ;NO SBE OR DBE
14699	060074			ELSE
14700	060076	104001		ERROR +1 ;EXPECTED SBE SO DBE MUST HAVE GOTTEN SET
14701	060100			END :OF IF #BIT15!BIT4
14702	060100	000002		RTI
14703				
14704				;DURING BRANCH GOBBLE THE CONDITION CODES WERE WRONG
14705	060102	004737	057476	\$PER35: CALL PERBNK
14706	060106	004737	042614	CALL BADSTACK
14707	060112	013737	002030	MOV BADPSW,BAD
14708	060120	012737	000012	MOV #12,GOOD
14709	060126	104047		ERROR +47
14710	060130	062706	000004	ADD #4,SP ;FIX STACK FROM TRAP
14711	060134	000207		RETURN ;ABORTING TEST
14712				
14713	060136	010037	002042	\$PER36: MOV R0,GOOD
14714	060142	010137	002050	MOV R1,BAD
14715	060146			SET HEADER
14716	060154	104023		ERROR +23
14717	060156			SET HEADER
14718	060164	000002		RTI

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 462  
ROUTINE SCOPE HANDLER

```

14721 .SBTTL ROUTINE SCOPE HANDLER
14722 *****
14723 *THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
14724 *AND LOAD THE DISPLAY DATA INTO THE DISPLAY REGISTER
14725 *THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
14726 *SW14=1 LOOP ON TEST
14727 *SW9=1 LOOP ON ERROR
14728 *CALL
14729 *
14730 060166 005237 065650 $SCOPE: INC $DEVCT ;;SCOPE=IOT ;TELL APT WE ARE ALIVE
14731 060172 IF RESULT IS LT
14732 060174 005037 065650 CLR $DEVCT
14733 060200 105237 065652 INCB $UNIT
14734 060204 END ;OF IF RESULT
14735 060204 104410 CKSWR ;;TEST FOR CHANGE IN SOFT-SWR
14736 060206 005737 006204 TST TRACE
14737 060212 001402 BEQ NOTRCE
14738 060214 004737 064144 CALL CONTT ;TRACE
14739 060220 NOTRCE:
14740 060220 005737 061322 TST CPERRF ;IS THERE A CPU ERROR REGISTER? ;R-C
14741 060224 001410 BEQ SKJ ;BRANCH IF NOT ;R-C
14742 060226 013737 177766 061320 MOV @#177766,CPSAVE ;GET CONTENTS OF ERROR REGISTER ;R-C
14743 060234 032737 000001 061320 BIT #BIT0,CPSAVE ;IS THE POWER FAIL MONITOR BIT SET? ;R-C
14744 060242 001401 BEQ SKJ ;BRANCH IF NOT ;R-C
14745 060244 104177 ERROR +177 ;REPORT IF SO ;R-C
14754 060246 SKJ: IF STOPOK IS TRUE AND #SW8 SET.IN @SWR ;R-C
14755 060264 005037 002414 CLR STOPOK
14756 060270 000137 047632 JMP EXIT
14757 060274 END ;OF IF STOPOK
14758 060274 IF NOSCOPE IS TRUE
14759 060302 000002 RTI
14760 060304 END ;OF IF NOSCOPE
14761 060304 1$: IF #SW14 SET.IN @SWR THEN GOTO $OVER
14762 :*****START OF CODE FOR THE XOR TESTER*****
14763 060314 000425 $XTSTR: BR 2$ ;;IF RUNNING ON THE 'XOR' TESTER CHANGE
14764 MOV ERRVEC,-(SP) ;;THIS INSTRUCTION TO A 'NOP' (NOP=240)
14765 060316 013746 000004 MOV #1$,ERRVEC ;;SAVE THE CONTENTS OF THE ERROR VECTOR
14766 060322 012737 060342 000004 TST 177060 ;;SET FOR TIMEOUT
14767 060330 005737 177060 MOV (SP)+,ERRVEC ;;RESTORE THE ERROR VECTOR
14768 060334 012637 000004 BR $SVLAD ;;GO TO THE NEXT TEST
14769 060340 000430 ADD #4,SP ;;FIX STACK FROM TRAP
14770 060342 062706 000004 1$: CMP #1$,PROTYP ;;IS THIS AN 11/44?
14771 060346 022737 000001 00375~ BNE 6$ ;BRANCH IF NOT
14772 060354 001002 CLR CPUERR ;RESET CPU ERROR REGISTER
14773 060356 005737 177766 6$: MOV (SP)+,ERRVEC ;RESTORE THE ERROR VECTOR
14774 060362 012637 000004 BR 4$ ;LOOP ON THE PRESENT TEST
14775 060366 000407 2$;*****END OF CODE FOR THE XOR TESTER*****
14776 060370 3$: TSTB $ERFLG ;HAS AN ERROR OCCURRED?
14777 060370 105737 002012 BEQ $SVLAD ;BR IF NO
14778 060374 001412 BIT #SW9,@SWR ;LOOP ON ERROR?
14779 060376 032777 001000 122216 BEQ 5$ ;BR IF NO
14780 060404 001404 4$: MOV $LPERR,$LPADR ;SET LOOP ADDRESS TO LAST SCOPE
14781 060406 013737 002610 002606 BR $OVER
14782 060414 000410 5$: CLRB $ERFLG ;ZERO THE ERROR FLAG
14783 060416 105037 002012 $SVLAD: MOV (SP),$LPADR ;SAVE SCOPE LOOP ADDRESS
14784 060422 011637 002606 MOV (SP),$LPERR ;SAVE ERROR LOOP ADDRESS
14785 060426 011637 002610

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 462-1  
ROUTINE SCOPE HANDLER

14786	060432	005037	002356
14787	060436	004737	060450
14788	060442	013716	002606
14789	060446	000002	

\$OVER: CLR  
CALL  
MOV  
RTI

\$ESCAPE  
GETDIS  
\$LPADR,(SP)

::CLEAR THE ESCAPE FROM ERROR ADDRESS  
::FUDGE RETURN ADDRESS  
::FIXES PS

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 463  
 ROUTINE SCOPE HANDLER

14791 060450

GETDIS: SUBTST <<SUBR DISPLAY>>

\*\*\*\*\*  
 ;\*SUBTEST SUBR DISPLAY  
 ;\*\*\*\*\*

14792 060450 113737 002100 002011  
 14793 060456 113737 002274 002010  
 14794 060464  
 14795 060466 005737 002124  
 14796 060472 001403  
 14797 060474 052737 100000 002010  
 14798 060502  
 14802 060502 013777 002010 122114  
 14803 060510 013737 002010 000174  
 14804 060516  
 14805 060520 000207

1\$:  
 MOV BANK,\$BANK  
 MOVB REALPAT,\$PATMAR  
 PUSH R0  
 TST RLFLAG ;ARE WE RELOCATED?  
 BEQ 1\$ ;NO - SKIP  
 BIS #BIT15,\$PATMAR ;YES - SET MSB  
 MOV \$PATMAR,@DISPLAY  
 MOV \$PATMAR,DISPREG ;SOFTWARE DISPLAY REGISTER  
 POP R0  
 RETURN

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 465  
ROUTINE ERROR HANDLER

```

14808 .SBTTL ROUTINE ERROR HANDLER
14809 *****
14810 ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
14811 ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERRCP CALL
14812 ;*AND GO TO $ERRTYP ON ERROR
14813 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
14814 ;*SW15=1 HALT ON ERROR
14815 ;*SW13=1 INHIBIT ERROR TYPEOUTS
14816 ;*SW10=1 BELL ON ERROR
14817 ;*SW9=1 LOOP ON ERROR
14818 ;*CALL
14819 ;*
14820 ERROR N ;;ERROR=EMT AND N=ERROR ITEM NUMBER
14821
14822 060522 105037 061316 .ENABL LSB
14823 060526 CLR B IBSAVE ;R-C
14824 060534 104410 IF NOERROR IS FALSE
14825 060536 CKSWR ;;TEST FOR CHANGE IN SOFT-SWR ;R-C
14826 060536 105237 002012 BACK:
14827 060542 001775 1$: INCB $ERFLG ;;SET THE ERROR FLAG
14828 060544 004737 060450 BEQ 1$ ;;DON'T LET THE FLAG GO TO ZERO
14829 060550 013737 002010 065644 CALL GETDIS ;;SETUP DISPLAY STUFF
14830 060556 032777 002000 122036 MOV $PATMAR,$TESTN ;FOR APT
14831 060564 001404 BIT #SW10,$SWR ;;BELL ON ERROR?
14832 060566 BEQ 2$ ;;NO - SKIP
14833 060572 TYPE $BELL ;;RING BELL
14834 060576 005237 002614 TYPE MSG014 ;CONTROL Z
14835 060602 012737 077777 002614 2$: INC $ERTTL ;;COUNT THE NUMBER OF ERRORS
14836 060604 IF RESULT IS MI
14837 060612 MOV #77777,$ERTTL
14838 060612 END ;OF IF RESULT
14839 060612 011637 002016 END ;OF IF NOERROR
14840 060616 162737 000002 002016 MOV (SP),ERRPC ;;GET ADDRESS OF ERROR INSTRUCTION
14841 060624 010637 002022 SUB #2,ERRPC
14842 060630 016637 000002 002026 MOV SP,ERRSP
14843 060636 117737 121154 002013 MOV 2(SP),ERRPSW
14844 CMPB #177,$ITEMB ;;STRIP AND SAVE THE ERROR ITEM CODE
14845 060644 122737 000177 002013 BEQ 1001$ ;IS THIS THE POWER FAIL CALL? ;R-C
14846 060652 001431 TSTB IBSAVE ;BRANCH IF SO ;R-C
14847 060654 105737 061316 BNE 1000$ ;2ND ERROR CALL? ;R-C
14848 060660 001024 TST CPERRF ;BRANCH IF SO ;R-C
14849 060662 005737 061322 BEQ 1001$ ;IS THERE A CPU ERROR REGISTER? ;R-C
14850 060666 001423 MOV 177766,CPSAVE ;BRANCH IF NOT ;R-C
14851 060670 013737 177766 061320 BIT #BIT0,CPSAVE ;SAVE CONTENTS ;R-C
14852 060676 032737 000001 061320 BEQ 1001$ ;POWER MONITOR BIT SET? ;R-C
14853 060704 001414 BIC #BIT0,177766 ;BRANCH IF NOT ;R-C
14854 060706 042737 000001 177766 MOVB #177,$ITEMB ;CLEAR THE BIT ;R-C
14855 060714 112737 002013 061316 MOVB #177,$ITEMB ;MAKE IBSAVE NON-ZERO FOR DUAL CALL ;R-C
14856 060722 112737 000177 002013 BR 1001$ ;SET $ITEMB TO POWER FAIL POINTER ;R-C
14857 060730 000402 1000$: CLRB IBSAVE ;R-C
14858 060732 105037 061316 1001$: IF NOERROR IS FALSE ;R-C
14859 060736 IF BADPC NE #0
14860 060744 MOV BADPC,ERRPC
14861 060752 013737 002020 002016 SUB #2,ERRPC
14862 060760 162737 000002 002016 MOV BADSP,ERRSP
14863 060766 013737 002024 002022

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 465-1  
ROUTINE ERROR HANDLER

14865	060774	013737	002030	002026	MOV BADPSW,ERRPSW	
14866	061002	005037	002020		CLR BADPC	
14867	061006				END ;IF	
14868	061006	013737	002016	065642	MOV ERRPC,\$FATAL	;FOR APT
14869	061014	004737	057476		CALL PERBNK	::LOG ERROR ON BANK
14870	061020				IF #SW13 SET.IN @SWR	
14871	061030	000420			BR 3\$	
14872	061032				END ;OF IF #SW13	
14873	061032				IF #SW5 SET.IN @SWR AND TOOMANY IS TRUE	
14874	061050				GOTO 3\$	
14875	061052				END ;OF IF #SW5	
14876	061052				END ;OF IF NOERROR	
14877	061052	004737	061324		CALL \$ERRTYP	::GO TO USER ERROR ROUTINE
14878	061056				IF MONFLG IS TRUE	::SHOULD WE RETURN TO XXDP MONITOR???
14879	061064	013706	002270		MOV SAVMON,SP	::GET MONITOR ADDRESS
14880	061070	000207			RTS PC	::GO TO MONITOR
14881	061072				END	::



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 466  
ROUTINE ERROR HANDLER

14883	061072			3\$:	IF NOERROR IS FALSE	
14884	061100	005777	121516		TST @SWR	::HALT ON ERROR
14885	061104	100002			BPL 7\$	::SKIP IF CONTINUE
14886	061106	000000		\$HALT:	HALT	::HALT ON ERROR!
14887	061110	104410			CKSWR	::TEST FOR CHANGE IN SOFT-SWR
14888	061112			7\$:	IF NOSCOPE IS FALSE AND #SW9 SET.IN @SWR	::FUDGE RETURN FOR LOOPING
14889	061130	013716	002610		MOV \$LPERR,(SP)	
14890	061134				END :OF IF NOSCOPE	
14891	061134	005737	002356		TST \$ESCAPE	::CHECK FOR AN ESCAPE ADDRESS
14892	061140	001402			BEQ 9\$	::BR IF NONE
14893	061142	013716	002356		MOV \$ESCAPE,(SP)	::FUDGE RETURN ADDRESS FOR ESCAPE
14894	061146			9\$:	IF DETFLAG IS FALSE	
14895	061154	022737	000001 003752		CMP #1,PROTYP	:IS THIS AN 11/44?
14896	061162	001002			BNE 11\$	
14897	061164	005037	177766		CLR CPUERR	
14898	061170			11\$:	IF ACTFLAG IS TRUE OR APTFLAG IS TRUE OR FATAL\$ IS TRUE	
14899	061212	012737	000001 065640		MOV #1,\$MSGTY	:FOR APT
14900	061220	000137	047632		JMP EXIT	
14901	061224				END :OF IF ACTFLAG	
14902	061224				IF XXDPCHAIN IS TRUE AND \$ERTTL HI #20	
14903	061242				TYPE MSG066	:ERROR COUNT EXCEEDED 20 - ABORTING FOR XXDP CHAIN
14904	061246	013700	000042		MOV 42,R0	
14905	061252	005037	000042		CLR 42	
14906	061256	000137	015212		JMP \$ZAP42	
14907	061262				END :OF IF XXDPCHAIN	
14908	061262				END :OF IF DETFLAG	
14909	061262				ELSE	
14910	061264				SET HEADER	
14911	061272				END :OF IF NOERROR	
14912	061272			10\$:	CLEAR TOOMANY,NOERROR	
14913	061302	105737	061316		TSTB IBSAVE	:POWER FAIL ERROR CALL? :R-C
14914	061306	001402			BEQ 213\$	:R-C
14915	061310	000137	060536		JMP BACK	:JUMP IF SO :R-C
14916	061314	000002		213\$:	RTI	::RETURN
14917	061316	000000		IBSAVE:	.WORD 0	:R-C
14918	061320	000000		CPSAVE:	.WORD 0	:R-C
14919	061322	000000		CPERRF:	.WORD 0	:R-C
14920					.DSABL LSB	

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 468  
ROUTINE ERROR MESSAGE TYPEOUT

## .SBTTL ROUTINE ERROR MESSAGE TYPEOUT

14923  
14924  
14925  
14926  
14927  
14928  
14929  
14930 061324 104415  
14931 061326  
14932 061332 005000  
14933 061334 153700 002013  
14934 061340 001004  
14935  
14936 061342  
14937 061350 000511  
14938 061352 122700 000177  
14939 061356 001003  
14940 061360 012700 061634  
14941 061364 000406  
14942 061366 005300  
14943 061370 006300  
14944 061372 006300  
14945 061374 006300  
14946 061376 062700 066262  
14947 061402 012037 061440  
14948 061406 001417  
14949 061410 005737 002424  
14950 061414 001003  
14951 061416 005737 002576  
14952 061422 100011  
14953 061424 005737 002062  
14954 061430 001402  
14955 061432  
14956 061436  
14957 061440 000000  
14958 061442  
14959 061446 012037 061472  
14960 061452 001412  
14961 061454 005737 002424  
14962 061460 001003  
14963 061462 005737 002576  
14964 061466 100004  
14965 061470  
14966 061472 000000  
14967 061474  
14968 061500 012001  
14969 061502 001427  
14970 061504 012002

```

*****
*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
*****
$ERRTYP: SAVREG
      TYPE      $CRLF      :: "CARRIAGE RETURN" & "LINE FEED"
      CLR      R0          :: PICKUP THE ITEM INDEX
      BISB     $ITEMB,R0
      BNE      1$          :: IF ITEM NUMBER IS ZERO, JUST
                          :: TYPE THE PC OF THE ERROR
      TYPOCT   ERRPC,<ERROR ADDRESS>
      BR       11$         :: GET OUT
1$:    CMPB    #177,R0      :: POWER MONITOR CALL?                ;R-C
      BNE     100$         :: BRANCH IF NOT                      ;R-C
      MOV     #PFECWS,R0   :: MOV ADDRESS OF PFE BIT ERROR TO R0 ;R-C
      BR      110$        :: ADJUST THE INDEX SO THAT IT WILL    ;R-C
                          :: WORK FOR THE ERROR TABLE
100$:  DEC     R0
      ASL     R0
      ASL     R0
      ASL     R0
      ADD     # $ERRTB,R0  :: FORM TABLE POINTER
110$:  MOV     (R0)+,3$    :: PICKUP "ERROR MESSAGE" POINTER      ;R-C
      BEQ     4$          :: SKIP TYPEOUT IF NO POINTER
      TST     NOERROR     :: IS THIS REALLY AN ERROR?
      BNE     12$         :: YES - SKIP
      TST     HEADER      :: TYPE HEADER?
      BPL     4$          :: NO - SKIP
12$:   TST     FATAL$     :: WAS IT A FATAL ERROR?
      BEQ     2$          :: NO - SKIP
      TYPE    MSG067      :: FATAL
2$:    TYPE    :: TYPE THE "ERROR MESSAGE"
3$:    .WORD   0          :: "ERROR MESSAGE" POINTER GOES HERE
      TYPE    $CRLF      :: "CARRIAGE RETURN" & "LINE FEED"
4$:    MOV     (R0)+,5$    :: PICKUP "DATA HEADER" POINTER
      BEQ     6$          :: SKIP TYPEOUT IF 0
      TST     NOERROR     :: IS THIS REALLY AN ERROR?
      BNE     13$         :: YES - SKIP
      TST     HEADER      :: TYPE HEADER?
      BPL     6$          :: NO - SKIP
13$:   TYPE    :: TYPE THE "DATA HEAD"
5$:    .WORD   0          :: "DATA HEADER" POINTER GOES HERE
      TYPE    $CRLF      :: "CARRIAGE RETURN" & "LINE FEED"
6$:    MOV     (R0)+,R1    :: PICKUP "DATA TABLE" POINTER
      BEQ     10$         :: BR IF NO DATA TO BE TYPED
      MOV     (R0)+,R2    :: PICKUP "DATA FORMAT" POINTER

```

Address	Offset	Value	Label	Operation	Comment
14973	061506	112203		7\$: MOVB	(R2)+,R3
14974	061510	006303		ASL	R3
14975	061512	004773	061520	CALL	@8\$(R3)
14976	061516	000412		BR	9\$
14977	061520	061744		8\$: TAG70\$	
14978	061522	061754		TAG71\$	
14979	061524	061764		TAG72\$	
14980	061526	062034		TAG73\$	
14981	061530	062074		TAG74\$	
14982	061532	062106		TAG75\$	
14983	061534	062120		TAG76\$	
14984	061536	062164		TAG77\$	
14985	061540	062172		TAG78\$	
14986	061542	062252		TAG79\$	
14991	061544	062701	000002	9\$: ADD	#2,R1
14992	061550	005711		TST	(R1)
14993	061552	001403		BEQ	10\$
14994	061554			TYPE	MSG018
14995	061560	000752		BR	7\$
14996					
14997	061562	005737	002106	10\$: TST	MUT
14998	061566	001402		BEQ	11\$
14999	061570	005237	002576	INC	HEADER
15000	061574	104416		11\$: RESREG	
15001	061576			IF #SW7	SET.IN @SWR AND DETFLAG IS FALSE AND NOERROR IS FALSE
15002	061622	004737	062274	CALL	DETAIL
15003	061626			END ;OF	IF #SW7
15004	061626			TYPE	MSG104
15005	061632	000207		RETURN	
15006				.EVEN	
15007	061634	061644	061700	PFECWS: .WORD	PFECM,PFECDH,PFECDT,PFECDF
15008	061642	061740		PFECM: .ASCIZ	'POWER MONITOR BIT FOUND SET'
	061644	120	117		
	061647	105	122		
	061652	115	117		
	061655	111	124		
	061660	122	040		
	061663	111	124		
	061666	106	117		
	061671	116	104		
	061674	123	105		
	061677	000			
15009	061700	124	105	PFECDH: .ASCIZ	'TESTNO ERR PC CPUERR'
	061703	124	116		
	061706	040	040		
	061711	122	122		
	061714	120	103		
	061717	040	103		
	061722	125	105		
	061725	122	000		
15010				.EVEN	
15011	061730	065644	002016	PFECDT: .WORD	\$TESTN,ERRPC,CPSAVE,0
	061736	000000			
15012	061740	000	000	PFECDF: .BYTE	0,0,0,0
	061743	000			
15013					

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 472  
 ROUTINE ERROR MESSAGE TYPEOUT

```

15016 *****
15017 :*** OCTAL ***
15018 :*****
15019 TAG70$: TYPOCT @ (R1) ;:TYPE AN OCTAL NUMBER
15020 061744 000207 RETURN
15021
15022 :*****
15023 :*** DECIMAL ***
15024 :*****
15025 TAG71$: TYPDEC @ (R1) ;:TYPE A DECIMAL NUMBER
15026 061752 000207 RETURN
15027
15028 :*****
15029 :*** INTERLEAVE ***
15030 :*****
15031 TAG72$: PUSH R1,R5
15032 061770 013701 002100 MOV BANK,R1
15033 061774 070127 000004 MUL #4,R1
15034 062000 SET NOTAB ;INDICATE NO TABLE TO BE PRINTED - NOW
15035 062006 TYPE MSG014
15036 062012 004737 041726 CALL TCFIG1
15037 062016 005037 002366 CLR NOTAB
15038 062022 POP R5,R1
15039 062026 TYPE MSG014 ;1 SPACE
15040 062032 000207 RETURN
15041
15042 :*****
15043 :*** CSR ***
15044 :*****
15045 TAG73$: PUSH R1,R5
15046 062034 013701 002100 MOV BANK,R1
15047 062040 070127 000004 MUL #4,R1
15048 062050 SET NOTAB
15049 062056 004737 042222 CALL TCFIG3
15050 062062 005037 002366 CLR NOTAB
15051 062066 POP R5,R1
15052 062072 000207 RETURN
15053
15054 :*****
15055 :*** PATTERN ***
15056 :*****
15057 TAG74$: TYPOCS REALPAT,<TYPE (0-77)>,2,2
15058 062074 000207 RETURN
15059
15060 :*****
15061 :*** BANK ***
15062 :*****
15063 TAG75$: TYPOCS BANK,<TYPE (0-167)>,3
15064 062106 000207 RETURN

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 473  
 ROUTINE ERROR MESSAGE TYPEOUT

```

15066 ;*****
15067 ;*** MTYPE ***
15068 ;*****
15069 062120 TAG76$: PUSH R1,R5
15070 062124 013701 002100 MOV BANK,R1
15071 062130 070127 000004 MUL #4,R1
15072 062134 SET NOTAB
15073 062142 TYPE MSG019
15074 062146 004737 042066 CALL TCFIG2
15075 062152 005037 002366 CLR NOTAB
15076 062156 POP R5,R1
15077 062162 000207 RETURN
15078 ;*****
15079 ;*** UNKNOWN DATA ***
15080 ;*****
15081 TAG77$: TYPE MSG061
15082 062164 000207 RETURN
15083 062170
15084 ;*****
15085 ;*** PHYSICAL ADDRESS ***
15086 ;*****
15087 TAG78$: MOV ADDRESS,PHYADD
15088 062172 013737 002032 002036 SUB #FIRST,PHYADD
15089 062200 162737 060000 002036 MOV BANK,PHYADD+2
15090 062206 013737 002100 002040 ASR PHYADD+2
15091 062214 006237 002040 BCC 1$
15092 062220 103003 BIS #BIT15,PHYADD
15093 062222 052737 100000 002036 1$: MOV #PHYADD,-(SP) ;POINTER TO DOUBLE WORD ON STACK
15094 062230 012746 002036 CALL $DB20 ;CALL DOUBLE PRECISION CONVERSION ROUTINE
15095 062234 004737 065520 ADD #2,SP ;FIX STACK
15096 062240 062706 000002 TYPE $OCT8
15097 062244 RETURN
15098 062250 000207
15099 ;*****
15100 ;*** OCTAL BYTE ***
15101 ;*****
15102 TAG79$: TYPE MSG018 ;2 SPACES
15103 062252 TYPOCS @ (R1),<TYPE BYTE>,3,2
15104 062256 TYPE MSG014 ;SPACE
15105 062266 RETURN
15106 062272 000207

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 476  
 ROUTINE ERROR MESSAGE TYPEOUT

15150 062274

DETAIL: SUBTST <<SUBR DETAILED ERROR REPORT>>

\*\*\*\*\*  
 :SUBTEST SUBR DETAILED ERROR REPORT  
 \*\*\*\*\*

15151	062274	005237	002216		INC	DETFLAG
15152	062300	022737	000003	002216	CMP	#3,DETFLAG
15153	062306	101473			BLOS	4\$
15154	062310	022737	000002	002216	CMP	#2,DETFLAG
15155	062316	001435			BEQ	2\$
15156	062320				PUSH	HEADER,MUT
15157	062330				SET	HEADER
15158	062336	005037	002106		CLR	MUT
15159	062342	010037	002176		MOV	R0,DETRO
15160	062346	012700	002200		MOV	#DETR1,R0
15161	062352	010120			MOV	R1,(R0)+
15162	062354	010220			MOV	R2,(R0)+
15163	062356	010320			MOV	R3,(R0)+
15164	062360	010420			MOV	R4,(R0)+
15165	062362	010520			MOV	R5,(R0)+
15166	062364	013720	002022		MOV	ERRSP,(R0)+
15167	062370	013720	002026		MOV	ERRPSW,(R0)+
15168	062374	013700	002176		MOV	DETRO,R0
15169	062400				SET	NOERROR
15170	062406	104013			ERROR	+13
15171	062410	000423			BR	1\$
15172	062412			2\$:	PUSH	HEADER,MUT
15173	062422				SET	HEADER
15174	062430	005037	002106		CLR	MUT
15175	062434				SET	NOERROR
15176	062442	104031			ERROR	+31
15177	062444	022737	000001	003752	CMP	#1,PROTYP
15178	062452	001002			BNE	1\$
15179	062454	005037	177766		CLR	CPUERR
15180	062460			1\$:	POP	MUT,HEADER
15181					;WARNING	RECURSIVE
15182	062470	004737	062274		CALL	DETAIL
15183	062474	000207			RETURN	

;IS THIS AN 11/44?

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 478  
 SUBR DETAILED ERROR REPORT

```

15186      ;SIMULATE CONTROL 'T'
15187 062476 004737 064144      4$: CALL CONTT      ;DISPLAY 'DISPLAY' INFO
15188
15189      ;TYPE CONTENTS OF ALL CSR'S
15190 062502      PUSH CSR,CSRNO,R1
15191 062514      TYPE MSG058
15192 062520      TYPE $CRLF
15193 062524 013701 002222      MOV TOTCSRS,R1
15194 062530      BEGIN DUMPCSRLOOP
15195 062530      FOR CSRNO := #0 TO #36 BY #2
15196 062534 006301      ASL R1
15197 062536      ON.ERROR
15198 062540 104426      READCSR
15199 062542      TYP OCT CSR
15200 062550      TYPE MSG018      ;2 SPACES
15201 062554      END ;OF ON.ERROR
15202 062554      IF R1 EQ #0 THEN LEAVE DUMPCSRLOOP
15203 062560      END ;OF FOR CSRNO
15204 062576      END DUMPCSRLOOP
15205 062576      POP R1,CSRNO,CSR
15206
15207      ;TYPE STACKS
15208 062610      PUSH R0,R1
15209 062614      TYPE MSG088      ;KERNEL STACK
15210 062620 013701 002560      MOV KSTACK,R1
15211 062624 162701 000002      SUB #2,R1
15212 062630      FOR R0 := SP TO R1 BY #2
15213 062632      TYPE $CRLF
15214 062636      TYP OCT R0
15215 062642      TYPE MSG018      ;2 SPACES
15216 062646      TYP OCT (R0)
15217 062652      END ;OF FOR R0
15218      ;SET PREVIOUS MODE TO SUPERVISOR
15219 062662 005737 002452      TST NOSUPER
15220 062666 001036      BNE DET1
15221 062670 042737 030000 177776      BIC #BIT13!BIT12,PSW
15222 062676 052737 010000 177776      BIS #BIT12,PSW
15223 062704 006506      MFPI SSP
15224 062706      POP R1,R0
15225 062712      TYPE MSG089      ;SUPERVISOR STACK
15226 062716      IF R0 LT #SUPSTK
15227 062724      FOR R0 := R0 TO #SUPSTK-2 BY #2
15228 062724      TYPE $CRLF
15229 062730      TYP OCT R0
15230 062734      TYPE MSG018      ;2 SPACES
15231 062740      TYP OCT (R0)
15232 062744      END ;OF FOR R0
15233 062756      ELSE
15234 062760      TYPE MSG091      ;IS EMPTY
15235 062764      END ;OF IF R0
15236      ;SET PREVIOUS MODE TO USER
15237 062764 052737 030000 177776 DET1:      BIS #BIT13!BIT12,PSW
15238 062772 006506      MFPI USP
15239 062774      POP R0
15240 062776      TYPE MSG090      ;USER STACK
15241 063002      IF R0 LT #USESTK
15242 063010      FOR R0 := R0 TO #USESTK-2 BY #2

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 478-1  
SUBR DETAILED ERROR REPORT

15243	063010		TYPE	\$CRLF	
15244	063014		TYPOCT	RO	
15245	063020		TYPE	MSG018	
15246	063024		TYPOCT	(RO)	:2 SPACES
15247	063030		END ;OF FOR RO		
15248	063042		ELSE		
15249	063044		TYPE MSG091		;IS EMPTY
15250	063050		END ;OF IF RO		
15251	063050		TYPE \$CRLF		
15252	063054	005037 002216	CLR DETFLAG		
15253	063060		POP RO		
15254	063062	000207	RETURN		



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 481  
 ROUTINE BINARY TO OCTAL (ASCII) AND TYPE

.SBTTL ROUTINE BINARY TO OCTAL (ASCII) AND TYPE

```

15292
15293
15294
15295
15296
15297
15298
15299
15300
15301
15302
15303
15304
15305
15306
15307
15308
15309
15310
15311
15312
15313
15314
15315
15316
15317 063064 017646 000000
15318 063070 116637 000001 063307
15319 063076 112637 063311
15320 063102 062716 000002
15321 063106 000406
15322 063110 112737 000001 063307
15323 063116 112737 000006 063311
15324 063124 112737 000005 063306
15325 063132 010346
15326 063134 010446
15327 063136 010546
15328 063140 113704 063311
15329 063144 005404
15330 063146 062704 000006
15331 063152 110437 063310
15332 063156 113704 063307
15333 063162 016605 000012
15334 063166 005003
15335 063170 006105
15336 063172 000404
15337 063174 006105
15338 063176 006105
15339 063200 006105
15340 063202 010503
15341 063204 006103
15342 063206 105337 063310
15343 063212 100016
15344 063214 042703 177770
15345 063220 001002
15346 063222 005704
15347 063224 001403
15348 063226 005204

*****
*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
*OCTAL (ASCII) NUMBER AND TYPE IT.
*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPOS      ;;CALL FOR TYPEOUT
*      .BYTE     N      ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
*      .BYTE     M      ;;M=1 OR 0
*                          ;;1=TYPE LEADING ZEROS
*                          ;;0=SUPPRESS LEADING ZEROS
*$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
*$TYPOS OR $TYPOC
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPON      ;;CALL FOR TYPEOUT
*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPOC      ;;CALL FOR TYPEOUT
*$TYPOS: MOV      @ (SP),-(SP)      ;;PICKUP THE MODE
        MOV      1(SP), $OFILL      ;;LOAD ZERO FILL SWITCH
        MOV      (SP)+, $OMODE+1      ;;NUMBER OF DIGITS TO TYPE
        ADD      #2, (SP)      ;;ADJUST RETURN ADDRESS
        BR      $TYPON
*$TYPOC: MOV      #1, $OFILL      ;;SET THE ZERO FILL SWITCH
        MOV      #6, $OMODE+1      ;;SET FOR SIX(6) DIGITS
*$TYPON: MOV      #5, $OCNT      ;;SET THE ITERATION COUNT
        MOV      R3, -(SP)      ;;SAVE R3
        MOV      R4, -(SP)      ;;SAVE R4
        MOV      R5, -(SP)      ;;SAVE R5
        MOV      $OMODE+1, R4      ;;GET THE NUMBER OF DIGITS TO TYPE
        NEG      R4
        ADD      #6, R4      ;;SUBTRACT IT FOR MAX. ALLOWED
        MOV      R4, $OMODE      ;;SAVE IT FOR USE
        MOV      $OFILL, R4      ;;GET THE ZERO FILL SWITCH
        MOV      12(SP), R5      ;;PICKUP THE INPUT NUMBER
        CLR      R3      ;;CLEAR THE OUTPUT WORD
        ROL      R5      ;;ROTATE MSB INTO 'C'
        BR      3$      ;;GO DO MSB
        PCL      R5      ;;FORM THIS DIGIT
        ROL      R5
        ROL      R5
        MOV      R5, R3
        ROL      R3      ;;GET LSB OF THIS DIGIT
        DECB      $OMODE      ;;TYPE THIS DIGIT?
        BPL      6$      ;;BR IF NO
        BIC      #177770, R3      ;;GET RID OF JUNK
        BNE      4$      ;;TEST FOR 0
        TST      R4      ;;SUPPRESS THIS 0?
        BEQ      5$      ;;BR IF YES
        INC      R4      ;;DON'T SUPPRESS ANYMORE 0'S
        1$:
        2$:
        3$:
        4$:
        5$:

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 481-1  
 ROUTINE BINARY TO OCTAL (ASCII) AND TYPE

15349	063230	052703	000060		BIS	#'0,R3	::MAKE THIS DIGIT ASCII
15350	063234	052703	000040	5\$:	BIS	#',R3	::MAKE ASCII IF NOT ALREADY
15351	063240	110337	063304		MOVB	R3,8\$	::SAVE FOR TYPING
15352	063244				TYPE	8\$	::GO TYPE THIS DIGIT
15353	063250	105337	063306	6\$:	DECB	\$OCNT	::COUNT BY 1
15354	063254	003347			BGT	2\$	::BR IF MORE TO DO
15355	063256	002402			BLT	7\$	::BR IF DONE
15356	063260	005204			INC	R4	::INSURE LAST DIGIT ISN'T A BLANK
15357	063262	000744			BR	2\$	::GO DO THE LAST DIGIT
15358	063264	012605		7\$:	MOV	(SP)+,R5	::RESTORE R5
15359	063266	012604			MOV	(SP)+,R4	::RESTORE R4
15360	063270	012603			MOV	(SP)+,R3	::RESTORE R3
15361	063272	016666	000002 000004		MOV	2(SP),4(SP)	::SET THE STACK FOR RETURNING
15362	063300	012616			MOV	(SP)+,(SP)	
15363	063302	000002			RTI		::RETURN
15364	063304	000		8\$:	.BYTE	0	::STORAGE FOR ASCII DIGIT
15365	063305	000			.BYTE	0	::TERMINATOR FOR TYPE ROUTINE
15366	063306	000		\$OCNT:	.BYTE	0	::OCTAL DIGIT COUNTER
15367	063307	000		\$OFILL:	.BYTE	0	::ZERO FILL SWITCH
15368	063310	000000		\$OMODE:	.WORD	0	::NUMBER OF DIGITS TO TYPE

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 482  
 ROUTINE CONVERT BINARY TO DECIMAL AND TYPE

```

15370 .SBTTL ROUTINE CONVERT BINARY TO DECIMAL AND TYPE
15371 *****
15372 *THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
15373 *SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
15374 *NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
15375 *BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
15376 *REPLACED WITH SPACES.
15377 *CALL:
15378 *
15379 *      MOV      NUM,-(SP)      ;;PUT THE BINARY NUMBER ON THE STACK
15380 *      TYPDS    ;;GO TO THE ROUTINE
15381 063312 012746 020200 $TYPDS: PUSH R0,R1,R2,R3,R5
15382 063324 016605 000020      MOV #20200,-(SP)      ;;SET BLANK SWITCH AND SIGN
15383 063334 100004      MOV 20(SP),R5      ;;GET THE INPUT NUMBER
15384 063336 005405      BPL 1$      ;;BR IF INPUT IS POS.
15385 063340 112766 000055 000001      NEG R5      ;;MAKE THE BINARY NUMBER POS.
15386 063346 005000      MOVB #'-,1(SP)      ;;MAKE THE ASCII NUMBER NEG.
15387 063350 012703 063526 1$: CLR R0      ;;ZERO THE CONSTANTS INDEX
15388 063354 112723 000040      MOV #SDBLK,R3      ;;SETUP THE OUTPUT POINTER
15389 063360 005002      MOVB #' ,(R3)+      ;;SET THE FIRST CHARACTER TO A BLANK
15390 063362 016001 063516 2$: CLR R2      ;;CLEAR THE BCD NUMBER
15391 063366 160105      MOV $DTBL(R0),R1      ;;GET THE CONSTANT
15392 063370 002402      SUB R1,R5      ;;FORM THIS BCD DIGIT
15393 063372 005202      BLT 4$      ;;BR IF DONE
15394 063374 000774      INC R2      ;;INCREASE THE BCD DIGIT BY 1
15395 063376 060105      BR 3$
15396 063400 005702      ADD R1,R5      ;;ADD BACK THE CONSTANT
15397 063402 001002      TST R2      ;;CHECK IF BCD DIGIT=0
15398 063404 105716      BNE 5$      ;;FALL THROUGH IF 0
15399 063406 100407      TSTB (SP)      ;;STILL DOING LEADING 0'S?
15400 063410 106316      BMI 7$      ;;BR IF YES
15401 063412 103003      ASLB (SP)      ;;MSD?
15402 063414 116663 000001 177777 5$: BCC 6$      ;;BR IF NO
15403 063422 052702 000060      MOVB 1(SP),-1(R3)      ;;YES--SET THE SIGN
15404 063426 052702 000040 6$: BIS #'0,R2      ;;MAKE THE BCD DIGIT ASCII
15405 063432 110223      BIS #' ,R2      ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
15406 063434 005720      MOVB R2,(R3)+      ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
15407 063436 020027 000010      TST (R0)+      ;;JUST INCREMENTING
15408 063442 002746      CMP R0,#10      ;;CHECK THE TABLE INDEX
15409 063444 003002      BLT 2$      ;;GO DO THE NEXT DIGIT
15410 063446 010502      BGT 8$      ;;GO TO EXIT
15411 063450 000764      MOV R5,R2      ;;GET THE LSD
15412 063452 105726 8$: BR 6$      ;;GO CHANGE TO ASCII
15413 063454 100003      TSTB (SP)+      ;;WAS THE LSD THE FIRST NON-ZERO?
15414 063456 116663 177777 177776 9$: BPL 9$      ;;BR IF NO
15415 063464 105013      MOVB -1(SP),-2(R3)      ;;YES--SET THE SIGN FOR TYPING
15416 063466      CLRB (R3)      ;;SET THE TERMINATOR
15417 063500      POP R5,R3,R2,R1,R0      ;;NOW TYPE THE NUMBER
15418 063504 016666 000002 000004      TYPE $DBLK      ;;ADJUST THE STACK
15419 063512 012616      MOV 2(SP),4(SP)
15420 063514 000002      MOV (SP)+,(SP)
15421 063516 023420      RTI      ;;RETURN TO USER
15422 063520 001750      $DTBL: 10000.
15423 063522 000144      1000.
15424 063524 000012      100.
15425 063526 000000 000000 000000 $DBLK: 10.
      063534 000000      .WORD 0,0,0,0

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 483

ROUTINE TTY INPUT

```

15427
15428
15429
15430
15431
15432
15433
15434 063536
15440 063536 005737 056276
15441 063542 001406
15442 063544 013746 056276
15443 063550 005037 056276
15444 063554 000137 063576
15445 063560 105777 117042
15446 063564 100130
15447 063566 117746 117036
15448 063572 042716 177600
15449 063576 022716 000006
15450 063602 001002
15451 063604 004737 050102
15452 063610 022716 000024
15453 063614 001002
15454 063616 004737 064144
15455 063622 022716 000003
15456 063626 001454
15457 063630 022716 000023
15458 063634 001002
15459 063636 004737 064220
15460 063642 022716 000013
15461 063646 001005
15462 063650
15463 063654 013706 002144
15464 063660 000207
15465 063662 022737 000176 002622 6$:
15466 063670 001067
15467 063672 022716 000007
15468 063676 001064
15469 063700 005737 002060
15470 063704 001061
15471 063706
15472 063712
15473 063716
15474 063724
15475 063730 005046
15476 063732 005046
15477 063734 105777 116666
15478 063740 100375
15479 063742 117746 116662
15480 063746 042716 177600
15481 063752 021627 000003
15482 063756 001006
15483 063760
15484 063764 062706 000006
15485 063770 000137 047524
15486 063774 021627 000025
15487 064000 001005
15488 064002

.SBTTL ROUTINE TTY INPUT
:*****
:*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
:*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
:*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
:*WHEN OPERATING IN TTY FLAG MODE.
.ENABLE LSB
$CKSWR: TST XOCHAR ;;SOMETHING THERE?
        BEQ NOCH ;; GO ON IF NOT
        MOV XOCHAR,-(SP) ;; USE IT
        CLR XOCHAR
        JMP CONTS1
NOCH: TSTB @STKS ;;CHAR THERE?
      BPL 12$ ;;IF NO, DON'T WAIT AROUND
      MOVB @STKB,-(SP) ;;SAVE THE CHAR
      BIC #^C177,(SP) ;;STRIP-OFF THE ASCII
CONTS1: CMP #6,(SP) ;;IS IT CONTROL F?
        BNE 1$ ;;NO SKIP
        CALL FIELDSERVICE
1$: CMP #24,(SP) ;;IS IT CONTROL T?
     BNE 16$ ;;NO - SKIP
     CALL CONTT ;;YES - CALL CONTROL T ROUTINE
16$: CMP #3,(SP) ;;IS IT CONTROL C?
     BEQ 5$ ;;YES EXIT *****NOTE***** STACK IS SCREWED UP!
2$: CMP #23,(SP) ;;IS IT CONTROL S?
     BNE 17$ ;;NO - SKIP
     CALL CONTS ;;YES - CALL CONTROL S ROUTINE
17$: CMP #13,(SP) ;;IS IT CONTROL K?
     BNE 6$ ;;NO - SKIP
     TYPE $CNTLK ;;TYPE A ^K
     MOV CTLKVEC,SP ;;RESET KSP TO AFTER PATTERN EXEC ROUTINE
     RETURN ;;RETURN TO PATTERN EXEC ROUTINE
6$: CMP #SWREG,SWR ;;IS THE SOFT-SWR SELECTED?
     BNE CKEND ;;BRANCH IF NO
     CMP #7,(SP) ;;IS IT A CONTROL G?
     BNE CKEND ;;NO, RETURN TO USER
     TST $AUTO ;;ARE WE RUNNING IN AUTO-MODE?
     BNE CKEND ;;BRANCH IF YES
     TYPE $CNTLG ;;ECHO THE CONTROL-G (^G)
SGTSWR: TYPE $MSWR ;;TYPE CURRENT CONTENTS
        TYPOCT @SWR ;;OF THE SWR
        TYPE $MNEW ;;PROMPT FOR NEW SWR
3$: CLR -(SP) ;;CLEAR COUNTER
     CLR -(SP) ;;THE NEW SWR
4$: TSTB @STKS ;;CHAR THERE?
     BPL 4$ ;;IF NOT TRY AGAIN
     MOVB @STKB,-(SP) ;;PICK UP CHAR
     BIC #^C177,(SP) ;;MAKE IT 7-BIT ASCII
     CMP (SP),#3 ;;IS IT A CONTROL-C?
     BNE 7$ ;;BRANCH IF NOT
5$: TYPE $CNTLC ;;YES, ECHO CONTROL-C (^C)
     ADD #6,SP ;;CLEAN UP STACK
     JMP BOOT ;;CONTROL-C RESTART
7$: CMP (SP),#25 ;;IS IT A CONTROL-U?
     BNE 9$ ;;BRANCH IF NOT
     TYPE $CNTLU ;;YES, ECHO CONTROL-U (^U)

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 483-1  
ROUTINE TTY INPUT

15489	064006	062706	000006	8\$:	ADD	#6,SP	::IGNORE PREVIOUS INPUT
15490	064012	000746			BR	3\$	::LET'S TRY IT AGAIN
15491	064014	021627	000015	9\$:	CMP	(SP),#15	::IS IT A <CR>?
15492	064020	001016			BNE	13\$	::BRANCH IF NO
15493	064022	005766	000004		TST	4(SP)	::YES, IS IT THE FIRST CHAR?
15494	064026	001403			BEQ	10\$	::BRANCH IF YES
15495	064030	016677	000002	116564	MOV	2(SP),@SWR	::SAVE NEW SWR
15496	064036	062706	000006	10\$:	ADD	#6,SP	::CLEAR UP STACK
15497	064042				TYPE	\$CRLF	::ECHO <CR> AND <LF>
15498	064046	000002		12\$:	RTI		::RETURN
15499	064050	062706	000002	CKEND:	ADD	#2,SP	::FIX STACK
15500	064054	000002			RTI		::RETURN
15501	064056	004737	056300	13\$:	CALL	\$TYPEC	::ECHO CHAR
15502	064062	021627	000060		CMP	(SP),#60	::CHAR < 0?
15503	064066	002420			BLT	15\$	::BRANCH IF YES
15504	064070	021627	000067		CMP	(SP),#67	::CHAR > 7?
15505	064074	003015			BGT	15\$	::BRANCH IF YES
15506	064076	042726	000060		BIC	#60,(SP)+	::STRIP-OFF ASCII
15507	064102	005766	000002		TST	2(SP)	::IS THIS THE FIRST CHAR
15508	064106	001403			BEQ	14\$	::BRANCH IF YES
15509	064110	006316			ASL	(SP)	::NO, SHIFT PRESENT
15510	064112	006316			ASL	(SP)	::CHAR OVER TO MAKE
15511	064114	006316			ASL	(SP)	::ROOM FOR NEW ONE.
15512	064116	005266	000002	14\$:	INC	2(SP)	::KEEP COUNT OF CHAR
15513	064122	056616	177776		BIS	-2(SP),(SP)	::SET IN NEW CHAR
15514	064126	000702			BR	4\$	::GET THE NEXT ONE
15515	064130			15\$:	TYPE	\$QUES	::TYPE ?<CR><LF>
15516	064134	000724			BR	8\$	::SIMULATE CONTROL-U
15517	064136	136	113	015	\$CNTLK:	.ASCIIZ	::CONTROL K ASCII STRING
	064141	012	000				
15518					.EVEN		
15519					.DSABL	LSB	

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 485  
ROUTINE TTY INPUT

15522 064144

CONTT: SUBTST &lt;&lt;CONTROL T&gt;&gt;

\*\*\*\*\*  
:SUBTEST CONTROL T  
\*\*\*\*\*

15523 064144

PUSH R0

15524 064146

TYPE \$CRLF

15534 064152

IF RLFLAG IS TRUE

15535 064160

TYPE MSGC92

;RELOCATED

15536 064164

END ;OF IF RLFLAG

15537 064164

TYPE MSG093

;BANK=

15538 064170

TYPOCS FANK,,3

;TYPE 3 DIGITS

15539 064200

TYPE MSG095

;PAT=

15540 064204

TYPOCS REAL, 2

;TYPE 2 DIGITS

15544 064214

POP R0

15545 064216

RETURN

000207

15546

15547 064220

CONTS: SUBTST &lt;&lt;CONTROL S &amp; CONTROL Q&gt;&gt;

\*\*\*\*\*  
:SUBTEST CONTROL S & CONTROL Q  
\*\*\*\*\*

15548 064220

POP R0

;GET RID OF RETURN ADDRESS FROM STACK

15549 064222

105777

116400

CONTS2: TSTB

@STKS

;WAIT FOR CHARACTER

15550 064226

100375

BPL CONTS2

15551 064230

117716

116374

MOVB @STKB,(SP)

;REPLACE OVER OLD CHARACTER ON STACK

15552 064234

042716

177600

BIC #^C177,(SP)

;STRIP ALL BUT ASCII

15553 064240

IF (SP) EQ #21

;IF IT IS A CONTROL Q

15554 064246

000137

063576

JMP CONTS1

15555 064252

ELSE

15556 064254

000762

BR CONTS2

15557 064256

END ;OF IF (SP)

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 486  
CONTROL S & CONTROL Q

```

15559 *****
15560 *THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
15561 *CALL:
15562 *      RDCHR                      ::INPUT A SINGLE CHARACTER FROM THE TTY
15563 *      RETURN HERE                ::CHARACTER IS ON THE STACK
15564 *                                  ::WITH PARITY BIT STRIPPED OFF
15565
15566
15567 064256 011646 $RDCHR: MOV      (SP),-(SP)      ::PUSH DOWN THE PC
15568 064260 016666 000004 000002 1$: MOV      4(SP),2(SP)  ::SAVE THE PS
15569 064266 105777 116334 1$: TSTB      @STKS      ::WAIT FOR
15570 064272 100375 116330 000004 BPL      1$          ::A CHARACTER
15571 064274 117766 177600 000004 MOVB     @STKB,4(SP)  ::READ THE TTY
15572 064302 042766 000004 000023 BIC      #'C<177>,4(SP) ::GET RID OF JUNK IF ANY
15573 064310 026627 000004 000023 CMP      4(SP),#23    ::IS IT A CONTROL-S?
15574 064316 001013 116302 BNE      3$          ::BRANCH IF NO
15575 064320 105777 116302 2$: TSTB      @STKS      ::WAIT FOR A CHARACTER
15576 064324 100375 116276 BPL      2$          ::LOOP UNTIL ITS THERE
15577 064326 117746 177600 000021 MOVB     @STKB,-(SP)  ::GET CHARACTER
15578 064332 042716 000021 BIC      #'C177,(SP)  ::MAKE IT 7-BIT ASCII
15579 064336 022627 000021 CMP      (SP)+,#21    ::IS IT A CONTROL-Q?
15580 064342 001366 000021 BNE      2$          ::IF NOT DISCARD IT
15581 064344 000750 000021 BR       1$          ::YES, RESUME
15582 064346 026627 000004 000021 3$: CMP      4(SP),#21    ::IS IT A RANDOM CONTROL-Q?      :R-C
15583 064354 001744 000004 000140 BEQ      1$          ::BRANCH BACK IF SO      :R-C
15584 064356 026627 000004 000140 CMP      4(SP),#140   ::IS IT UPPER CASE?
15585 064364 002407 000004 000175 BLT      4$          ::BRANCH IF YES
15586 064366 026627 000004 000175 CMP      4(SP),#175   ::IS IT A SPECIAL CHAR?
15587 064374 003003 000040 000004 BGT      4$          ::BRANCH IF YES
15588 064376 042766 000040 000004 BIC      #40,4(SP)   ::MAKE IT UPPER CASE
15589 064404 000002 4$: RTI                      ::GO BACK TO USER
15590 *****
15591 *THIS ROUTINE WILL INPUT A STRING FROM THE TTY
15592 *CALL:
15593 *      RDLIN                      ::INPUT A STRING FROM THE TTY
15594 *      RETURN HERE                ::ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
15595 *                                  ::TERMINATOR WILL BE A BYTE OF ALL 0'S
15596 064406 010346 $RDLIN: MOV      R3,-(SP)      ::SAVE R3
15597 064410 005046 CLR      -(SP)      ::CLEAR THE RUBOUT KEY
15598 064412 012703 064704 1$: MOV      #STTYIN,R3    ::GET ADDRESS
15599 064416 022703 064730 2$: CMP      #STTYIN+20.,R3 ::BUFFER FULL?
15600 064422 101477 8$ BLOS      8$          ::BR IF YES
15601 064424 104411 RDCHR      ::GO READ ONE CHARACTER FROM THE TTY
15602 064426 112613 MOVB     (SP)+,(R3)  ::GET CHARACTER
15603 064430 122713 000003 CMPB      #3,(R3)    ::IS IT A CONTROL-C?
15604 064434 001016 BNE      3$          ::BRANCH IF NO
15605 064436 TYPE      $CNTLC      ::TYPE A CONTROL-C (^C)
15606 064442 005726 TST      (SP)+      ::CLEAN RUBOUT KEY OFF OF THE STACK
15607 064444 012603 MOV      (SP)+,R3    ::RESTORE R3
15608 064446 032777 000400 116146 BIT      #BIT8,@SWR  ::IS THERE A HALT FLAG SET IN THE SWR?
15609 064454 001404 BEQ      11$         ::BRANCH IF NOT TO BOOT ROUTINE
15610 064456 005037 002414 CLR      STOPOK    ::GET READY TO HALT PROGRAM
15611 064462 000137 047632 JMP      EXIT      ::GO HALT PROGRAM
15612 064466 000137 047524 11$: JMP      BOOT      ::GOTO CONTROL-C RESTART
15613 064472 122713 000177 3$: CMPB      #177,(R3)   ::IS IT A RUBOUT
15614 064476 001022 BNE      5$          ::BR IF NO
15615 064500 005716 TST      (SP)      ::IS THIS THE FIRST RUBOUT?

```

CZMSPA0 PS11- /M/P MC DRY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 486-1  
CONTROL S & C CONTROL Q

15616	064502	001007			BNE	4\$	::BR IF NO
15617	064504	112737	000134	064702	MOVB	#'\,10\$	::TYPE A BACK SLASH
15618	064512				TYPE	10\$	
15619	064516	012716	177777		MOV	#-1,(SP)	::SET THE RUBOUT KEY
15620	064522	00530		4\$:	DEC	R3	::BACKUP BY ONE
15621	064524	020327	064704		CMP	R3,#\$TTYIN	::STACK EMP'Y?
15622	064530	103434			BLO	8\$	::BR IF YES
15623	064532	1.1337	064702		MOVB	(R3),10\$	::SETUP TO TYPEOUT THE DELETED CHAR.
15624	064536				TYPE	10\$	::GO TYPE
15625	064542	000775			BR	2\$	::GO READ ANOTHER CHAR.
15626	064544	005716		5\$:	TST	(SP)	::RUBOUT KEY SET?
15627	064546	001406			RFQ	6\$	::BR IF NO
15628	064550	112737	000134	064702	MOVB	#'\,10\$	::TYPE A BACK SLASH
15629	064556				TYPE	10\$	
15630	064562	005016			CLR	(SP)	::CLEAR THE RUBOUT KEY
15631	064564	122713	000025	6\$:	CMPB	#25,(R3)	::IS CHARACTER A CTRL U?
15632	064570	001003			BNE	7\$	::BR IF NO
15633	064572				TYPE	\$CNTLU	::TYPE A CONTROL 'U'
15634	064576	000705			BR	1\$	::GO START OVER
15635	064600	122713	000022	7\$:	CMPB	#26,(R3)	::IS CHARACTER A 'R'?
15636	064604	001011			BNE	9\$	::BRANCH IF NO
15637	064606	105013			CLRB	(R3)	::CLEAR THE CHARACTER
15638	064610				TYPE	\$CRLF	::TYPE A 'CR' & 'LF'
15639	064614				TYPE	\$TTYIN	::TYPE THE INPUT STRING
15640	064620	000676			BR	2\$	::GO PICKUP ANOTHER CHARACTER
15641	064622			8\$:	TYPE	\$QUES	::TYPE A '?'
15642	064624	000771			BR	1\$	::CLEAR THE BUFFER AND LOOP
15643	064630	111557	064702	9\$:	MOVB	(R3),10\$	::ECHO THE CHARACTER
15644	064634				TYPE	10\$	
15645	064640	122723	000015		CMPB	#15,(R3)+	::CHECK FOR RETURN
15646	064644	001264			BNE	2\$	::LOOP IF NOT RETURN
15647	064646	105063	177777		CLRB	-1(R3)	::CLEAR RETURN (THE 15)
15648	064652				TYPE	\$LF	::TYPE A LINE FEED
15649	064656	005726			MOV	(SP)+	::CLEAN RUBOUT KEY FROM THE STACK
15650	064660	012603			MOV	(SP)+,R3	::RESTORE R3
15651	064662	011646			MOV	(SP)-,(SP)	::ADJUST THE STACK AND PUT ADDRESS OF THE
15652	064664	016666	000004	000002	MOV	4(SP),2(SP)	::FIRST ASCII CHARACTER ON IT
15653	064672	012766	064704	000004	MOV	#\$TTYIN,4(SP)	
15654	064700	000002			RTI		::RETURN
15655	064702	000		10\$:	.BYTE	0	::STORAGE FOR ASCII CHAR. TO TYPE
15656	064703	000			.BYTE	0	::TERMINATOR
15657	064704	000024			\$TTYIN:	.REPT 20	::RESERVE SIZE BYTES FOR TTY INPUT
15660	064730	136	103	015	\$CNTLC:	.ASCIIZ /'^C/<15><12>	::CONTROL 'C'
	064733	012	000				
15661	064735	136	125	015	\$CNTLU:	.ASCIIZ /'^U/<15><12>	::CONTROL 'U'
	064740	012	000				
15662	064742	136	107	015	\$CNTLG:	.ASCIIZ /'^G/<15><12>	::CONTROL 'G'
	064745	012	000				
15663	064747	015	012	123	\$MSWR:	.ASCIIZ <15><12>/SWR = /	
	064752	127	122	040			
	064755	075	040	000			
15664	064760	040	040	116	\$MNEW:	.ASCIIZ / NEW = /	
	064763	105	127	040			
	064766	075	040	000			
15665					.EVEN		



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 487  
 ROUTINE READ AN OCTAL NUMBER FROM THE TTY

```

15667 .SBTTL ROUTINE READ AN OCTAL NUMBER FROM THE TTY
15668 *****
15669 ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
15670 ;*CHANGE IT TO BINARY.
15671 ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
15672 ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
15673 ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
15674 ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
15675 ;*CALL:
15676 ;*      RDOCT          ;;READ AN OCTAL NUMBER
15677 ;*      RETURN HERE    ;;LOW ORDER BITS ARE ON TOP OF THE STACK
15678 ;*                    ;;HIGH ORDER BITS ARE IN $HIOCT
15679 064772 011646 $RDOCT: MOV (SP),-(SP) ;;PROVIDE SPACE FOR THE
15680 064774 016666 000004 000002 MOV 4(SP),2(SP) ;;INPUT NUMBER
15681 065002 PUSH R0,R1,R2
15682 065010 104412 1$: RDLIN ;;READ AN ASCII LINE
15683 065012 012600 MOV (SP)+,R0 ;;GET ADDRESS OF 1ST CHARACTER
15684 065014 010037 065120 MOV R0,$$ ;;AND SAVE IT
15685 065020 005001 CLR R1 ;;CLEAR DATA WORD
15686 065022 005002 CLR R2
15687 065024 112046 2$: MOVB (R0)+,-(SP) ;;PICKUP THIS CHARACTER
15688 065026 001420 BEQ 3$ ;;IF ZERO GET OUT
15689 065030 122716 000060 CMPB #'0,(SP) ;;MAKE SURE THIS CHARACTER
15690 065034 003026 BGT 4$ ;;IS AN OCTAL DIGIT
15691 065036 122716 000067 CMPB #'7,(SP)
15692 065042 002423 BLT 4$
15693 065044 006301 ASL R1 ;;*2
15694 065046 006102 ROL R2
15695 065050 006301 ASL R1 ;;*4
15696 065052 006102 ROL R2
15697 065054 006301 ASL R1 ;;*8
15698 065056 006102 ROL R2
15699 065060 042716 177770 BIC #'^C7,(SP) ;;STRIP THE ASCII JUNK
15700 065064 062601 ADD (SP)+,R1 ;;ADD IN THIS DIGIT
15701 065066 000756 BR 2$ ;;LOOP
15702 065070 005726 3$: TST (SP)+ ;;CLEAN TERMINATOR FROM STACK
15703 065072 010166 000012 MOV R1,12(SP) ;;SAVE THE RESULT
15704 065076 010237 065140 MOV R2,$HIOCT
15705 065102 POP R2,R1,R0
15706 065110 000002 RTI ;;RETURN
15707 065112 005726 4$: TST (SP)+ ;;CLEAN PARTIAL FROM STACK
15708 065114 105010 CLRB (R0) ;;SET A TERMINATOR
15709 065116 TYPE ;;TYPE UP THRU THE BAD CHAR.
15710 065120 000000 5$: .WORD 0
15711 065122 TYPE MSG062 ;;INPUT MUST BE A
15712 065126 TYPE MSG063 ;N OCTAL
15713 065132 TYPE MSG064 ;NUMBER
15714 065136 000724 BR 1$ ;;TRY AGAIN
15715 065140 000000 $HIOCT: .WORD 0 ;;HIGH ORDER BITS GO HERE
15716 .SBTTL ROUTINE READ A DECIMAL NUMBER FROM THE TTY
15717 *****
15718 ;*THIS ROUTINE WILL READ A DECIMAL (ASCII) NUMBER FROM THE TTY AND
15719 ;*CHANGE IT TO BINARY. IF TOO MANY CHARACTERS OR ANY ILLEGAL CHARACTERS
15720 ;*ARE READ A "?" FOLLOWED BY A CARRIAGE RETURN-LINE FEED WILL BE TYPED.
15721 ;*THE COMPLETE NUMBER MUST BE RETYPED. THE INPUT IS TERMINATED BY THE
15722 ;*USER TYPING A CARRIAGE RETURN. THE RANGE OF THE INPUT NUMBER IS
15723

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 487-1  
ROUTINE READ A DECIMAL NUMBER FROM THE TTY

```

15724      ;*POSITIVE 32767 TO NEGATIVE 32768.
15725      ;*CALL:
15726      ;*      RDDEC      ;:READ A DECIMAL NUMBER
15727      ;*      RETURN HERE ;:NUMBER IS ON TOP OF THE STACK
15728      ;
15729
15730 065142 011646 $RDDEC: MOV      (SP),-(SP)      ;:PROVIDE SPACE FOR
15731 065144 016666 000004 000002 MOV      4(SP),2(SP) ;:THE INPUT NUMBER
15732 065152      PUSH      R0,R1,R2
15733 065160 104412 1$:      RDLIN      ;:READ AN ASCII LINE
15734 065162 012600      MOV      (SP)+,R0      ;:ADDRESS OF 1ST CHAR.
15735 065164 010037 065310      MOV      R0,6$      ;:SAVE INCASE OF BAD INPUT
15736 065170 005046      CLR      -(SP)      ;:CLEAR DATA WORD
15737 065172 005002      CLR      R2      ;:SIGN SET POSITIVE
15738 065174 122710 000055      CMPB     #'-(R0)      ;:SEE IF A MINUS SIGN WAS TYPED
15739 065200 001001      BNE      2$      ;:BR IF NO MINUS SIGN
15740 065202 112002      MOVB     (R0)+,R2      ;:SAVE FOR LATER USE
15741 065204 112001 2$:      MOVB     (R0)+,R1      ;:PICKUP THIS CHARACTER
15742 065206 001424      BEQ      3$      ;:GET OUT IF ZERO
15743 065210 122701 000060      CMPB     #'0,R1      ;:MAKE SURE THIS CHARACTER
15744 065214 003032      BGT      5$      ;:IS A DIGIT BETWEEN 0 & 9
15745 065216 122701 000071      CMPB     #'9,R1
15746 065222 002427      BLT      5$
15747 065224 032716 170000      BIT      #'C7777,(SP)      ;:DON'T LET NUMBER GET TO BIG
15748 065230 001024      BNE      5$      ;:BR IF NUMBER WOULD OVERFLOW
15749 065232 006316      ASL      (SP)      ;:*2
15750 065234 011646      MOV      (SP),-(SP)      ;:SAVE FOR LATER
15751 065236 006316      ASL      (SP)      ;:*4
15752 065240 006316      ASL      (SP)      ;:*8
15753 065242 062616      ADD      (SP)+,(SP)      ;:*10
15754 065244 102416      BVS      5$      ;:OVERFLOW ISN'T ALLOWED
15755 065246 162701 000060      SUB      #'0,R1      ;:STRIP AWAY THE ASCII JUNK
15756 065252 060116      ADD      R1,(SP)      ;:ADD IN THIS DIGIT
15757 065254 102412      BVS      5$      ;:OVERFLOW ISN'T ALLOWED
15758 065256 000752      BR       2$      ;:LOOP
15759 065260 005702 3$:      TST      R2      ;:CHECK IF NUMBER IS NEG
15760 065262 001401      BEQ      4$      ;:BR IF NO
15761 065264 005416      NEG      (SP)      ;:YES--NEGATE THE NUMBER
15762 065266 012666 000012 4$:      MOV      (SP)+,12(SP) ;:SAVE THE RESULT
15763 065272      POP      R2,R1,R0
15764 065300 000002      RTI      ;:RETURN
15765
15766 065302 005726 5$:      TST      (SP)+      ;:CLEAN PARTIAL NUMBER FROM STACK
15767 065304 105010      CLRB     (R0)      ;:SET A TERMINATOR
15768 065306      TYPE      ;:TYPE THE INPUT UP TO BAD CHAR.
15769 065310 000000 6$:      .WORD     0      ;:POINTER GOES HERE
15770 065312      TYPE      MSG062      ;:INPUT MUSST BE A
15771 065316      TYPE      MSG065      ;:DECIMAL
15772 065322      TYPE      MSG064      ;:NUMBER
15773 065326 000714      BR       1$      ;:TRY AGAIN

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 488  
 ROUTINE SAVE AND RESTORE R0-R5

```

15775                      .SBTTL  ROUTINE SAVE AND RESTORE R0-R5
15776
15777                      :*****
15778                      :*SAVE R0-R5
15779                      :*CALL:
15780                      :*      SAVREG
15781                      :*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
15782                      :*
15783                      :*TOP---(+16)
15784                      :* +2---(+18)
15785                      :* +4---R5
15786                      :* +6---R4
15787                      :* +8---R3
15788                      :*+10---R2
15789                      :*+12---R1
15790                      :*+14---R0
15791
15792 065330                $SAVREG:
15793 065330                PUSH    R0,R1,R2,R3,R4,R5
15794 065344 016646 000022    MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
15795 065350 016646 000022    MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
15796 065354 016646 000022    MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
15797 065360 016646 000022    MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
15798 065364 000002          RTI
15799
15800                      :*RESTORE R0-R5
15801                      :*CALL:
15802                      :*      RESREG
15803 065366                $RESREG:
15804 065366 012666 000022    MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
15805 065372 012666 000022    MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
15806 065376 012666 000022    MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
15807 065402 012666 000022    MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
15808 065406                POP     R5,R4,R3,R2,R1,R0
15809 065422 000002          RTI

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 489  
 ROUTINE RANDOM NUMBER GENERATOR

```

15811                                     .SBTTL  ROUTINE RANDOM NUMBER GENERATOR
15812
15813 :*****
15814 :*THIS ROUTINE IS A DOUBLE PRECISION PSEUDO RANDOM NUMBER GENERATOR
15815 :*WITH A RANGE OF 0 TO 2**(+33)-1.
15816 :*CALL:
15817 :*      CALL      $RAND      ;;CALL THE ROUTINE
15818 :*      RETURN     ;;RETURN HERE THE RANDOM
15819 :*                  ;;NUMBER WILL BE IN
15820 :*                  ;;$HINUM,$LONUM
15821
15822 $RAND:  PUSH      R0,R1,R2      ;;SET R0 WITH LOW
15823         MOV       SEEDLO,R0    ;;SET R1 WITH HIGH
15824         MOV       SEEDHI,R1    ;;SET SHIFT COUNT
15825         MOV       #7,R2        ;;SHIFT R0 LEFT AND
15826         1$:      ASL        R0  ;;ROTATE CARRY INTO R1 AND
15827         ROL       R1
15828         SOB       R2,1$
15829         ADD       SEEDLO,R0    ;;ADD NUMBER TO MAKE X 129
15830         ADC       R1          ;;PROPOGATE CARRY
15831         ADD       SEEDHI,R1    ;;ADD NUMBER TO MAKE X 129
15832         ADD       #1057,R0     ;;ADD LOW CONSTANT
15833         ADC       R1          ;;PROPOGATE CARRY
15834         ADD       #47401,R1    ;;ADD HIGH CONSTANT
15835         MOV       R0,SEEDLO    ;;SAVE R0
15836         MOV       R1,SEEDHI    ;;SAVE R1
15837         POP       R2,R1,R0
15838         RETURN
  
```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 491  
 ROUTINE DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT

```

15841 .SBTTL ROUTINE DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT
15842 *****
15843 :THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
15844 :UNSIGNED OCTAL ASCII NUMBER.
15845 :CALL
15846 :*      MOV      #PNTR,-(SP)      ;; POINTER TO LOW WORD OF BINARY NUMBER
15847 :*      CALL     $DB20            ;; CALL THE ROUTINE
15848 :*      RETURN                      ;; THE ADDRESS OF THE FIRST ASCII CHAR. IS ON THE STACK
15849
15850
15851 065520 104415 $DB20: SAVREG                      ;; SAVE ALL REGISTERS
15852 065522 016601 000002      MOV      2(SP),R1      ;; PICKUP THE POINTER TO LOW WORD
15853 065526 012705 065637      MOV      #SOCTVL+13.,R5  ;; POINTER TO DATA TABLE
15854 065532 012704 000014      MOV      #12.,R4       ;; DO ELEVEN CHARACTERS
15855 065536 012703 177770      MOV      #^C7,R3       ;; MASK
15856 065542 012100      MOV      (R1)+,R0      ;; LOWER WORD
15857 065544 012101      MOV      (R1)+,R1      ;; HIGH WORD
15858 065546 005002      CLR      R2            ;; TERMINATOR
15859 065550 110245 1$:      MOVB      R2,-(R5)      ;; PUT CHARACTER IN DATA TABLE
15860 065552 010002      MOV      R0,R2          ;; GET THIS DIGIT
15861 065554 005304      DEC      R4            ;; COUNT THIS CHARACTER
15862 065556 003007      BGT      3$            ;; BR IF NOT THE LAST DIGIT
15863 065560 001405      BEQ      2$            ;; BR IF IT IS THE LAST DIGIT
15864 065562 005205      INC      R5            ;; ALL DIGITS DONE-ADJUST POINTER FOR FIRST
15865 065564 010566 000002      MOV      R5,2(SP)      ;; ASCII CHAR. & PUT IT ON THE STACK
15866 065570 104416      RESREG                      ;; RESTORE ALL REGISTERS
15867 065572 000207      RETURN                      ;; RETURN TO USER
15868 065574 006203 2$:      ASR      R3            ;; POSITION THE MASK FOR THE LAST DIGIT
15869 065576 006001 3$:      ROR      R1            ;; POSITION THE BINARY NUMBER FOR
15870 065600      ROR      R0                      ;; THE NEXT OCTAL DIGIT
15871 065602      ROR      R1
15872 065604      ROR      R0
15873 065606      ROR      R1
15874 065610      ROR      R0
15875 065612      BIC      R3,R2                      ;; MASK OUT ALL JUNK
15876 065614 040302 000060      ADD      #^0,R2      ;; MAKE THIS CHAR. ASCII
15877 065620 000753      BR      1$            ;; GO PUT IT IN THE DATA TABLE
15878 065622 000016      $OCTVL: .REPT 14.      ;; RESERVE DATA TABLE
15881 065626      $OCT8=$OCTVL+4      ;; POINTER TO 11 DIGIT NUMBER

```

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 492

## TABLES

			.SBTTL TABLES	
15883				
15884				
15885			.SBTTL APT MAILBOX-ETABLE	
15886	065640		\$MAIL:	
15887	065640	000000	\$MSGTY:	.WORD 0 ::MESSAGE TYPE CODE
15888	065642	000000	\$FATAL:	.WORD 0 ::FATAL ERROR NUMBER (ERROR PC)
15889	065644	000000	\$TESTN:	.WORD 0 ::TEST PATTERN NUMBER
15890	065646	000000	\$PASS:	.WORD 0 ::PASS COUNT
15891	065650	000000	\$DEVCT:	.WORD 0 ::DEVICE COUNT
15892	065652	000000	\$UNIT:	.WORD 0 ::I/O UNIT NUMBER
15893	065654	000000	\$MSGAD:	.WORD 0 ::MESSAGE ADDRESS
15894	065656	000000	\$MSGLG:	.WORD 0 ::MESSAGE LENGTH
15895	065660		\$ETABLE:	
15896	065660	000	\$ENV:	.BYTE 0 ::ENVIRONMENT BYTE ;SET TO A 1 FOR APT AUTO MODE
15897			\$NOTE:	IF BIT #7 IS SET IN \$ENVM THE TABLE BELOW (BEGINNING AT \$MAMS1 AND
15898				ENDING AT \$MADR4) MUST BE FILLED IN TO INDICATE THE PROPER AMOUNT OF
15899				EACH TYPE OF MEMORY.
15900	065661	000	\$ENVM:	.BYTE 0 ::ENVIRONMENT MODE
15901				BIT7(200)=USE APT SIZE INFO ;BIT5(40)=NO CONSOLE
15902	065662	000101	\$SWREG:	.WORD 101 ::APT SWITCH REGISTER
15903	065664	000000	\$USWR:	.WORD 0 ::USED TO LIMIT THE NUMBER OF PASSES
15904	065666	000000	\$CPUOP:	.WORD 0 ::CPU TYPE,OPTIONS
15905				BITS 15-11=CPU TYPE
15906				11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
15907				11/70=06,PDQ=07,Q=10
15908				BIT 10=REAL TIME CLOCK
15909				BIT 9=FLOATING POINT PROCESSOR
15910				BIT 8=MEMORY MANAGEMENT
15911	065670	001	\$MAMS1:	.BYTE 1 ::HIGH ADDRESS,M.S. BYTE ;DEFAULT = 64K
15912	065671	004	\$MTYP1:	.BYTE 4 ::MEM. TYPE,BLK#1
15913				MEM. TYPE BYTE -- (HIGH BYTE)
15914				900 NSEC CORE=001
15915				300 NSEC BIPOLAR=002
15916				PARITY MOS=003
15917				ERROR CORRECTING MOS=004
15918	065672	177776	\$MADR1:	.WORD 177776 ::HIGH ADDRESS,BLK#1
15919				MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF 'TYPE' ABOVE
15920	065674	000	\$MAMS2:	.BYTE 0 ::HIGH ADDRESS,M.S. BYTE
15921	065675	000	\$MTYP2:	.BYTE 0 ::MEM. TYPE,BLK#2
15922	065676	000000	\$MADR2:	.WORD 0 ::MEM.LAST ADDRESS,BLK#2
15923	065700	000	\$MAMS3:	.BYTE 0 ::HIGH ADDRESS,M.S.BYTE
15924	065701	000	\$MTYP3:	.BYTE 0 ::MEM. TYPE,BLK#3
15925	065702	000000	\$MADR3:	.WORD 0 ::MEM.LAST ADDRESS,BLK#3
15926	065704	000	\$MAMS4:	.BYTE 0 ::HIGH ADDRESS,M.S.BYTE
15927	065705	000	\$MTYP4:	.BYTE 0 ::MEM. TYPE,BLK#4
15928	065706	000000	\$MADR4:	.WORD 0 ::MEM.LAST ADDRESS,BLK#4
15929	065710	000000	\$VECT1:	.WORD 0 ::INTERRUPT VECTOR#1,BUS PRIORITY#1
15930	065712	000000	\$VECT2:	.WORD 0 ::INTERRUPT VECTOR#2BUS PRIORITY#2
15931	065714	000000	\$BASE:	.WORD 0 ::BASE ADDRESS OF EQUIPMENT UNDER TEST
15932	065716	000000	\$DEVN:	.WORD 0 ::DEVICE MAP
15933				
15934	065720	000000	\$CDW1:	.WORD 0
15935	065722	000000	\$CDW2:	.WORD 0

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 493  
APT MAILBOX-ETABLE

```

15937      ;THE FOLLOWING LOCATIONS SPECIFY WHICH PATTERNS
15938      ;ARE TO BE RUN FOR PARTICULAR MEMORIES
15939      ;
15940      ;REFERENCE THE TABLE LISTED BELOW TO RELATE BITS TO PATTERNS.
15941      ;BITO SET WILL RUN THE FIRST ENTRY IN THE TABLE, BITO SET
15942      ;IN THE SECOND WORD WILL RUN THE 17TH ENTRY IN THE TABLE ...
15943      ;
15944      ;NOTE** NULL TESTS DO NOT TAKE ANY TIME
15945      ;
15946      ;FIELD SERVICE VALUE
15946 065724 177777 $DDW0: .WORD 177777 ;ECC CSR TESTS 177777 TABLE = MKCSRT:
15947 065726 177777 $DDW1: .WORD 177777 ;ECC CSR TESTS 177777 TABLE = MKCSRT:
15948 065730 177777 $DDW2: .WORD 177777 ;ECC PATTERNS 103777 TABLE = MKPAT:
15949 065732 177777 $DDW3: .WORD 177777 ;ECC PATTERNS 177777 TABLE = MKPAT:
15950 065734 177777 $DDW4: .WORD 177777 ;PARITY PATTERNS 003777 TABLE = MJPAT:
15951 065736 177777 $DDW5: .WORD 177777 ;PARITY PATTERNS 177774 TABLE = MJPAT:
15955 065740 $ETEND:
15956      ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
15957      ;INTERFACE SPEC.
15958
15959 065740 $APTHD:
15960 065740 000000 $HIBTS: .WORD 0 ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
15961 065742 065640 $MBADR: .WORD $MAIL ;;ADDRESS OF APT MAILBOX (BITS 0-15)
15962 065744 000043 $STMT: .WORD 35. ;;RUN TIM OF LONGEST TEST
15963 065746 001274 $PASTM: .WORD 700. ;;RUN TIME IN SECS. OF 1ST PASS ON 128K (QUICK VERIFY)
15964 065750 000000 $UNITM: .WORD 0. ;;EXTRA RUN TIME OF A PASS FOR EACH ADDITIONAL 128K (QV)
15965 065752 000040 .WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)

```

```

15967
15968
15969
15970
15971
15972
15973
15974
15975 065754 010046
15976 065756 016600 000002
15977 065762 005740
15978 065764 111000
15979 065766 006300
15980 065770 016000 066016
15981 065774 000200
15982
15983
15984
15985
15986 065776 011646
15987 066000 016666 000004 000002
15988 066006 000002
15989
15990 066010
15991 066014 000000

```

```

.SBTTL  ROUTINE TRAP DECODER

*****
*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE 'TRAP' INST
*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTIN
*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IN
*GO TO THAT ROUTINE.

$TRAP:  MOV      R0,-(SP)          ;;SAVE R0
        MOV      2(SP),R0         ;;GET TRAP ADDRESS
        TST      -(R0)            ;;BACKUP BY 2
        MOVB     (R0),R0          ;;GET RIGHT BYTE OF TRAP
        ASL      R0               ;;POSITION FOR INDEXING
        MOV      $TRPAD(R0),R0    ;;INDEX TO TABLE
        RTS      R0               ;;GO TO ROUTINE

;;THIS IS USE TO HANDLE THE 'GETPRI' MACRO

$TRAP2: MOV      (SP),-(SP)        ;;MOVE THE PC DOWN
        MOV      4(SP),2(SP)      ;;MOVE THE PSW DOWN
        RTI                      ;;RESTORE THE PSW

$NOTRAP:TYPE     MSG006           ;;UNDEFINED TRAP INSTRUCTION
$HALT2:  HALT

```



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 496  
TRAP TABLE

## .SBTTL TRAP TABLE

:\*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED  
:\*BY THE 'TRAP' INSTRUCTION.

	ROUTINE
15994	
15995	
15996	
15997	
15998	
15999	
16000	
16001 066016 065776	\$TRPAD: .WORD \$TRAP2
16002 066020 056152	\$TYPE :CALL=TYPEIT TRAP+1(104401) TTY TYPEOUT ROUTINE
16003 066022 063110	\$TYPOC :CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
16004 066024 063064	\$TYPOS :CALL=TYPOS TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
16005 066026 066010	\$NOTRAP:\$TYPON :CALL=TYPON TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
16006 066030 063312	\$TYPDS :CALL=TYPDS TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
16007 066032 066010	\$NOTRAP:\$TYPBN :CALL=TYPBN TRAP+6(104406) TYPE BINARY (ASCII) NUMBER
16008	
16009 066034 063712	\$GTSWR :CALL=GTSWR TRAP+7(104407) GET SOFT-SWR SETTING
16010 066036 063536	\$CKSWR :CALL=CKSWR TRAP+10(104410) TEST FOR CHANGE IN SOFT-SWR
16011	
16012 066040 064256	\$RDCHR :CALL=RDCHR TRAP+11(104411) TTY TYPEIN CHARACTER ROUTINE
16013 066042 054406	\$RDLIN :CALL=RDLIN TRAP+12(104412) TTY TYPEIN STRING ROUTINE
16014 066044 064772	\$RDOCT :CALL=RDOCT TRAP+13(104413) READ AN OCTAL NUMBER FROM TTY
16015 066046 065142	\$RDDEC :CALL=RDDEC TRAP+14(104414) READ A DECIMAL NUMBER FROM TTY
16016	
16017 066050 065330	\$SAVREG :CALL=SAVREG TRAP+15(104415) SAVE R0-R5 ROUTINE
16018 066052 065366	\$RESREG :CALL=RESREG TRAP+16(104406) RESTORE R0-R5 ROUTINE
16019	
16020 066054 042644	\$KERNEL :CALL=KERNEL TRAP+17(104417) ENTER KERNEL MODE
16021 066056 042654	\$ENERGIZE:CALL=ENERGIZETRAP+20(104420) TURN ON MEMORY MANAGEMENT & TRAPS
16022 066060 042664	\$DEENERGI:CALL=DEENERGITRAP+21(104421) TURN OFF MEMORY MANAGEMENT & TRAPS
16023	
16024 066062 045100	\$KMAP :CALL=KMAP TRAP+22(104422) MAP KERNEL 1 TO 1
16025	
16026 066064 042674	\$CACHN :CALL=CACHON TRAP+23(104423) TURN CACHE ON
16027 066066 042720	\$CACHF :CALL=CACHOFF TRAP+24(104424) TURN CACHE OFF
16028	
16029 066070 042736	\$LOADC :CALL=LOADCSR TRAP+25(104425) LOAD CORRECT CSR
16030 066072 043032	\$READC :CALL=READCSR TRAP+26(104426) READ CORRECT CSR
16031	
16032 066074 056506	\$PER01 :CALL=PERR01 TRAP+27(104427) PROGRAM DETECTED ERROR
16033 066076 056534	\$PER02 :CALL=PERR02 TRAP+30(104430) PROGRAM DETECTED ERROR
16034 066100 056562	\$PER03 :CALL=PERR03 TRAP+31(104431) PROGRAM DETECTED ERROR
16035 066102 056612	\$PER04 :CALL=PERR04 TRAP+32(104432) PROGRAM DETECTED ERROR
16036 066104 056674	\$PER07 :CALL=PERR07 TRAP+33(104433) PROGRAM DETECTED ERROR
16037 066106 056716	\$PER10 :CALL=PERR10 TRAP+34(104434) PROGRAM DETECTED ERROR
16038 066110 056746	\$PER11 :CALL=PERR11 TRAP+35(104435) PROGRAM DETECTED ERROR
16039 066112 056766	\$PER12 :CALL=PERR12 TRAP+36(104436) PROGRAM DETECTED ERROR
16040 066114 057010	\$PER13 :CALL=PERR13 TRAP+37(104437) PROGRAM DETECTED ERROR
16041 066116 057030	\$PER14 :CALL=PERR14 TRAP+40(104440) PROGRAM DETECTED ERROR
16042 066120 057052	\$PER15 :CALL=PERR15 TRAP+41(104441) PROGRAM DETECTED ERROR
16043 066122 057074	\$PER16 :CALL=PERR16 TRAP+42(104442) PROGRAM DETECTED ERROR
16044 066124 057114	\$PER17 :CALL=PERR17 TRAP+43(104443) PROGRAM DETECTED ERROR
16045 066126 057132	\$PER20 :CALL=PERR20 TRAP+44(104444) PROGRAM DETECTED ERROR
16046 066130 057150	\$PER21 :CALL=PERR21 TRAP+45(104445) PROGRAM DETECTED ERROR
16047 066132 057170	\$PER22 :CALL=PERR22 TRAP+46(104446) PROGRAM DETECTED ERROR
16048 066134 057206	\$PER23 :CALL=PERR23 TRAP+47(104447) PROGRAM DETECTED ERROR
16049 066136 057224	\$PER24 :CALL=PERR24 TRAP+50(104450) PROGRAM DETECTED ERROR
16050 066140 053762	\$PER25 :CALL=PERR25 TRAP+51(104451) PROGRAM DETECTED ERROR

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 496-1  
TRAP TABLE

16051 066142 057414	\$PER26 :CALL=PERR26	TRAP+52(104452) PROGRAM DETECTED ERROR
16052 066144 057434	\$PER27 :CALL=PERR27	TRAP+53(104453) PROGRAM DETECTED ERROR
16053 066146 054210	\$PER30 :CALL=PERR30	TRAP+54(104454) PROGRAM DETECTED ERROR
16054 066150 057624	\$PER31 :CALL=PERR31	TRAP+55(104455) PROGRAM DETECTED ERROR
16055 066152 057722	\$PER32 :CALL=PERR32	TRAP+56(104456) PROGRAM DETECTED ERROR
16056 066154 057770	\$PER33 :CALL=PERR33	TRAP+57(104457) PROGRAM DETECTED ERROR
16057 066156 060050	\$PER34 :CALL=PERR34	TRAP+60(104460) PROGRAM DETECTED ERROR
16058 066150 060102	\$PER35 :CALL=PERR35	TRAP+61(104461) PROGRAM DETECTED ERROR
16059 066162 060136	\$PER36 :CALL=PERR36	TRAP+62(104462) PROGRAM DETECTED ERROR
16060 066164 066010	\$NOTRAP :CALL=PERR37	TRAP+63(104463) PROGRAM DETECTED ERROR
16061 066166 066010	\$NOTRAP :CALL=PERR40	TRAP+64(104464) PROGRAM DETECTED ERROR
16062 066170 066010	\$NOTRAP :CALL=PERR41	TRAP+65(104465) PROGRAM DETECTED ERROR
16063 066172 066010	\$NOTRAP :CALL=PERR42	TRAP+66(104466) PROGRAM DETECTED ERROR
16064 066174 066010	\$NOTRAP :CALL=PERR43	TRAP+67(104467) PROGRAM DETECTED ERROR
16065		
16065 066176 043256	\$ECCDIS :CALL=ECCDIS	TRAP+70(104470) DISABLE ECC ON ALL CSR'S
16067 066200 043272	\$ECC1DIS:CALL=ECC1DIS	TRAP+71(104471) DISABLE ECC ON 1 SELECTED CSR
16068 066202 043304	\$ECCINIT:CALL=ECCINIT	TRAP+72(104472) INITIALIZE ALL MK11 CSR'S
16069 066204 043320	\$ECC1INIT:CALL=ECC1INIT	TRAP+73(104473) INITIALIZE 1 SELECTED MK11 CSR
16070 066206 043360	\$CBCSR :CALL=CBCSR	TRAP+74(104474) WRITE GENERATED CHECKBITS IN ALL CSR'S
16071 066210 043402	\$CB1CSR :CALL=CB1CSR	TRAP+75(104475) WRITE GENERATED CHECKBITS IN 1 SELECTED CSR
16072 066212 043422	\$WASSBE :CALL=WASSBE	TRAP+76(104476) WAS THERE A SBE ON ANY CSR?
16073 066214 043536	\$WAS1SBE:CALL=WAS1SBE	TRAP+77(104477) WAS THERE A SBE ON 1 SELECTED CSR?
16074 066216 043566	\$WASDBE :CALL=WASDBE	TRAP+100(104500) WAS THERE A DBE ON ANY CSR?
16075 066220 043702	\$WAS1DBE:CALL=WAS1DBE	TRAP+101(104501) WAS THERE A DBE ON 1 SELECTED CSR?
16076 066222 043732	\$CLRCSR :CALL=CLRCSR	TRAP+102(104502) CLEAR ALL CSR'S
16077 066224 043744	\$CLR1CSR:CALL=CLR1CSR	TRAP+103(104503) CLEAR 1 SELECTED CSR
16078 066226 043754	\$CHKDIS :CALL=CHKDIS	TRAP+104(104504) DISABLE ECC & WRITE CKBITS FROM ALL CSR'S
16079 066230 043770	\$CHK1DIS:CALL=CHK1DIS	TRAP+105(104505) DISABLE ECC & WRITE CKBITS FROM 1 CSR
16080 066232 043332	\$ENASBE :CALL=ENASBE	TRAP+106(104506) ENABLE TRAPS ON SBE'S FROM ALL CSR'S
16081 066234 043346	\$ENA1SBE:CALL=ENA1SBE	TRAP+107(104507) ENABLE TRAPS ON SBE'S FROM 1 SELECTED CSR
16082 066236 043052	\$TSTRD :CALL=TSTREAD	TRAP+110(104510) TEST LOC (R1) & TST FOR SBE (WITHOUT FETCHES
16083 066240 044050	\$INVALID:CALL=INVALID	TRAP+111(104511) INVALIDATE BACKGROUND PATTERN ON BANK
16084 066242 044100	\$ERRGEN :CALL=ERRGEN	TRAP+112(104512) TEST ERROR ADDRESS
16085 066244 044350	\$CBREG :CALL=CBREG	TRAP+112(104513) ENABLE CHECK/SYNDROME BIT REGISTER
16086 066246 066010	\$NOTRAP	
16087 066250 066010	\$NOTRAP	
16088 066252 066010	\$NOTRAP	
16089 066254 066010	\$NOTRAP	
16090 066256 066010	\$NOTRAP	
16091 066260 066010	\$NOTRAP	

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 498  
TRAP TABLE

16094      177776      ST      =      177776      ;STATUS REGISTER

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 500  
TABLE ERROR POINTER

## .SBTTL TABLE ERROR POINTER

;\*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.  
;\*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN  
;\*LOCATION \$ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.  
;\*NOTE1: IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (ERRPC).  
;\*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

;\* EM ::POINTS TO THE ERROR MESSAGE  
;\* DH ::POINTS TO THE DATA HEADER  
;\* DT ::POINTS TO THE DATA  
;\* DF ::POINTS TO THE DATA FORMAT

\$ERRTB: :ERROR 1  
EM24  
DH13  
DT13  
DF11  
:ERROR 2  
EM2  
DH1  
DT1  
DF2  
:ERROR 3  
EM3  
DH3  
DT3  
DF9  
:ERROR 4  
EM4  
DH3  
DT4  
DF9  
:ERROR 5  
EM5  
DH5  
DT5  
DF2  
:ERROR 6  
EM6  
DH5  
DT5  
DF2  
:ERROR 7  
EM7  
DH5  
DT5  
DF2  
:ERROR 10  
EM53  
DH25  
DT25  
DF2

16097  
16098  
16099  
16100  
16101  
16102  
16103  
16104  
16105  
16106  
16107  
16108  
16109  
16110  
16111 066262  
16112 066262 070750  
16113 066264 073105  
16114 066266 067306  
16115 066270 067667  
16116  
16117 066272 067735  
16118 066274 072414  
16119 066276 067132  
16120 066300 067545  
16121  
16122 066302 067773  
16123 066304 072474  
16124 066306 067150  
16125 066310 067662  
16126  
16127 066312 070025  
16128 066314 072474  
16129 066316 067160  
16130 066320 067662  
16131  
16132 066322 070073  
16133 066324 072530  
16134 066326 067170  
16135 066330 067545  
16136  
16137 066332 070150  
16138 066334 072530  
16139 066336 067170  
16140 066340 067545  
16141  
16142 066342 070175  
16143 066344 072530  
16144 066346 067170  
16145 066350 067545  
16146  
16147 066352 072161  
16148 066354 073627  
16149 066356 067464  
16150 066360 067545

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 502  
TABLE ERROR POINTER

16153			:ERROR 11
16154	066362	070235	EM11
16155	066364	072654	DH7
16156	066366	067222	DT7
16157	066370	067571	DF3
16158			:ERROR 12
16159	066372	070235	EM11
16160	066374	072654	DH7
16161	066376	067222	DT7
16162	066400	067604	DF4
16163			:ERROR 13
16164	066402	070257	EM12
16165	066404	072764	DH10
16166	066406	067252	DT10
16167	066410	067545	DF2
16168			:ERROR 14
16169	066412	070235	EM11
16170	066414	072654	DH7
16171	066416	067222	DT7
16172	066420	067617	DF5
16173			:ERROR 15
16174	066422	070235	EM11
16175	066424	072654	DH7
16176	066426	067222	DT7
16177	066430	067632	DF6
16178			:ERROR 16
16179	066432	070303	EM13
16180	066434	073105	DH13
16181	066436	067306	DT13
16182	066440	067667	DF11
16183			:ERROR 17
16184	066442	070335	EM14
16185	066444	073105	DH13
16186	066446	067306	DT13
16187	066450	067667	DF11
16188			:ERROR 20
16189	066452	070401	EM15
16190	066454	073105	DH13
16191	066456	067306	DT13
16192	066460	067667	DF11
16193			:ERROR 21
16194	066462	072210	EM55
16195	066464	073665	DH26
16196	066466	067476	DT26
16197	066470	067545	DF2
16198			:ERROR 22
16199	066472	070447	EM17
16200	066474	072654	DH7
16201	066476	067222	DT7
16202	066500	067617	DF5
16203			:ERROR 23
16204	066502	072030	EM50
16205	066504	073501	DH23
16206	066506	067422	DT23
16207	066510	067700	DF13

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 504  
TABLE ERROR POINTER

16210			:ERROR 24	
16211	066512	070507	EM19	
16212	066514	073105	DH13	
16213	066516	067306	DT13	
16214	066520	067667	DF11	
16215			:ERROR 25	
16216	066522	070561	EM20	
16217	066524	073105	DH13	
16218	066526	067306	DT13	
16219	066530	067667	DF11	
16220			:ERROR 26	
16221	066532	000000	0	:NO MESSAGE
16222	066534	073100	DH12	
16223	066536	067302	DT12	
16224	066540	067545	DF2	
16225			:ERROR 27	
16226	066542	070640	EM21	
16227	066544	073062	DH11	
16228	066546	067274	DT11	
16229	066550	067545	DF2	
16230			:ERROR 30	
16231	066552	070674	EM22	
16232	066554	073105	DH13	
16233	066556	067306	DT13	
16234	066560	067667	DF11	
16235			:ERROR 31	
16236	066562	000000	0	:NO MESSAGE
16237	066564	073202	DH14	
16238	066566	067330	DT14	
16239	066570	067545	DF2	
16240			:ERROR 32	
16241	066572	070721	EM23	
16242	066574	072530	DH5	
16243	066576	067170	DT5	
16244	066600	067545	DF2	
16245			:ERROR 33	
16246	066602	071027	EM25	
16247	066604	073261	DH15	
16248	066606	067346	DT16	
16249	066610	067645	DF7	
16250			:ERROR 34	
16251	066612	071054	EM26	
16252	066614	073400	DH16	
16253	066616	067376	DT17	
16254	066620	067571	DF3	

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 506  
TABLE ERROR POINTER

16257			:ERROR 35
16258	066622	072134	EM52
16259	066624	073627	DH25
16260	066626	067464	DT25
16261	066630	067545	DF2
16262			:ERROR 36
16263	066632	071125	EM27
16264	066634	073400	DH16
16265	066636	067376	DT17
16266	066640	067660	DF8
16267			:ERROR 37
16268	066642	071622	EM35
16269	066644	072654	DH7
16270	066646	067222	DT7
16271	066650	067571	DF3
16272			:ERROR 40
16273	066652	071215	EM29
16274	066654	072654	DH7
16275	066656	067222	DT7
16276	066660	067571	DF3
16277			:ERROR 41
16278	066662	071277	EM30
16279	066664	072654	DH7
16280	066666	067222	DT7
16281	066670	067617	DF5
16282			:ERROR 42
16283	066672	072331	EM60
16284	066674	073422	DH20
16285	066676	067422	DT23
16286	066700	067700	DF13
16287			:ERROR 43
16288	066702	071407	EM32
16289	066704	072654	DH7
16290	066706	067222	DT7
16291	066710	067571	DF3
16292			:ERROR 44
16293	066712	071514	EM33
16294	066714	072654	DH7
16295	066716	067222	DT7
16296	066720	067571	DF3
16297			:ERROR 45
16298	066722	072064	EM51
16299	066724	073560	DH24
16300	066726	067444	DT24
16301	066730	067710	DF14
16302			:ERROR 46
16303	066732	071707	EM36
16304	066734	072607	DH6
16305	066736	067206	DT6
16306	066740	067545	DF2

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 508  
 TABLE ERROR POINTER

16309			:ERROR 47
16310	066742	071756	EM40
16311	066744	072451	DH2
16312	066746	067404	DT20
16313	066750	067545	DF2
16314			:ERROR 50
16315	066752	072231	EM56
16316	066754	073703	DH27
16317	066756	067504	DT27
16318	066760	067544	DF1
16319			:ERROR 51
16320	066762	072373	EM61
16321	066764	073560	DH24
16322	066766	067444	DT24
16323	066770	067710	DF14
16324			:ERROR 52
16325	066772	071215	EM29
16326	066774	073105	DH13
16327	066776	067306	DT13
16328	067000	067667	DF11
16329			:ERROR 53
16330	067002	071027	EM25
16331	067004	073757	DH30
16332	067006	067524	DT30
16333	067010	067726	DF16
16334			:ERROR 54
16335	067012	072263	EM57
16336	067014	073757	DH30
16337	067016	067524	DT30
16338	067020	067726	DF16
16339			:ERROR 55
16340	067022	070561	EM20
16341	067024	073560	DH24
16342	067026	067444	DT24
16343	067030	067710	DF14
16344			:ERROR 56
16345	067032	070561	EM20
16346	067034	073757	DH30
16347	067036	067524	DT30
16348	067040	067726	DF16
16349			:ERROR 57
16350	067042	070507	EM19
16351	067044	073560	DH24
16352	067046	067444	DT24
16353	067050	067710	DF14
16354			:ERROR 60
16355	067052	070303	EM13
16356	067054	073422	DH20
16357	067056	067422	DT23
16358	067060	067700	DF13
16359			:ERROR 61
16360	067062	071277	EM30
16361	067064	073422	DH20
16362	067066	067422	DT23
16363	067070	067700	DF13
16364			:ERROR 62
16365	067072	070561	EM20



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 508-1  
TABLE ERROR POINTER

16366	067074	073560	DH24	
16367	067076	067444	DT24	
16368	067100	067710	DF14	
16369			:ERROR	63
16370	067102	070674	EM22	
16371	067104	073560	DH24	
16372	067106	067444	DT24	
16373	067110	067710	DF14	
16374			:ERROR	64
16375	067112	100300	EM62	
16376	067114	072530	DH5	
16377	067116	067170	DT5	
16378	067120	067545	DF2	
16379			:ERROR	65
16380	067122	070674	EM22	
16381	067124	073422	DH20	
16382	067126	067422	DT23	
16383	067130	067700	DF13	

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 509  
ERROR DATA TAGS (DT)

					.SBTTL	ERROR DATA TAGS (DT)
16385						
16386	067132	002016	002032	002042	DT1: .WORD	ERRPC,ADDRESS,GOOD,BAD,0
	067140	002050	000000			
16387	067144	002016	000000		DT2: .WORD	ERRPC,0
16388	067150	002016	002034	002070	DT3: .WORD	ERRPC,PADDRESS,PARCNT,0
	067156	000000				
16389	067160	002016	002032	002066	DT4: .WORD	ERRPC,ADDRESS,NEMCNT,0
	067166	000000				
16390	067170	002016	177572	177574	DT5: .WORD	ERRPC,MMR0,MMR1,MMR2,MMR3,CPUERR,0
	067176	177576	172516	177766		
	067204	000000				
16391	067206	002016	002416	002374	DT6: .WORD	ERRPC,APTPAR,LSIZE,APTECC,MSIZE,0
	067214	002420	002376	000000		
16392	067222	002016	002174	002032	DT7: .WORD	ERRPC,DUMMY,ADDRESS,DUMMY,GCOD,BAD,BADXOR
	067230	002174	002042	002050		
	067236	002056				
16393	067240	002174	002174	002174	.WORD	DUMMY,DUMMY,DUMMY,DUMMY,0
	067246	002174	000000			
16394	067252	002176	002200	002202	DT10: .WORD	DETRO,DETR1,DETR2,DETR3,DETR4,DETR5,DETSP,DETPSW,0
	067260	002204	002206	002210		
	067266	002212	002214	000000		
16395	067274	002016	002146	000000	DT11: .WORD	ERRPC,CSR,0
16396	067302	002146	000000		DT12: .WORD	CSR,0
16397	067306	002016	002174	002032	DT13: .WORD	ERRPC,DUMMY,ADDRESS,DUMMY,TSTDAT,TSTDAT+2,CHECK,CSR,0
	067314	002174	002244	002246		
	067322	002310	002146	000000		
16398	067330	177746	177572	177574	DT14: .WORD	CONTRL,MMR0,MMR1,MMR2,MMR3,CPUERR,0
	067336	177576	172516	177766		
	067344	000000				
16399	067346	002016	002174	002174	DT16: .WORD	ERRPC,DUMMY,DUMMY,GOOD,GOOD2,GOOD3
	067354	002042	002044	002046		
16400	067362	002050	002052	002054	.WORD	BAD,BAD2,BAD3,DUMMY,DUMMY,0
	067370	002174	002174	000000		
16401	067376	002016	002174	000000	DT17: .WORD	ERRPC,DUMMY,0
16402	067404	002016	002042	002050	DT20: .WORD	ERRPC,GOOD,BAD,0
	067412	000000				
16403	067414	002016	002174	000000	DT22: .WORD	ERRPC,DUMMY,0
16404	067422	002016	002174	002042	DT23: .WORD	ERRPC,DUMMY,GOOD,BAD,DUMMY,DUMMY,DUMMY,DUMMY,0
	067430	002050	002174	002174		
	067436	002174	002174	000000		
16405	067444	002016	002174	002146	DT24: .WORD	ERRPC,DUMMY,CSR,DUMMY,DUMMY,DUMMY,DUMMY,0
	067452	002174	002174	002174		
	067460	002174	000000			
16406	067464	002016	002042	002146	DT25: .WORD	ERRPC,GOOD,CSR,CSRNO,0
	067472	002150	000000			
16407	067476	002016	002050	000000	DT26: .WORD	ERRPC,BAD,0
16408	067504	002016	002174	002032	DT27: .WORD	ERRPC,DUMMY,ADDRESS,DUMMY,DUMMY,DUMMY,DUMMY,0
	067512	002174	002174	002174		
	067520	002174	000000			
16409	067524	002016	002174	002174	DT30: .WORD	ERRPC,DUMMY,DUMMY,GOOD,BAD,CSR,DUMMY,0
	067532	002042	002050	002146		
	067540	002174	000000			

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 511  
 ERROR DATA FORMATS (DF)

				.SBTTL	ERROR DATA FORMATS (DF)
16412					
16413	067544	000		DF1: .BYTE	0
16414	067545	000	000	DF2: .BYTE	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
	067550	000	000		
	067553	000	000		
	067556	000	000		
	067561	000	000		
	067564	000	000		
	067567	000	000		
16415	067571	000	005	DF3: .BYTE	0,5,0,8.,0,0,0,3,6,2,4
	067574	010	000		
	067577	000	003		
	067602	002	004		
16416	067604	000	005	DF4: .BYTE	0,5,0,8.,0,8.,8.,3,6,2,4
	067607	010	000		
	067612	010	003		
	067615	002	004		
16417	067617	000	005	DF5: .BYTE	0,5,0,8.,9.,9.,9.,3,6,2,4
	067622	010	011		
	067625	011	003		
	067630	002	004		
16418	067632	000	005	DF6: .BYTE	0,5,0,8.,9.,8.,8.,3,6,2,4
	067635	010	011		
	067640	010	003		
	067643	002	004		
16419	067645	000	005	DF7: .BYTE	0,5,8.,0,0,9.,0,0,9.,2,4
	067650	000	000		
	067653	000	000		
	067656	002	004		
16420	067660	000	005	DF8: .BYTE	0,5
16421	067662	000	001	DF9: .BYTE	0,1,1,1,1
	067665	001	001		
16422	067667	000	005	DF11: .BYTE	0,5,0,8.,0,0,0,0,0
	067672	010	000		
	067675	000	000		
16423	067700	000	005	DF13: .BYTE	0,5,0,0,3,6,2,4
	067703	000	003		
	067706	002	004		
16424	067710	000	005	DF14: .BYTE	0,5,0,3,6,2,4
	067713	003	006		
	067716	004			
16425	067717	000	005	DF15: .BYTE	0,5,0,8.,3,6,4
	067722	010	003		
	067725	004			
16426	067726	000	005	DF16: .BYTE	0,5,8.,0,0,3,4
	067731	000	003		
	067734	004			

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 513  
ERROR MESSAGES (EM)

16429					.SBTTL	ERROR MESSAGES (EM)
16435 067735	103	101	116	EM2:	.ASCIZ	/CAN'T SET 22 BIT MODE IN MMR3/
16436 067773	120	101	122	EM3:	.ASCIZ	/PARITY ERROR(S) IN BANK 0/
16437 070025	116	117	116	EM4:	.ASCIZ	/NON-EXISTANT MEMORY (HOLES) IN BANK 0/
16438 070073	111	114	114	EM5:	.ASCIZ	/ILLEGAL OR RESERVED INSTRUCTION (TRAP TO 10)/
16439 070150	125	116	105	EM6:	.ASCIZ	/UNEXPECTED TRAP TO 4/
16440 070175	115	105	115	EM7:	.ASCIZ	/MEMORY MANAGEMENT (TRAP TO 250)/
16441 070235	115	105	115	EM11:	.ASCIZ	/MEMORY DATA ERROR/
16442 070257	104	105	124	EM12:	.ASCIZ	/DETAILED ERROR DUMP/
16443 070303	115	111	123	EM13:	.ASCIZ	/MISSING EXPECTED SBE FLAG/
16444 070335	127	122	111	EM14:	.ASCIZ	/WRITE BYTE FAILED TO CLEAR SBE FLAG/
16445 070401	106	101	111	EM15:	.ASCIZ	/FAILED TO GET INTERRUPT WITH DBE FLAG/
16446 070447	115	105	115	EM17:	.ASCIZ	/MEMORY DATA ERROR IN CHECK BITS/
16447 070507	123	102	105	EM19:	.ASCIZ	/SBE-DBE CAUSED PARITY TRAP WHEN INHIBITED/
16448 070561	123	102	105	EM20:	.ASCIZ	/SBE-DBE DID NOT CAUSE PARITY TRAP WHEN ENABLED/
16449 070640	123	102	105	EM21:	.ASCIZ	/SBE-DBE ON MASTER TEST WORD/
16450 070674	115	111	123	EM22:	.ASCIZ	/MISSING EXPECTED DBE/
16451 070721	125	116	105	EM23:	.ASCIZ	/UNEXPECTED PARITY TRAP/
16452 070750	122	105	103	EM24:	.ASCIZ	/RECEIVED DBE FLAG WHEN EXPECTING ONLY SBE FLAG/
16453 071027	103	110	105	EM25:	.ASCIZ	/CHECK BIT DATA ERROR/
16454 071054	101	104	104	EM26:	.ASCIZ	/ADDRESS PARITY ERROR DID NOT CAUSE ABORT/
16455 071125	105	103	103	EM27:	.ASCIZ	/ECC INHIBIT MODE POINTER FAILURE - DID NOT PROTECT BANK/
16456 071215	103	117	122	EM29:	.ASCIZ	/CORRECTION FAILURE WITH ECC ENABLED ON FORCED SBE/
16457 071277	127	122	111	EM30:	.ASCII	/WRITE BYTE WITH ECC ENABLED FAILED TO CLEAR DATA AT/<CRLF>
16458 071363	106	117	122		.ASCIZ	/FORCED SBE LOCATION/
16459 071407	115	117	126	EM32:	.ASCIZ	/MOVB #360,(R2)+ WITH ECC ENABLED CHANGED DATA AT FORCED DBE LOCATION/
16460 071514	115	117	126	EM33:	.ASCIZ	/MOV #177400,(R1) WITH ECC ENABLED CHANGED DATA AT FORCED DBE LOCATION/
16461 071622	125	116	105	EM35:	.ASCIZ	/UNEXPECTED CORRECTION WITH ECC DISABLE ON FORCED SBE/
16462 071707	101	120	124	EM36:	.ASCIZ	/APT SIZE DISAGREES WITH PROGRAM SIZING/
16463 071756	102	122	101	EM40:	.ASCIZ	/BRANCH GOBBLE FAILED CONDITION CODES TEST/
16464 072030	102	101	104	EM50:	.ASCIZ	/BAD ERROR ADDRESS GENERATED/
16465 072064	106	114	101	EM51:	.ASCIZ	/FLAGS NOT SET ON FORCED UNCORRECTED SBE/
16466 072134	102	111	124	EM52:	.ASCIZ	/BIT SET ERROR IN CSR/
16467 072161	102	111	124	EM53:	.ASCIZ	/BIT CLEAR ERROR IN CSR/
16468 072210	111	114	114	EM55:	.ASCIZ	/ILLEGAL CSR TYPE/
16469 072231	102	101	104	EM56:	.ASCIZ	/BAD PARITY TRAP GENERATED/
16470 072263	127	122	117	EM57:	.ASCIZ	/WRONG CHECK BIT READ BACK FROM MEMORY/
16471 072331	127	122	117	EM60:	.ASCIZ	/WRONG SYNDROME BITS READ INTO CSR/
16472 072373	103	123	122	EM61:	.ASCIZ	/CSR UPDATE ERROR/

16475						.SBTTL	ERROR	DATA HEADERS (DH)												
16476	072414	040	040	120	DH1:	.ASCIZ	/ PC	DEV ADD	GOOD	BAD/										
16477	072451	040	040	120	DH2:	.ASCIZ	/ PC	GD-CC	BD-CC/											
16478	072474	040	040	120	DH3:	.ASCIZ	/ PC	1ST ADD	# OF ERRORS/											
16479	072530	040	040	120	DH5:	.ASCIZ	/ PC	MMR0	MMR1	MMR2	MMR3	CPUERR/								
16480	072607	040	040	120	DH6:	.ASCIZ	/ PC	APTPAR	LSIZE	APTECC	MSIZE/									
16481	072654	040	040	120	DH7:	.ASCII	/ PC	BANK	VADD	PADD	GOOD/									
16482	072720	040	040	040		.ASCIZ	/	BAD	XOR	CSR	MTYP	INT	PAT/							
16483	072764	040	040	122	DH10:	.ASCIZ	/ R0	R1	R2	R3	R4	R5								
16484	073062	040	040	120	DH11:	.ASCIZ	/ PC	CSR/												
16485	073100	040	103	123	DH12:	.ASCIZ	/ CSR/													
16486	073105	040	040	120	DH13:	.ASCII	/ PC	BANK	VADD	PADD	WROTE1	WROTE2/								
16487	073162	040	103	110		.ASCIZ	/ CHKBITS	CSR/												
16488	073202	103	117	116	DH14:	.ASCIZ	/CONTRL	MMR0	MMR1	MMR2	MMR3	CPUERR/								
16489	073261	040	040	120	DH15:	.ASCII	/ PC	BANK	PADD	GD-WD1	GD-WD2	GD-CHK/								
16490	073336	040	102	101		.ASCIZ	/	BAD-WD1	BAD-WD2	BAD-CHK	INT	PAT/								
16491	073400	040	040	120	DH16:	.ASCIZ	/ PC	BANK/												
16492	073415	040	040	120	DH19:	.ASCIZ	/ PC/													
16493	073422	040	040	120	DH20:	.ASCIZ	/ PC	BANK	GD-CSR	(CSR)	CSR	MTYP	INT	PAT/						
16494	073501	040	040	120	DH23:	.ASCIZ	/ PC	BANK	GD-ERR	BAD-ERR	CSR	MTYP	INT	PAT/						
16495	073560	040	040	120	DH24:	.ASCIZ	/ PC	BANK	(CSR)	CSR	MTYP	INT	PAT/							
16496	073627	040	040	120	DH25:	.ASCIZ	/ PC	GD-DAT	(CSR)	CSRNO/										
16497	073665	040	040	120	DH26:	.ASCIZ	/ PC	BADCODE/												
16498	073703	040	040	120	DH27:	.ASCIZ	/ PC	BANK	VADD	PADD	CSR	MTYP	PAT/							
16499	073757	040	040	120	DH30:	.ASCIZ	/ PC	BANK	PADD	WROTE	READ	CSR	PAT/							

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 517  
MESSAGES

16502						.SBTTL	MESSAGES	
16503	074036	200	040	040	MSG001:	.ASCII	<CRLF>/	MEMORY CONFIGURATION MAP/
16504	074120	200	040	040	MSG002:	.ASCII	<CRLF>/	16K WORD BANKS/
16505	074175	200	040	040	MSG003:	.ASCII	<CRLF>/	1 2 3/
16506	074237	040	040	040		.ASCII	/	4 5 6 7 /
16507	074302	200	040	040	MSG004:	.ASCII	<CRLF>/	012345670123456701234567/
16508	074343	060	061	062		.ASCII	/	01234567012345670123456701234567/
16509	074410	200	105	122	MSG005:	.ASCII	<CRLF>/ERRORS /	
16510	074422	200	125	116	MSG006:	.ASCII	<CRLF>/UNDEFINED TRAP INSTRUCTION /<32>	
16511	074457	200	111	116	MSG007:	.ASCII	<CRLF>/INTRLV /	:INTERLEAVED CSR #
16512	074471	200	115	105	MSG009:	.ASCII	<CRLF>/MEMTYPE /	:MEMORY TYPE
16513	074503	200	120	122	MSG010:	.ASCII	<CRLF>/PROTECT /	:MEMORY PROTECTED
16514	074515	040	040	040	MSG011:	.ASCII	/	0 1 2 3 4 5 6/
16515	074603	064	065	066	MSG012:	.ASCII	/	45670123456701234567012345670123456701234567/
16516	074700	130	000		MSG013:	.ASCII	/X/	
16517	074702	040	000		MSG014:	.ASCII	/ /	:SPACE
16518	074704	000	000		MSG015:	.BYTE	0,0	:FOR SINGLE ASCII CHARACTERS & TERMINATOR
16519	074706	200	103	123	MSG016:	.ASCII	<CRLF>/CSR /	
16520	074720	040	040	040	MSG017:	.ASCII	/ /	:8 SPACES
16521	074731	040	040	000	MSG018:	.ASCII	/ /	:2 SPACES
16522	074734	040	040	040	MSG019:	.ASCII	/ /	:3 SPACES
16523	074740	200	106	123	MSG020:	.ASCII	<CRLF>/FS COMMAND MODE/	
16524	074761	200	103	117	MSG021:	.ASCII	<CRLF>/COMMANDS AVAILABLE:/	
16525	075005	200	060	040		.ASCII	<CRLF>/0 = EXIT/	
16526	075016	200	061	040		.ASCII	<CRLF>/1 = READ CSR/	
16527	075033	200	062	040		.ASCII	<CRLF>/2 = LOAD CSR/	
16528	075050	200	063	040		.ASCII	<CRLF>/3 = EXAMINE MEMORY/	
16529	075073	200	064	040		.ASCII	<CRLF>/4 = MODIFY MEMORY/	
16530	075115	200	065	040		.ASCII	<CRLF>/5 = SELECT BANK & TEST/	
16531	075144	200	066	040		.ASCII	<CRLF>/6 = TYPE CONFIG MAP/	
16532	075170	200	067	040		.ASCII	<CRLF>/7 = SOB-A-LONG TEST/	
16533	075214	200	070	040		.ASCII	<CRLF>/8 = ERROR SUMMARY/	
16534	075236	200	071	075		.ASCII	<CRLF>/9= REFRESH TEST/	
16535	075257	200	061	060		.ASCII	<CRLF>/10= SET FILL COUNT/	
16536	075302	200	061	061		.ASCII	<CRLF>/11= ENTER KAMIKAZE MODE/	
16537	075332	200	061	062		.ASCII	<CRLF>/12= EXIT KAMIKAZE MODE/	
16538	075361	200	061	063		.ASCII	<CRLF>/13= TURN CACHE OFF/	
16539	075404	200	061	064		.ASCII	<CRLF>/14= TURN CACHE ON/	
16540	075426	200	061	065		.ASCII	<CRLF>/15= TEST SELECTED BANKS/	
16541	075456	200	061	066		.ASCII	<CRLF>/16= TEST ALL BANKS/	
16542	075501	200	061	067		.ASCII	<CRLF>/17= ENABLE TRACE/	
16543	075522	200	061	070		.ASCII	<CRLF>/18= DISABLE TRACE/	
16544	075544	015	012	000		.BYTE	15,12,0	
16545	075547	200	127	110	MSG022:	.ASCII	<CRLF>/WHICH CSR(0-F)? /	
16546	075571	200	103	123	MSG023:	.ASCII	<CRLF>/CSR WORD? /	
16547	075605	200	103	123	MSG025:	.ASCII	<CRLF>/CSR DOES NOT EXIST/	
16548	075631	200	103	117	MSG026:	.ASCII	<CRLF>/COMMAND:/	
16549	075643	200	117	114	MSG027:	.ASCII	<CRLF>/OLD CSR WAS/	
16550	075660	200	103	123	MSG028:	.ASCII	<CRLF>/CSR IS NOW/	
16551	075674	200	105	130	MSG029:	.ASCII	<CRLF>/EXAMINE MEMORY/	
16552	075714	200	102	101	MSG030:	.ASCII	<CRLF>/BANK(0-167)? /	
16553	075733	200	120	110	MSG031:	.ASCII	<CRLF>/PHYSICAL ADDRESS(0-17757776)? /	
16554	075773	200	120	101	MSG032:	.ASCII	<CRLF>/PARITY ABORT/<32>	
16555	076012	200	124	111	MSG033:	.ASCII	<CRLF>/TIMEOUT TRAP/<32>	
16556	076031	200	102	131	MSG034:	.ASCII	<CRLF>/BYPASSING ECC TESTS ON BANK /	
16557	076067	040	104	125	MSG034:	.ASCII	/ DUE TO SBE LOCATIONS/	
16558	076115	121	126	000	MSG035:	.ASCII	/QV/	

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 517-1

## MESSAGES

16559	076120	200	115	117	MSG036:	.ASCIIZ	<CRLF>/MODIFY MEMORY/
16560	076137	200	117	114	MSG037:	.ASCIIZ	<CRLF>/OLD DATA WAS /
16561	076156	200	104	101	MSG038:	.ASCIIZ	<CRLF>/DATA IS NOW /
16562	076174	200	111	116	MSG039:	.ASCIIZ	<CRLF>/INPUT NEW DATA? /
16563	076216	200	123	105	MSG040:	.ASCIIZ	<CRLF>/SELECT BANK & TEST/
16564	076242	200	102	101	MSG041:	.ASCIIZ	<CRLF>/BANK NOT ACCESSABLE/
16565	076267	200	124	105	MSG042:	.ASCIIZ	<CRLF>/TEST(0-47)? /
16566	076305	200	124	105	MSG043:	.ASCIIZ	<CRLF>/TEST 0 DATA IS? /
16567	076327	200	124	117	MSG046:	.ASCIIZ	<CRLF>/TO ESCAPE TYPE ANY KEY/<CRLF><12><12>
16568	076362	200	124	105	MSG047:	.ASCIIZ	<CRLF>/TEST COMPLETE/
16569	076401	040	116	117	MSG048:	.ASCIIZ	/ NOT AVAILABLE NOW - TRY LATER! /
16570	076441	200	102	101	MSG049:	.ASCIIZ	<CRLF>/BANK REQUIRES RELOCATION/
16571							.EVEN
16572	076474	120	117	127	MSG051:	.ASCIIZ	/POWER RECOVERY/
16573	076513	200	123	117	MSG055:	.ASCIIZ	<CRLF>/SOB-A-LONG TEST/
16574	076534	200	102	105	MSG056:	.ASCIIZ	<CRLF>/BELL = EACH PASS COMPLETE/
16575	076567	200	040	040	MSG058:	.ASCIIZ	<CRLF>/ CSR CSR .../
16576	076611	077	077	077	MSG061:	.ASCIIZ	/?????/
16577	076620	111	116	120	MSG062:	.ASCIIZ	/INPUT MUST BE A/
16578	076640	116	040	117	MSG063:	.ASCIIZ	/N OCTAL /
16579	076651	116	125	115	MSG064:	.ASCIIZ	/NUMBER/<CRLF>
16580	076661	040	104	105	MSG065:	.ASCIIZ	/ DECIMAL /
16581	076673	200	105	122	MSG066:	.ASCIIZ	<CRLF>/ERRORS > 20 - ABORTING FOR XXDP CHAIN/
16582	076742	106	101	124	MSG067:	.ASCIIZ	/FATAL /
16583	076751	113	040	127	MSG070:	.ASCIIZ	/K WORDS OF MEMORY TOTAL/<CRLF>
16584	077002	200	122	105	MSG073:	.ASCIIZ	<CRLF>/REFRESH TEST/
16585	077020	200	122	105	MSG075:	.ASCIIZ	<CRLF>/RELOCATION NOT POSSIBLE/<32>
16586	077052	200	040	040	MSG076:	.ASCIIZ	<CRLF>/ BANK ERRORS/<CRLF>
16587	077073	200	105	116	MSG077:	.ASCIIZ	<CRLF>/END PASS #/
16588	077107	040	105	122	MSG079:	.ASCIIZ	/ ERROR(S) DETECTED/<CRLF>
16589	077133	200	106	111	MSG085:	.ASCIIZ	<CRLF>/FILL COUNT(OCTAL)? /
16590	077160	200	113	105	MSG088:	.ASCIIZ	<CRLF>/KERNEL STACK/
16591	077176	200	123	125	MSG089:	.ASCIIZ	<CRLF>/SUPERVISOR STACK/
16592	077220	200	125	123	MSG090:	.ASCIIZ	<CRLF>/USER STACK/
16593	077234	040	111	123	MSG091:	.ASCIIZ	/ IS EMPTY/
16594	077246	122	105	114	MSG092:	.ASCIIZ	/RELOCATED /
16595	077262	102	101	116	MSG093:	.ASCIIZ	/BANK=/
16596	077270	040	040	124	MSG095:	.ASCIIZ	/ TEST=/
16597	077300	200	105	116	MSG101:	.ASCIIZ	<CRLF>/ENTERING KAMIKAZE MODE/
16598	077330	200	114	105	MSG102:	.ASCIIZ	<CRLF>/LEAVING KAMIKAZE MODE/
16599	077357	200	114	105	MSG103:	.ASCIIZ	<CRLF>/LEAVING FS MODE/<CRLF>
16600	077401	032	000		MSG104:	.BYTE	32,0 ;CONTROL Z
16601	077403	200	105	116	MSG105:	.ASCIIZ	<CRLF>/ENTER BANKS - USE NUMBER 200 TO TERMINATE/
16602	077456	200	103	101	MSG106:	.ASCIIZ	<CRLF>/CACHE IS OFF/
16603	077474	200	103	101	MSG107:	.ASCIIZ	<CRLF>/CACHE IS ON (EXCEPT DURING ACTUAL PATTERNS)/
16604	077551	200	117	116	MSG110:	.ASCIIZ	<CRLF>/ONLY SELECTED BANKS WILL BE TESTED/
16605	077615	200	101	114	MSG111:	.ASCIIZ	<CRLF>/ALL BANKS WILL BE TESTED/
16606	077647	113	040	117	MSG112:	.ASCIIZ	/K OF MS11-L/<CRLF>
16607	077664	113	040	117	MSG113:	.ASCIIZ	/K OF MS11-M/<CRLF>
16608	077701	113	040	117	MSG114:	.ASCIIZ	/K OF MS11-P/<CRLF>
16609	077716	200	040	040	MSG117:	.ASCIIZ	<CRLF>' 11/44'
16610	077730	200	040	040	MSG119:	.ASCIIZ	<CRLF>/ NO/
16611	077737	040	103	101	MSG120:	.ASCIIZ	/ CACHE AVAILABLE/
16612	077760	040	103	101	MSG121:	.ASCIIZ	/ CACHE BYPASSED/
16613	100000	200	103	123	MSG122:	.ASCII	<CRLF>/CSR NUMBER /
16614	100014	000			MSG122:	.ASCII	0
16615	100015	040	103	117	MSG122:	.BYTE	0
					.ASCIIZ		/ CONTROLS TOO MANY BANKS/

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 517-2  
 MESSAGES

16616	100046	040	120	101	MSG125:	.ASCIIZ	/ PASSES COMPLETED/
16617	100070	200	120	122	MSG126:	.ASCIIZ	<CRLF>/PROGRAM CSR COULD NOT BE DETERMINED/
16618	100135	200	124	122	MSG127:	.ASCIIZ	<CRLF>/TRACE ENABLED/
16619	100154	200	124	122	MSG128:	.ASCIIZ	<CRLF>/TRACE DISABLED/
16620	100174	200	200	040	MSG008:	.ASCIIZ	<CRLF><CRLF>/ CSR MAP/<CRLF>
16621	100226	200	040	103	MSG000:	.ASCIIZ	<CRLF>' CZMSPA MS11-L/M/P MEMORY DIAGNOSTIC'
16622	100275	200	200	000	MSG129:	.ASCIIZ	<CRLF><CRLF>
16623	100300	120	122	117	EM62:	.ASCIIZ	/PROCESSOR NOT SUPPORTED BY THIS DIAGNOSTIC/
16624						.EVEN	
16630	100354					\$\$END	
16631	100354				END:		
16632	100354	001272				.PRINT	60000-SUPLIMIT ;SUPERVISOR ADDRESSES LEFT
16636		000200				.END	START3



CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M111' 26-APR-82 09:41 PAGE 517-3  
 SYMBOL TABLE

ABORTF	002142	B0	004614	B63	044014	CPUERR=	177766	DH14	073202
ACFLAG	002114	B1	005550	B64	045234	CR	= 000015	DH15	073261
ACTFLA	002344	B10	013462	B65	045242	CRLF	= 000200	DH16	073400
ADDRES	002032	B100	055070	B66	045372	CSR	002146	DH19	073415
ANA2	010002	B101	055076	B67	045406	CSRADD=	172100	DH2	072451
APTDOW	047746	B102	062530	B7	013420	CSRCAS	020174	DH20	073422
APTECC	002420	B103	062534	B70	052536	CSRFBA	002230	DH23	073501
APTFLA	002346	B104	062632	B71	052760	CSRFIR	002224	DH24	073560
APTHAN	015316	B105	062724	B72	053234	CSRHOL	002522	DH25	073627
APTHLT	050014	B106	063010	B73	053530	CSRINC	002324	DH26	073665
APTPAR	002416	B11	014214	B74	053530	CSRINF	002456	DH27	073703
APTSIZ	002440	B12	014302	B75	053626	CSRINT	002234	DH3	072474
BACK	060536	B13	014344	B76	054566	CSRLAS	002226	DH30	073757
BACKGN	041224	B14	014570	B77	054572	CSRLBA	002232	DH5	072530
BAD	002050	B15	015362	CACHKF	002544	CSRL00	002326	DH6	072607
BADPC	002020	B16	017000	CACHKN	002540	CSRMAP	005774	DH7	072654
BADPSW	002030	B17	017250	CACHOF=	104424	CSRNO	002150	DIAGFL	002002
BADSP	002024	B2	006010	CACHON=	104423	CSROUT	044002	DISPLA	002624
BADSTA	042614	B20	017256	CACHVE=	000114	CSRSTU=	000021	DISPRE=	000174
BADXOR	002056	B21	017274	CBCSR =	104474	CSRIS	002316	DISPTB	014770
BAD2	002052	B22	017336	CBITS	002312	CTEST	006656	DOBACK	015352
BAD3	002054	B23	017410	CBREG =	104513	CTLKVE	002144	DONE	006642
BAFPAF	015452	B24	024170	CB1CSR=	104475	DATARG=	177754	DSWR	= 177570
BAFPAR	015560	B25	024174	CHECK	002310	DATBUF	002240	DT1	067132
BAKPAT	002616	B26	024374	CHKDIS=	104504	DBEMSK	002254	DT10	067252
BANK	002100	B27	024402	CHKGEN	044366	DDISP =	177570	DT11	067274
BANKIN	002102	B3	006060	CHKTAB	044474	DEENER=	104421	DT12	067302
BANKMO	046410	B30	024700	CHKTRP	040614	DETAIL	062274	DT13	067306
BANKOK	047446	B31	032410	CHK1DI=	104505	DETFLA	002216	DT14	067330
BAWPAF	015666	B32	034126	CKEND	064050	DETPSW	002214	DT16	067346
BAWPAR	016016	B33	034164	CKSWR =	104410	DETRO	002176	DT17	067376
BGTEST	036244	B34	034606	CLRCR=	104502	DETR1	002200	DT2	067144
BITNO	002320	B35	034644	CLREX	007752	DETR2	002202	DT20	067404
BIT0 =	000001	B36	034660	CLRMEM	007642	DETR3	002204	DT22	067414
BIT1 =	000002	B37	035000	CLR1CS=	104503	DETR4	002206	DT23	067422
BIT10 =	002000	B4	006366	CMD16A	053614	DETR5	002210	DT24	067444
BIT11 =	004000	B40	035160	CMD16L=	000073	DETSP	002212	DT25	067464
BIT12 =	010000	B41	035202	CMD5B	052000	DET1	062764	DT26	067476
BIT13 =	020000	B42	036376	CMD5C	052274	DF1	067544	DT27	067504
BIT14 =	040000	B43	036536	CMD7B	052532	DF11	067667	DT3	067150
BIT15 =	100000	B44	036574	CMD7C	052606	DF13	067700	DT30	067524
BIT2 =	000004	B45	037066	CMD9B	053230	DF14	067710	DT4	067160
BIT3 =	000010	B46	037254	CMD9C	053304	DF15	067717	DT5	067170
BIT4 =	000020	B47	037452	CONFGE	002444	DF16	067726	DT6	067206
BIT5 =	000040	B5	006422	CONFIE	003654	DF2	067545	DT7	067222
BIT6 =	000100	B50	037646	CONF IG	002350	DF3	067571	DUMMY	002174
BIT7 =	000200	B51	037656	CONTFI	002220	DF4	067604	DUMPCS=	000102
BIT8 =	000400	B52	040166	CONTRL=	177746	DF5	067617	ECCDIS=	104470
BIT9 =	001000	B53	040336	CONTS	064220	DF6	067632	ECCINI=	104472
BLOCK1	050016	B54	040352	CONTS1	063576	DF7	067645	ECCTYP	005714
BLOCK2	050036	B55	040416	CONTS2	064222	DF8	067660	ECC1DI=	104471
BLOCK3	050052	B56	043434	CONTS3	056370	DF9	067662	ECC1IN=	104473
BMFLAG	002126	B57	043440	CONTT	064144	DH1	072414	EMTVEC=	000030
BOOT	047524	B6	012574	COUNT	002364	DH10	072764	EM11	070235
BOOT1	047570	B60	043600	CPERRF	061322	DH11	073062	EM12	070257
BRGOBB	036246	B61	043604	CPSAVE	061320	DH12	073100	EM13	070303
BSIZE	002370	B62	044010	CPUBIT	002104	DH13	073105	EM14	070335

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 517-4  
SYMBOL TABLE

EM15	070401	E100	055132	E66	045606	HT	=	000011	LOOP	014742		
EM17	070447	E101	055132	E67	045520	I	=	002446	LOWMAP	046344		
EM19	070507	E102	062576	E7	013562	IBSAVE	=	061316	LSIZE	002374		
EM2	067735	E103	062576	E70	052600	IIII	=	177777	LWDBE	=	000060	
EM20	070561	E104	062662	E71	053062	ILLCSR	=	014124	LWSBE	=	000056	
EM21	070640	E105	062756	E72	053276	IMPTES	=	013064	L0	=	004624	
EM22	070674	E106	063042	E73	053570	INCBNK	=	047510	L1	=	004666	
EM23	070721	E11	014274	E74	053570	INCPAT	=	047464	L10	=	005164	
EM24	070750	E12	014342	E75	053644	INCRPT	=	047464	L100	=	013054	
EM25	071027	E13	014372	E76	054626	INHBAN	=	002534	L101	=	013056	
EM26	071054	E14	014710	E77	054626	INHECC	=	002532	L102	=	013324	
EM27	071125	E15	015450	FASTCI=	177640	INTFLA	=	002134	L103	=	013324	
EM29	071215	E16	017020	FATAL\$	002062	INT64K	=	002136	L104	=	013224	
EM3	067773	E17	017604	FCMD10	053370	INVALI=	=	104511	L105	=	013232	
EM30	071277	E2	006054	FCMD11	053416	IOTVEC=	=	000020	L106	=	013324	
EM32	071407	E20	017564	FCMD12	053440	JMPRL1	=	045740	L107	=	013276	
EM33	071514	E21	017514	FCMD13	053460	KAMIKA	=	002004	L11	=	005172	
EM35	071622	E22	017514	FCMD14	053502	KAMITE	=	026774	L110	=	013304	
EM36	071707	E23	017464	FCMD15	053520	KDIAG	=	000010	L111	=	013330	
EM4	070025	E24	024352	FCMD16	053604	KDPAR0=	=	172360	L112	=	013546	
EM40	071756	E25	024336	FCMD17	053646	KDPAR6=	=	172374	L113	=	013546	
EM5	070073	E26	024560	FCMD18	053662	KDPAR7=	=	172376	L114	=	013536	
EM50	072030	E27	024544	FIELDS	050102	KERNEL=	=	104417	L115	=	013534	
EM51	072064	E3	006176	FINDBA=	000066	KERSTK=	=	002000	L116	=	013540	
EM52	072134	E30	024756	FINT	007174	KFLAG	=	002524	L117	=	014264	
EM53	072161	E31	032474	FIRST	=	060000	KIPAR0=	=	172340	L12	=	005242
EM55	072210	E32	034402	FLIPLO	=	002602	KIPAR4=	=	172350	L120	=	014260
EM56	072231	E33	034370	FLIPWA	=	041074	KIPAR5=	=	172352	L121	=	014252
EM57	072263	E34	035152	FLUSH	=	015062	KIPAR6=	=	172354	L122	=	014256
EM6	070150	E35	035136	FSCMD0	=	050300	KIPDR0=	=	172300	L123	=	014264
EM60	072331	E36	034752	FSCMD1	=	050402	KMAP	=	104422	L124	=	014360
EM61	072373	E37	035122	FSCMD2	=	050512	KPFLAG	=	002112	L125	=	014430
EM62	100300	E4	006564	FSCMD3	=	050660	KSIZE	=	002372	L126	=	014542
EM7	070175	E40	035176	FSCMD4	=	051134	KSTACK	=	002560	L127	=	014742
ENASBE=	104506	E41	035222	FSCMD5	=	051454	LAST	=	157776	L13	=	005236
ENA1SB=	104507	E42	036520	FSCMD6	=	052372	LASTBA	=	002552	L130	=	014700
END	100354	E43	036724	FSCMD7	=	052400	LASTBL	=	002554	L131	=	014616
ENERGI=	104420	E44	036712	FSCMD8	=	052672	LASTER	=	002014	L132	=	014620
ENEXBK	047436	E45	037172	FSCMD9	=	053076	LBLS0	=	000607	L133	=	014660
ERRADD	002454	E46	037424	FSINFL	=	002436	LBLS1	=	000106	L134	=	014674
ERRGEN=	104512	E47	037622	FSPAT	=	052154	LBLS2	=	000601	L135	=	014740
ERRMAX	002550	E5	006554	FSSTAC	=	002302	LBLS3	=	000574	L136	=	014742
ERROR	=	E50	040132	FS1	=	050166	LBLS4	=	000446	L137	=	015056
ERRPC	002016	E51	040116	FS7FLA	=	002442	LBLS5	=	000450	L14	=	005242
ERRPSW	002026	E52	040322	FULLRE	=	002536	LBLS6	=	000023	L140	=	015166
ERRSP	002022	E53	040606	GBLENG=	=	000076	LCSROU=	=	000062	L141	=	015176
ERRVEC=	000004	E54	040576	GETCSR	=	041324	LCSRRE=	=	000100	L142	=	015346
EUFLAG	002130	E55	040564	GETDAT	=	054256	LCSRSA=	=	000076	L143	=	015346
EVEN	002360	E56	043506	GETDA1	=	054354	LEGALC=	=	000010	L144	=	015434
EXBANK	047020	E57	043506	GETDIS	=	060450	LF	=	000012	L145	=	015546
EXCMD3	051066	E6	013012	GOOD	=	002042	LINK1	=	002516	L146	=	015654
EXCMD4	051406	E60	043652	GOOD2	=	002044	LINK2	=	002520	L147	=	016004
EXIT	047632	E61	043652	GOOD3	=	002046	LKS	=	177546	L15	=	005502
EXIT2	047636	E62	044044	GTSWR	=	104407	LOADBA	=	002426	L150	=	016134
E0	004642	E63	044044	HEADER	=	002576	LOADCS=	=	104425	L151	=	016264
E1	005666	E64	045372	HIPAT	=	047500	LOADER=	=	000064	L152	=	016436
E10	013546	E65	045352	HOLDLO=	=	000016	LOADHO	=	002562	L153	=	016566

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 517-5

## SYMBOL TABLE

L154	016740	L237	023266	L324	034714	L420	042160	L502	052450
L155	017020	L24	005762	L325	034732	L421	042204	L503	052452
L156	017056	L240	023536	L326	034740	L422	042306	L504	052564
L157	017066	L241	023546	L327	034764	L423	043220	L505	053062
L16	005502	L242	023546	L33	006142	L424	043226	L506	053046
L160	017514	L243	023556	L330	035122	L425	043464	L507	053016
L161	017476	L244	024322	L331	035022	L426	043464	L51	006544
L162	017546	L245	024270	L332	035034	L427	043526	L510	053146
L163	020016	L246	024322	L333	035066	L43	006422	L511	053150
L164	020126	L247	024366	L334	035052	L430	043534	L512	053262
L165	020264	L25	005772	L335	035064	L431	043630	L513	053552
L166	020312	L250	024530	L336	035110	L432	043630	L514	053554
L167	020316	L251	024476	L337	035076	L433	043672	L515	054020
L17	005614	L252	024530	L34	006152	L434	043700	L516	054034
L170	020320	L253	024742	L340	035110	L435	044022	L517	054040
L171	020364	L254	024742	L341	035222	L436	045206	L520	054056
L172	020420	L255	025050	L342	036570	L437	045222	L521	054204
L173	020424	L256	025022	L343	036574	L44	006452	L522	054206
L174	020426	L257	025100	L344	036660	L440	045234	L523	054252
L175	020574	L260	025270	L345	036676	L441	045234	L524	054476
L176	021250	L261	025622	L346	036702	L442	045336	L525	054604
L177	022116	L262	025112	L351	037020	L443	045366	L526	055110
L2	004774	L263	026150	L352	037156	L444	045510	L527	056426
L20	005642	L264	026250	L354	037330	L445	045510	L530	056430
L200	022126	L265	026274	L355	037400	L446	045510	L531	057260
L201	022126	L266	026346	L357	037534	L447	045470	L532	057272
L202	022136	L267	026416	L36	006250	L45	006524	L533	057310
L203	022174	L27	006144	L360	037604	L450	045510	L534	057312
L204	022204	L270	026460	L362	037760	L451	045552	L535	057332
L205	022204	L271	026524	L363	040036	L452	045552	L536	057344
L206	022214	L272	026564	L364	040054	L453	045602	L537	057362
L207	022244	L273	026624	L365	040102	L454	045722	L54	006600
L21	005660	L274	026664	L37	006254	L455	047156	L540	057364
L210	022250	L275	026724	L370	040232	L456	047364	L541	057400
L211	022300	L276	027016	L371	040274	L457	047546	L542	057602
L212	022310	L277	027024	L373	040412	L46	006514	L543	057610
L213	022310	L3	005016	L374	040416	L460	047652	L544	057636
L214	022320	L30	006124	L375	040470	L461	047656	L545	057650
L215	022364	L300	027030	L376	040472	L462	047664	L546	057666
L216	022374	L301	030646	L377	040530	L463	047700	L547	057700
L217	022374	L302	030644	L4	005034	L464	047712	L55	006642
L220	022404	L303	030646	L40	006320	L465	050124	L550	057712
L221	022454	L304	032344	L400	040550	L466	050134	L551	057734
L222	022464	L305	032402	L401	040554	L467	050270	L552	060002
L223	022464	L306	032462	L405	040646	L47	006516	L553	060062
L224	022474	L31	006116	L406	040644	L470	050330	L554	060076
L225	022526	L310	034160	L407	040666	L471	050334	L555	060100
L226	022542	L311	034164	L41	006362	L472	050364	L556	060204
L227	022552	L312	034260	L410	040666	L473	050376	L557	060274
L23	005764	L313	034336	L411	041004	L474	051564	L56	011562
L230	022552	L314	034354	L412	041052	L475	051624	L560	060304
L231	022632	L315	034360	L413	041464	L476	051664	L561	060612
L232	022642	L32	006124	L414	042050	L477	052072	L562	060612
L233	022642	L320	034630	L415	042152	L5	005112	L563	061052
L234	022660	L321	034640	L416	042142	L50	006540	L564	061006
L235	023170	L322	034752	L417	042150	L500	052106	L565	061032
L236	023222	L323	034702	L42	006416	L501	052120	L566	061052

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 517-6

## SYMBOL TABLE

L567	061072	MMR2	= 177576	MSG055	076513	MTPA25	035664	MT0006	021750
L57	011566	MMR3	= 172516	MSG056	076534	MTPA26	036014	MT0007	022004
L570	061264	MMTRAP	042570	MSG058	076567	MTPB03	027550	MT0010	022046
L571	061134	MMVEC	= 000250	MSG061	076611	MTPB04	027702	MT0011	022102
L572	061262	MONFLG	002272	MSG062	076620	MTPB21	034460	MT0012	022160
L573	061212	MSEEDH	002572	MSG063	076640	MTPB24	035314	MT0013	022264
L574	061224	MSEEDL	002574	MSG064	076651	MTPB25	035706	MT0014	022350
L575	061262	MSG012	100014	MSG065	076661	MTPB26	036030	MT0015	022440
L576	061272	MSG034	076031	MSG066	076673	MTPC03	027610	MT0016	022516
L577	061626	MSG034	076067	MSG067	076742	MTPC21	034514	MT0017	022574
L6	005172	MSG000	100226	MSG070	076751	MTPC24	035330	MT0020	022616
L60	011570	MSG001	074036	MSG073	077002	MTPC25	035746	MT0021	022706
L600	062554	MSG002	074120	MSG075	077020	MTPC26	036064	MT0022	023160
L601	062760	MSG003	074175	MSG076	077052	MTPD03	027626	MT0023	023212
L602	062764	MSG004	074302	MSG077	077073	MTPD21	034550	MT0024	023256
L603	063044	MSG005	074410	MSG079	077107	MTPD25	035612	MT0025	023522
L604	063050	MSG006	074422	MSG085	077133	MTPD26	036104	MT0026	023600
L605	064164	MSG007	074457	MSG088	077160	MTPE25	035634	MT0027	024102
L606	064254	MSG008	100174	MSG089	077176	MTP000	027400	MT0030	024566
L607	064256	MSG009	074471	MSG090	077220	MTP001	027424	MT0031	025070
L61	012402	MSG010	074503	MSG091	077234	MTP002	027456	MT0032	025260
L62	012416	MSG011	074515	MSG092	077246	MTP005	027722	MT0033	025612
L63	012542	MSG012	074603	MSG093	077262	MTP006	027756	MT0034	026000
L64	012520	MSG013	074700	MSG095	077270	MTP007	030156	MT0035	026152
L65	012532	MSG014	074702	MSG101	077300	MTP010	030256	MT0036	026264
L66	012546	MSG015	074704	MSG102	077330	MTP011	030364	MT0037	026336
L67	012776	MSG016	074706	MSG103	077357	MTP012	031162	MT0040	026404
L7	005172	MSG017	074720	MSG104	077401	MTP013	031550	MT0041	026406
L70	012776	MSG018	074731	MSG105	077403	MTP014	032264	MT0042	026450
L71	012776	MSG019	074734	MSG106	077456	MTP015	032510	MT0043	026514
L72	012654	MSG020	074740	MSG107	077474	MTP016	033254	MT0044	026554
L73	012660	MSG021	074761	MSG110	077551	MTP017	034036	MT0045	026614
L74	012776	MSG022	075547	MSG111	077615	MTP020	034114	MT0046	026654
L75	012730	MSG023	075571	MSG112	077647	MTP022	034600	MT0047	026714
L76	012776	MSG025	075605	MSG113	077664	MTP025	035346	MT0999	026760
L77	013050	MSG026	075631	MSG114	077701	MTP030	036122	MT1	017026
MAINT	= 177750	MSG027	075643	MSG117	077716	MTP031	036132	MT2	017032
MAPHO	= 170202	MSG028	075660	MSG119	077730	MTP032	036210	MUT	002106
MAPKER	046706	MSG029	075674	MSG120	077737	MTP033	036242	NC	056430
MAPLO	= 170200	MSG030	075714	MSG121	077760	MTP034	036340	NEMCNT	002066
MAPL1	= 170204	MSG031	075733	MSG122	100000	MTP035	036364	NEWBAN	002304
MAPPER	044534	MSG032	075773	MSG125	100046	MTP036	036526	NEWKER	046640
MASK	002314	MSG033	076012	MSG126	100070	MTP037	036752	NEWLOA	046742
MBERR	014002	MSG035	076115	MSG127	100135	MTP041	037024	NOCH	063560
MEMDON	015010	MSG036	076120	MSG128	100154	MTP042	037176	NOERRO	002424
MFPT	= 000007	MSG037	076137	MSG129	100275	MTP043	037432	NOFSMO	002422
MJPAT	020620	MSG038	076156	MSIZE	002376	MTP044	037626	NONEM	002076
MJTEST	020514	MSG039	076174	MTA030	024600	MTP045	040142	NONEXI	042512
MKCNT	017666	MSG040	076216	MTEST	016752	MTP046	040330	NOOJ	041464
MKCONT	017046	MSG041	076242	MTLA11	030412	MTP047	040670	NOPAR	002074
MKCSRT	020204	MSG042	076267	MTLB11	030424	MTST3	012422	NORES	003664
MKFLAG	002116	MSG043	076305	MTLC11	030436	MT0000	020700	NOSCOP	002434
MKLOOP	017230	MSG046	076327	MTLD11	030532	MT0001	020760	NOSUPE	002452
MKPAT	020434	MSG047	076362	MTPA03	027510	MT0002	021100	NOTAB	002366
MKTEST	020274	MSG048	076401	MTPA04	027646	MT0003	021240	NOTRCE	060220
MMRO	= 177572	MSG049	076441	MTPA21	034430	MT0004	021472	NO22BI	002450
MMR1	= 177574	MSG051	076474	MTPA24	035254	MT0005	021614	NULLFL	002340

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 517-7

## SYMBOL TABLE

NXTCRS	005642	PERR32=	104456	SBENT	020146	SUPDR4	002166	TSTRD1	043230
OLDCAC	002276	PERR33=	104457	SBESYN	034410	SUPDR5	002170	TSTREA=	104510
OLDCSR	002154	PERR34=	104460	SBETES	017670	SUPDR6	002172	TST1	005522
ONES	002600	PERR35=	104461	SCOPE =	000004	SUPLIM	056506	TST2	011404
PADDRE	002034	PERR36=	104462	SDPAR0=	172260	SUPSTK=	000740	TST3	011570
PAFBAF	016146	PERR37=	104463	SDPAR5=	172272	SWAPAT	002620	TST4	012556
PAFBAW	016276	PERR40=	104464	SDPAR6=	172274	SWR	002622	TST5	014742
PARBAF	016450	PERR41=	104465	SDPAR7=	172276	SWREG =	000176	TST6	015014
PARBAW	016600	PERR42=	104466	SEEDHI	002566	SW0 =	000001	TYPDS =	104405
PARCNT	002070	PERR43=	104467	SEEDLO	002570	SW1 =	000002	TYPEIT=	104401
PARITY	042406	PERXOR	057452	SELONL	002000	SW10 =	002000	TYPOC =	104402
PARTHE	002300	PFECDF	061740	SETPAT	047500	SW11 =	004000	TYP0S =	104403
PARVEC=	000114	PFECDH	061700	SHADL1	012452	SW12 =	010000	TYPS0 =	000000
PASFLG	002262	PFECDT	061730	SHUTUP	047664	SW13 =	020000	TYPS1 =	000002
PASSNO	002264	PFECES	061644	SIPAR0=	172240	SW14 =	040000	TYPS2 =	000000
PATERR	002072	PFECWS	061634	SIPAR3=	172246	SW15 =	100000	TYPS3 =	000000
PATPLU	004606	PFLAG	002120	SIPAR5=	172252	SW2 =	000004	TYPS4 =	000000
PATTER	002110	PGMCSR	002526	SIPAR6=	172254	SW3 =	000010	TYPS5 =	000000
PCBUMP	002322	PHEBE	014004	SIPDR0=	172200	SW4 =	000020	TYPS6 =	000002
PCONF1	041352	PHYADD	002036	SIZE =	040000	SW5 =	000040	T12A	033254
PCONFS	041652	PMEMFL	002140	SKIPKA	002006	SW6 =	000100	T12B	033276
PCONF1	041562	PROTYP	003752	SKIPMK	002336	SW7 =	000200	UDPAR0=	177660
PCONF2	041620	PSIZE	002400	SKJ	060246	SW8 =	000400	UDPAR7=	177676
PDP110	042602	PSW =	177776	SKPERR	002064	SW9 =	001000	UIPAR0=	177640
PD1	054476	PTABLE	036732	SKUB	045602	SYSS17	003754	UIPAR1=	177642
PERA05	056644	PWRVEC=	000024	SKUJ	014006	TAG2\$	012042	UIPAR2=	177644
PERBNK	057476	QUICK	002432	SOBK	002556	TAG3\$	012076	UIPAR3=	177646
PERECC	057556	QVFLAG	002342	SOBLEN=	000056	TAG4\$	027124	UIPAR4=	177650
PERRAB	057314	RANODD	036044	SOFTPA	002604	TAG70\$	061744	UIPAR5=	177652
PERRAW	057242	RDCHR =	104411	SOURCE	002306	TAG71\$	061754	UIPAR6=	177654
PERRA3	054044	RDDEC =	104414	SPLTCS	002236	TAG72\$	061764	UIPDR0=	177600
PERRA7	057366	RDLIN =	104412	SSP =	X000006	TAG73\$	062034	UNITOP	002412
PERR01=	104427	RDOCT =	104413	ST =	177776	TAG74\$	062074	UNMAP	046774
PERR02=	104430	READCS=	104426	STACK =	002000	TAG75\$	062106	UNRELO	046056
PERR03=	104431	READON	002404	START	003654	TAG76\$	062120	UPPFLG	002263
PERR04=	104432	REALPA	002274	START1	000300	TAG77\$	062164	USERMA	046556
PERR05	056640	REFRES	035154	START2	000310	TAG78\$	062172	USESTK=	000700
PERR06	056666	REFSUB	035224	STAT3	000200	TAG79\$	062252	USP =	X000006
PERR07=	104433	REGCOP	041064	STAR27	024162	TAG9\$	011670	WARN1	011756
PERR10=	104434	RELENT	045612	STOPOK	002414	TBG4\$	027302	WARN2	027522
PERR11=	104435	RELOCA	045172	STRIPE	002362	TCFIG1	041726	WARN3	027536
PERR12=	104436	RELOC1	045626	SUBAAA	004644	TCFIG2	042066	WARN4	027562
PERR13=	104437	RESREG=	104416	SUBAAB	004774	TCFIG3	042222	WARN5	027576
PERR14=	104440	RESTAR	002612	SUBAAI	012446	TCONF1	041654	WARN6	041306
PERR15=	104441	RESVEC=	000010	SUBAAP	014166	TEMP	002430	WARN6A	041246
PERR16=	104442	RES0	050376	SUBAAR	013372	TESTAD	002406	WARN6B	041300
PERR17=	104443	RES1	050456	SUBAAS	011400	TESTMO	002546	WARN7	024140
PERR20=	104444	RES2	050624	SUCCE5	002330	TIME	002334	WASDBE=	104500
PERR21=	104445	RLFLAG	002124	SUPDOA	002260	TIMEOU	042556	WASSBE=	104476
PERR22=	104446	RRFLAG	002122	SUPDO1	027030	TKVEC =	000060	WAS1DB=	104501
PERR23=	104447	RTNVAL=X000000		SUPDO2	027044	TMFLAG	002132	WAS1SB=	104477
PERR24=	104450	RWCSR	006206	SUPDO3	027206	TOOMAN	002402	WHICHC	053674
PERR25=	104451	SAVCSR	002152	SUPDO4	027222	TOTCSR	002222	WOOPEN	055704
PERR26=	104452	SAVMON	002270	SUPDR0	002156	TRACE	006204	WOOPS	055336
PERR27=	104453	SAVPAR	002266	SUPDR1	002160	TRAPVE=	000034	WOOPSA	055734
PERR30=	104454	SAVREG=	104415	SUPDR2	002162	TSTBAN	012310	WOOPUP	055522
PERR31=	104455	SBEMSK	002250	SUPDR3	002164	TSTDAT	002244	WORST	002564

CZMSPA0 MS11-L/M/P MEMORY DIAG. MACRO M1113 26-APR-82 09:41 PAGE 517-8

## SYMBOL TABLE

XOCHAR	056276	\$DOAGN	015242	\$LF	002645	\$PER10	056716	\$SWREG	065662
XXDPCH	002350	\$DOWN	054726	\$LL =	000105	\$PER11	056746	\$T =	000610
ZEROS	002332	\$DTBL	063516	\$LOADC	042736	\$PER12	056766	\$TESTN	065644
\$APTHD	065740	\$ECCDI	043256	\$LPADR	002606	\$PER13	057010	\$TKB	002630
\$AUTO	002060	\$ECCIN	043304	\$LPERR	002610	\$PER14	057030	\$TKS	002626
\$BANK	002011	\$ECC1D	043272	\$LS =	000000	\$PER15	057052	\$TN =	000007
\$BASE	065714	\$ECC1I	043320	\$MADR1	065672	\$PER16	057074	\$TPB	002634
\$BELL	002637	\$ENASB	043332	\$MADR2	065676	\$PER17	057114	\$TPFLG	002354
\$CACHF	042720	\$ENA1S	043346	\$MADR3	065702	\$PER20	057132	\$TPS	002632
\$CACHN	042674	\$ENDAD	015232	\$MADR4	065706	\$PER21	057150	\$TRAP	065754
\$CBCSR	043360	\$ENERG	042654	\$MAIL	065640	\$PER22	057170	\$TRAP2	065776
\$CBREG	044350	\$ENV	065660	\$MAMS1	065670	\$PER23	057206	\$TRPAD	066016
\$CB1CS	043402	\$ENVM	065661	\$MAMS2	065674	\$PER24	057224	\$TSTM	065744
\$CDW1	065720	\$EOP	015066	\$MAMS3	065700	\$PER25	053762	\$TSTRD	043052
\$CDW2	065722	\$ERFLG	002012	\$MAMS4	065704	\$PER26	057414	\$TTYIN	064704
\$CHARC	056464	\$ERRGE	044100	\$MBADR	065742	\$PER27	057434	\$TYPDS	063312
\$CHKDI	043754	\$ERROR	060522	\$MNEW	064760	\$PER30	054210	\$TYPE	056152
\$CHK1D	043770	\$ERRTB	066262	\$MSGAD	065654	\$PER31	057624	\$TYPEC	056300
\$CKSWR	063536	\$ERRTY	061324	\$MSGLG	065656	\$PER32	057722	\$TYPEX	056466
\$CLRCS	043732	\$ERTTL	002614	\$MSGTY	065640	\$PER33	057770	\$TYPOC	063110
\$CLR1C	043744	\$ESCAP	002356	\$MSWR	064747	\$PER34	060050	\$TYPON	063124
\$CMTAG	002000	\$ETABL	065660	\$MTYP1	065671	\$PER35	060102	\$TYPOS	063064
\$CMTGE	002540	\$ETEND	065740	\$MTYP2	065675	\$PER36	060136	\$T1 =	000000
\$CNTLC	064730	\$EXHAL	047656	\$MTYP3	065701	\$PWDRN	054356	\$T2 =	000607
\$CNTLG	064742	\$ES =	000001	\$MTYP4	065705	\$PWRUP	054732	\$UNIT	065652
\$CNTLK	064136	\$FATAL	065642	\$NOTRA	066010	\$QUES	002643	\$UNITM	065750
\$CNTLU	064735	\$FILLC	002636	\$NULL	002352	\$R =	177777	\$USWR	065664
\$CPUOP	065666	\$FILLS	002353	\$NWTST=	000001	\$RAND	065424	\$VECT1	065710
\$CRLF	002644	\$FS =	000000	\$OCNT	063306	\$RDCHR	064256	\$VECT2	065712
\$DBLK	063526	\$GTSWR	063712	\$OCTVL	065622	\$RDDEC	065142	\$WASDB	043566
\$DB2D	065520	\$HALT	061106	\$OCT8 =	065626	\$RDLIN	064406	\$WASSB	043422
\$DDW0	065724	\$HALT2	066014	\$OMODE	063310	\$RDOCT	064772	\$WASID	043702
\$DDW1	065726	\$HIBTS	065740	\$OVER	060436	\$READC	043032	\$WASIS	043536
\$DDW2	065730	\$HIOCT	065140	\$OS =	000000	\$RESRE	065366	\$XTSTR	060314
\$DDW3	065732	\$ILLUP	055330	\$PASS	065646	\$SAVRE	065330	\$Y\$ =	000000
\$DDW4	065734	\$INVAL	044050	\$PASTM	065746	\$SAVR6	055334	\$ZAP42	015212
\$DDW5	065736	\$ITEMB	002013	\$PATMA	002010	\$SCOPE	060166	\$Z\$ =	000000
\$DEENE	042664	\$IS =	000001	\$PER01	056506	\$STN =	000001	\$ZS =	000000
\$DEVCT	065650	\$KERNE	042644	\$PER02	056534	\$SVLAD	060422	\$SS =	000000
\$DEVN	065716	\$KMAP	045100	\$PER03	056562	\$SVS =	000000	\$ST =	000573
\$DIDDO=	000000	\$KS =	000102	\$PER04	056612	\$SWR =	163000	\$STT =	000601
\$DOAGA	015346	\$L =	000107	\$PER07	056674			\$OFILL	063307

. ABS. 100354 000  
000600 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 26094 WORDS ( 102 PAGES)

DYNAMIC MEMORY: 21558 WORDS ( 82 PAGES)

ELAPSED TIME: 00:29:08

CZMSPA.BIN,CZMSPA/-SP/CR=CZMSPA/ML,CZMSPA.P11



CZMSPA SYMBOL	CREATED BY CROSS REFERENCE VALUE	MACRO ON 26-APR-82 AT 09:54 REFERENCES	PAGE 1 CREF							
ABORTF	002142	#139-5600	*337-11069	421-13214	423-13260	*423-13276	454-14585	454-14588	454-14597	454-14600
ACFLAG	002114	#139-5589	170-6863	172-6903	182-7045	191-7343	193-7359	195-7389	197-7419	199-7456
		201-7497	203-7535	205-7580	207-7618	211-7712	238-8445	238-8474	240-8517	363-11619
ACTFLA	002344	*371-11906	*371-11918	371-11920	393-12463	397-12533				
		#139-5665	*153-5981	167-6724	173-6939	217-7908	217-7923	226-8178	226-8187	226-8202
ADDRES	002032	227-8213	229-8227	229-8237	231-8252	234-8372	253-8814	363-11613	377-12024	466-14898
		#139-5565	*160-6196	*167-6718	*273-9213	*273-9219	*273-9240	*273-9253	*275-9329	*275-9343
		*277-9417	*282-9476	*283-9560	*283-9566	*285-9658	*285-9665	*305-10119	*305-10131	*315-10328
		*315-10343	*316-10380	*317-10418	*323-10594	*323-10610	*339-11095	*421-13213	*421-13213	*421-13239
		*423-13259	*423-13259	423-13262	423-13272	*451-14478	*451-14479	*451-14484	*451-14485	*451-14490
		*451-14491	*451-14496	*451-14497	*451-14504	*451-14512	*451-14517	*451-14517	*451-14522	*451-14527
		*453-14533	*453-14538	*453-14543	*453-14548	*453-14553	*453-14558	*453-14563	*453-14568	*453-14573
		*453-14578	458-14649	*460-14677	*460-14686	473-15088	509-16386	509-16389	509-16392	509-16397
		509-16408								
ANA2	010002	#166-6449								
APTDOW	047746	153-5975	#377-12058	377-12061	*377-12063					
APTECC	002420	#139-5689	*186-7232	186-7237	186-7237	509-16391				
APTFLA	002346	#139-5666	*153-5974	186-7208	189-7325	226-8178	226-8187	226-8202	227-8213	229-8227
		229-8237	231-8252	234-8372	253-8814	363-11613	377-12024	466-14898		
APTHAN	015316	#189-7327	189-7333							
APTHLT	050014	#377-12065								
APTPAR	002416	#139-5688	*186-7229	186-7237	509-16391					
APTSIZ	002440	#139-5697	*153-5971	186-7208						
BACK	060536	#465-14825	466-14915							
BACKGN	041224	222-8115	222-8133	224-8161	232-8268	234-8335	246-8710	246-8715	#328-10835	
BAD	002050	#139-5571	*157-6102	*160-6244	*282-9490	*282-9504	*291-9784	*291-9796	*317-10417	*319-10454
		*321-10494	*321-10503	*322-10541	*322-10551	*323-10596	*323-10609	*324-10660	*324-10668	*325-10722
		*326-10779	*326-10789	*421-13216	*421-13242	*423-13262	*423-13274	*451-14480	*451-14486	*451-14493
		*451-14498	*451-14503	*451-14513	*451-14518	*451-14523	*451-14528	*453-14534	*453-14539	*453-14544
		*453-14549	*453-14554	*453-14559	*453-14564	*453-14569	*453-14574	*453-14579	456-14626	*458-14648
		*460-14678	*460-14687	*460-14688	*460-14707	*460-14714	509-16386	509-16392	509-16400	509-16402
		509-16404	509-16407	509-16409						
BADPC	002020	#139-5560	*339-11122	421-13227	454-14586	454-14598	456-14609	458-14661	460-14676	460-14685
		460-14696	465-14861	465-14862	*465-14866					
BADPSW	002030	#139-5564	*339-11123	460-14707	465-14865					
BADSP	002024	#139-5562	*339-11120	*339-11121	465-14864					
BADSTA	042614	337-11070	337-11078	339-11103	339-11107	339-11110	#339-11119	421-13227	451-14506	454-14586
		454-14598	456-14609	458-14661	460-14676	460-14685	460-14696	460-14706		
BADXOR	002056	#139-5574	*456-14626	*456-14627	509-16392					
BAD2	002052	#139-5572	*421-13243	509-16400						
BAD3	002054	#139-5573	*421-13245	*421-13246	509-16400					
BAFPAF	015452	187-7248	#193-7355	193-7380						
BAFPAF	015560	187-7249	#195-7385	195-7410						
BAKPAT	002616	#141-5742	*147-5871	232-8267	234-8334					
BANK	002100	#139-5583	*169-6739	*169-6751	169-6752	169-6754	169-6761	*169-6780	*170-6860	*170-6866
		*170-6869	*170-6873	170-6875	*172-6899	*173-6936	173-6936	*173-6938	174-6948	*182-7041
		*182-7071	182-7071	*191-7341	*191-7349	191-7349	*193-7356	*195-7386	*197-7416	*199-7453
		*201-7492	*203-7530	*205-7575	*207-7613	211-7681	211-7708	211-7745	*211-7761	213-7817
		213-7842	232-8278	232-8292	234-8345	236-8395	236-8408	*238-8443	238-8447	*238-8465
		238-8465	*238-8472	238-8475	*238-8494	238-8494	*240-8514	*240-8523	240-8523	*240-8533
		242-8551	244-8580	246-8644	255-8822	257-8855	319-10430	321-10473	321-10479	326-10763
		354-11366	361-11576	*363-11617	363-11620	*363-11630	363-11630	365-11723	*365-11724	*365-11728

CZMSPA SYMBOL	CREATED BY CROSS REFERENCE VALUE	MACRO ON 26-APR-82 AT 09:54	PAGE 2 REF							
		REFERENCES								
		*365-11732	369-11857	371-11909	*373-11988	373-11989	386-12209	*386-12214	*386-12218	386-12221
		386-12230	*386-12245	388-12251	*388-12256	*388-12260	388-12263	388-12269	*388-12297	390-12303
		*390-12308	390-12309	390-12311	390-12330	390-12339	390-12354	390-12363	*390-12427	390-12428
		393-12440	*393-12461	*393-12467	393-12467	*393-12474	393-12475	395-12481	*395-12490	395-12491
		395-12498	*395-12504	395-12504	*395-12506	397-12511	*397-12531	*397-12537	397-12537	*397-12544
		397-12545	430-13473	*430-13474	430-13475	*430-13495	432-13514	*432-13515	432-13516	*432-13529
		458-14635	463-14792	472-15032	472-15046	472-15063	473-15070	473-15090	485-15538	
BANKIN	002102	#139-5584	*169-6757	169-6813	170-6862	172-6902	182-7043	211-7687	226-8192	246-8674
		246-8700	329-10859	355-11375	361-11554	363-11625	365-11726	365-11730	*371-11912	
BANKMO	046410	189-7324	363-11623	365-11721	*367-11774	377-12048				
BANKOK	047446	201-7495	203-7533	205-7578	207-7616	*373-11963				
BAWPAF	015666	187-7250	#197-7415	197-7439	197-7447					
BAWPAR	016016	187-7251	#199-7452	199-7476	199-7484					
BGTEST	036244	#313-10264	313-10285							
BITNO	002320	#139-5655	*291-9763	*291-9770	291-9806	*316-10362	*316-10370	316-10393	*325-10697	*325-10705
		325-10734								
BIT0	= 000001	#111-4739	158-6126	159-6159	159-6166	160-6197	160-6235	160-6261	162-6298	162-6311
		166-6521	166-6541	166-6579	166-6600	166-6610	166-6644	166-6652	169-6820	169-6839
		315-10323	315-10338	333-10935	335-10988	341-11136	341-11140	341-11147	344-11208	344-11210
		346-11231	346-11235	346-11239	346-11243	348-11278	348-11280	348-11286	348-11289	350-11313
		350-11315	350-11323	350-11325	363-11641	363-11694	363-11703	365-11742	365-11745	371-11941
		386-12224	388-12265	458-14638	462-14743	465-14852	465-14854			
BIT1	= 000002	#111-4738	151-5944	158-6133	158-6136	159-6160	159-6166	162-6298	162-6311	162-6319
		163-6363	163-6393	166-6521	166-6541	166-6548	166-6654	182-7084	182-7107	182-7119
		325-10709	335-10954	335-10989	335-11013	346-11223	346-11227	346-11239	346-11243	346-11248
		346-11253	352-11339	352-11344	371-11916					
BIT10	= 002000	#111-4729	315-10331							
BIT11	= 004000	#111-4728	158-6128	166-6581	182-7060	279-9454	283-9601	287-9703	371-11953	
BIT12	= 010000	#111-4727	155-6038	155-6039	155-6045	166-6577	182-7060	182-7090	182-7121	226-8193
		335-10952	363-11705	371-11950	390-12361	478-15221	478-15222	478-15237		
BIT13	= 020000	#111-4726	155-6038	155-6045	157-6095	157-6096	174-6950	174-6966	189-7286	246-8680
		246-8694	273-9282	279-9451	282-9487	283-9598	287-9700	321-10492	321-10501	323-10592
		326-10777	326-10787	343-11172	344-11198	354-11369	363-11627	363-11674	365-11731	365-11736
		365-11754	365-11767	375-12016	478-15221	478-15237				
BIT14	= 040000	#111-4725	160-6236	166-6629	166-6631	255-8844	297-9965	297-9966	321-10476	344-11217
		355-11401	355-11403	363-11680	365-11739	367-11798	367-11814	371-11947	400-12611	400-12627
		426-13326	426-13332	429-13429	429-13433	430-13476	432-13517			
BIT15	= 100000	#111-4724	157-6115	163-6405	166-6594	166-6620	213-7811	213-7829	213-7832	213-7836
		273-9224	324-10658	325-10721	326-10766	344-11207	344-11217	350-11302	363-11626	363-11642
		363-11680	363-11689	363-11710	365-11727	365-11739	367-11798	367-11814	406-12646	426-13326
		429-13433	460-14697	463-14797	473-15093					
BIT2	= 000004	#111-4737	154-6018	154-6019	154-6020	160-6236	162-6300	166-6523	166-6567	291-9774
		291-9792	315-10323	316-10374	325-10709	337-11072	346-11248	346-11253	352-11339	352-11344
		355-11429								
BIT3	= 000010	#111-4736	154-6018	154-6019	154-6022	157-6093	160-6211	160-6221	160-6251	160-6269
BIT4	= 000020	#111-4735	137-5526	145-5821	145-5829	157-6093	166-6594	166-6620	189-7320	213-7811
		213-7836	273-9224	321-10482	344-11207	348-11267	348-11287	460-14697		
BIT5	= 000040	#111-4734	153-5967	166-6514	166-6585	166-6615	189-7320	363-11685	363-11693	365-11749
		365-11752	371-11956	375-12007	377-12052					
BIT6	= 000100	#111-4733	172-6919	331-10875	331-10912	335-11037	363-11645	371-11913	390-12350	390-12423
		393-12456	393-12472	397-12527	397-12542					
BIT7	= 000200	#111-4732	153-5970	154-5989	154-5990	170-6871	273-9273	279-9440	283-9587	287-9689



CZMSPA SYMBOL SYMBOL	CREATED BY CROSS REFERENCE VALUE	MACRO REFERENCES	ON 26-APR-82 AT 09:54	PAGE 3 CREF						
BIT8	= 000400	331-10874	344-11193	363-11641	363-11645	371-11935	390-12349	393-12455	397-12526	
BIT9	= 001000	#111-4731	184-7157	363-11644	371-11929	486-15608				
BLOCK1	050016	#111-4730	174-6976	174-6991	184-7158	371-11932				
		169-6748	220-8022	220-8044	220-8070	222-8094	222-8097	222-8123	222-8143	232-8296
		232-8299	232-8303	232-8306	234-8355	236-8412	236-8420	238-8438	240-8507	244-8609
		246-8678	328-10848	344-11203	#379-12075					
BLOCK2	050036	#379-12080								
BLOCK3	050052	222-8098	222-8099	222-8124	222-8144	234-8356	234-8357	236-8413	236-8415	242-8553
		246-8647	363-11681	365-11740	367-11799	367-11815	#379-12084	430-13479	430-13482	
BMFLAG	002126	#139-5594	197-7421	199-7458	203-7537	207-7620	363-11619	*371-11907	*371-11927	*371-11943
		*371-11946								
BOOT	047524	#375-11993	483-15485	486-15612						
BOOT1	047570	375-12001	375-12006	#375-12008						
BRGOBB	036246	#313-10265	313-10269	313-10303						
BSIZE	002370	#139-5678	*184-7171	*184-7172	*184-7173	*184-7174	184-7175			
CACHKF	002544	#141-5719	*145-5821	341-11154	382-12159					
CACHKN	002540	#141-5718	*145-5823	154-6026	*154-6026	*154-6027	341-11144	341-11146	341-11151	380-12106
		382-12157	382-12159	382-12162	399-12575	*399-12575	*399-12576	399-12581	*399-12581	425-13296
		429-13453	434-13591	434-13651						
CACHOF	= 104424	#110-4615	154-6025	162-6270	166-6510	167-6708	169-6770	169-6787	169-6804	169-6830
		172-6897	211-7724	213-7795	255-8846	257-8883	282-9473	291-9756	296-9927	316-10356
		317-10409	321-10472	322-10525	323-10567	324-10646	325-10687	326-10762	380-12110	399-12574
		434-13595								
CACHON	= 104423	#110-4614	147-5890	163-6413	166-6686	167-6692	167-6711	169-6778	169-6791	169-6812
		169-6835	173-6944	211-7736	213-7847	213-7855	255-8848	257-8885	282-9512	291-9810
		296-9905	316-10397	317-10421	321-10514	322-10557	323-10637	324-10677	325-10739	326-10797
		399-12582	425-13299							
CACHVE	= 000114	#111-4754	111-4755							
CBCSR	= 104474	#110-4658								
CBITS	002312	#139-5652	*166-6550	*166-6553	166-6555					
CBREG	= 104513	#110-4673	174-6977	174-6992	282-9477	291-9759	291-9787	316-10358	321-10483	321-10511
		322-10526	322-10544	324-10647	324-10663	325-10692	326-10767			
CB1CSR	= 104475</									

CZMSPA  
SYMBOL CROSS REFERENCE  
SYMBOL VALUE

CREATED BY MACRO ON 26-APR-82 AT 09:54

PAGE 4  
CREF

SYMBOL	VALUE	REFERENCES
CMD16A	053614	326-10794 421-13236
CMD5B	052000	400-12600 #400-12623
CMD5C	052274	#390-12354 390-12379
CMD7B	052532	390-12335 390-12346 #390-12422
CMD7C	052606	#393-12461 393-12469
CMD9B	053230	393-12452 #393-12471
CMD9C	053304	#397-12531 397-12539
CONFGE	002444	397-12523 #397-12541
CONFIE	003654	#139-5699 *166-6602 *166-6646 *169-6821 *169-6840 *182-7067 *182-7146
CONF IG	002650	#143-5785 151-5940
		#143-5782 151-5938 *154-5989 *154-5990 166-6514 *166-6570 *166-6574 *166-6577 *166-6581
		*166-6585 166-6586 *166-6586 166-6587 *166-6587 *166-6600 *166-6601 166-6603 *166-6603
		166-6604 *166-6604 166-6613 *166-6613 166-6614 *166-6614 *166-6615 *166-6644 *166-6645
		*167-6720 169-6762 *169-6820 *169-6822 *169-6839 *169-6843 *170-6871 172-6912 *172-6919
		173-6924 174-6976 174-6991 *174-7018 *174-7019 182-7046 182-7054 *182-7060 *182-7061
		182-7083 182-7090 182-7107 182-7119 182-7121 184-7156 184-7157 184-7158 184-7178
		189-7285 211-7688 211-7699 226-8193 246-8680 *246-8694 246-8701 315-10331 329-10860
		333-10935 335-10952 335-10954 335-10960 335-10981 335-10983 335-11012 335-11033 335-11037
		*354-11369 *363-11626 *363-11627 363-11641 363-11642 363-11643 363-11644 363-11645 363-11645
		*363-11674 363-11699 363-11705 363-11707 *365-11727 *365-11731 *365-11736 365-11753 371-11913
		371-11917 371-11922 371-11929 371-11932 371-11935 371-11941 371-11947 371-11950 371-11953
		371-11956 390-12314 390-12341 390-12361 395-12493 395-12499 *400-12611 *400-12627 *458-14638
		*458-14639 *458-14641 458-14642
CONTFL	002220	#139-5625 *187-7243 *193-7370 *195-7400 *197-7432 *199-7469 *201-7503 *203-7543 *205-7586
CONTRL	= 177746	*207-7626 209-7662 *211-7759
		#111-4760 145-5817 154-6005 *154-6018 *154-6019 154-6020 154-6022 *341-11146 *341-11147
		*341-11154 380-12108 *382-12164 425-13298 *429-13455 434-13593 *434-13653 509-16398
CONTS	064220	483-15459 #485-15547
CONTS1	063576	483-15444 #483-15449 485-15554
CONTS2	064222	#485-15549 485-15550 485-15556
CONTS3	056370	#434-13629 434-13630 434-13636
CONTT	064144	462-14738 478-15187 483-15454 #485-15522
COUNT	002364	#139-5676 *296-9906 296-9909 *296-9909 *296-9910 296-9911 *296-9918 *296-9925 297-9930
		*297-9930 *297-9931 297-9932 *297-9951
CPERRF	061322	*145-5857 *145-5859 462-14740 465-14849 #466-14919
CPSAVE	061320	*462-14742 462-14743 *465-14851 465-14852 #466-14918 470-15011
CPUBIT	002104	#139-5585 *151-5944 167-6720 169-6822 169-6843 184-7156 184-7178 335-10981 363-11643
		371-11917 390-12314
CPUERR	= 177766	#111-4766 *145-5860 *151-5956 *164-6446 *167-6721 *169-6781 *382-12176 *382-12180 *384-12192
		*384-12204 *386-12246 *388-12298 *462-14773 *466-14897 *476-15179 509-16390 509-16398
CR	= 000015	#111-4694 434-13642
CRLF	= 000200	#111-4695 434-13571 513-16457 517-16503 517-16504 517-16505 517-16507 517-16509 517-16510
		517-16511 517-16512 517-16513 517-16519 517-16523 517-16524 517-16525 517-16526 517-16527
		517-16528 517-16529 517-16530 517-16531 517-16532 517-16533 517-16534 517-16535 517-16536
		517-16537 517-16538 517-16539 517-16540 517-16541 517-16542 517-16543 517-16545 517-16546
		517-16547 517-16548 517-16549 517-16550 517-16551 517-16552 517-16553 517-16554 517-16555
		517-16556 517-16559 517-16560 517-16561 517-16562 517-16563 517-16564 517-16565 517-16566
		517-16567 517-16567 517-16568 517-16570 517-16573 517-16574 517-16575 517-16579 517-16581
		517-16583 517-16584 517-16585 517-16586 517-16586 517-16587 517-16588 517-16589 517-16590
		517-16591 517-16592 517-16597 517-16598 517-16599 517-16599 517-16601 517-16602 517-16603
		517-16604 517-16605 517-16606 517-16607 517-16608 517-16609 517-16610 517-16613 517-16617
		517-16618 517-16619 517-16620 517-16620 517-16620 517-16621 517-16622 517-16622

CZMSPA  
SYMBOL  
SYMBOL  
CSR

CREATED BY  
CROSS REFERENCE  
VALUE  
002146

MACRO ON 26-APR-82 AT 09:54

PAGE 5  
CREF

## REFERENCES

		#139-5602	*160-6209	*160-6219	*160-6245	*160-6267	166-6536	166-6620	*174-6950	*174-6966
		*174-6978	*174-6980	*174-6993	*174-6995	213-7811	213-7836	273-9224	273-9224	*282-9478
		*282-9487	282-9488	282-9490	*291-9775	*291-9781	291-9782	291-9784	*291-9789	291-9793
		291-9796	*315-10323	*315-10338	*316-10375	319-10448	*321-10486	*321-10492	321-10493	321-10494
		*321-10498	*321-10501	321-10502	321-10503	*322-10532	*322-10538	322-10539	322-10541	*322-10547
		322-10548	322-10551	*323-10580	*323-10588	323-10591	*323-10604	323-10607	323-10609	*324-10650
		324-10658	324-10660	*324-10666	324-10667	324-10668	*324-10674	*325-10710	325-10721	325-10722
		*326-10768	*326-10771	*326-10777	326-10778	326-10779	*326-10787	326-10788	326-10789	*343-11172
		343-11173	*343-11181	344-11207	344-11207	*344-11218	*346-11223	*346-11227	*346-11231	*346-11235
		*346-11239	*346-11243	*346-11247	*346-11248	*346-11252	*346-11253	348-11267	348-11287	350-11302
		350-11321	*352-11330	*352-11334	*352-11339	*352-11344	355-11392	*355-11428	*355-11429	*384-12195
		*421-13229	421-13235	426-13343	*428-13410	460-14697	478-15190	478-15199	*478-15205	509-16395
		509-16396	509-16397	509-16405	509-16406	509-16409				
CSRADD	= 172100	*113-4971	157-6085	*162-6300	*162-6303	*162-6305	*162-6308	162-6310	*163-6353	163-6368
		*163-6385	163-6398	*166-6523	*166-6526	*166-6624	166-6626	*166-6629	166-6630	*166-6631
		*343-11173	343-11181	344-11187	*355-11401	355-11402	*355-11403			
CSRCAS	020174	211-7733	*215-7861							
CSRFBA	002230	*139-5630	*211-7682	211-7707	*211-7752					
CSRFIR	002224	*139-5628	*211-7707	211-7713	211-7715	*211-7749	211-7749			
CSRHOL	002522	*139-5708	*211-7692	*211-7702	211-7704					
CSRINC	002324	*139-5657								
CSRINF	002456	*139-5704	*157-6092	*157-6093	157-6101	157-6102	*158-6126	*158-6133	*158-6136	159-6158
		159-6159	159-6160	159-6166	160-6197	*160-6211	*160-6221	160-6235	*160-6251	160-6261
		*160-6269	162-6298	162-6311	162-6319	163-6363	163-6393	166-6521	166-6541	166-6548
		166-6571	166-6652	166-6654	182-7038	*182-7049	182-7077	182-7079		
CSRINT	002234	*139-5632	*211-7684	*211-7698	211-7751					
CSRLAS	002226	*139-5629	*211-7713	*211-7714	211-7740					
CSRLBA	002232	*139-5631	*211-7683	*211-7696	211-7749					
CSRLOO	002326	*139-5658	*211-7686	*211-7697	211-7755					
CSRMAP	005774	157-6117	*159-6140							
CSRNO	002150	*139-5603	*160-6195	*162-6286	*162-6292	*163-6347	*163-6360	*163-6379	*163-6390	*166-6502
		*166-6509	166-6623	*172-6915	172-6916	172-6917	*173-6927	*211-7704	*211-7705	*246-8705
		246-8706	*329-10864	343-11162	343-11168	343-11170	343-11180	344-11188	344-11194	344-11196
		*348-11263	*348-11273	348-11273	*350-11298	*350-11308	350-11308	*354-11353	*354-11359	354-11359
		355-11400	380-12109	*382-12156	390-12338	*390-12345	390-12369	*390-12371	*390-12430	*406-12648
		*406-12662	425-13302	*426-13339	*426-13346	426-13346	*428-13407	*428-13414	428-13414	*429-13449
		478-15190	*478-15195	*478-15203	478-15203	*478-15205	509-16406			
		346-11224	346-11232	346-11240	346-11249	352-11331	352-11340	#354-11349		
CSROUT	044002	*139-5654	*160-6202	*160-6203	160-6213	160-6217	160-6218			
CSRIS	002316	157-6118	*162-6279							
CTEST	006656	*139-5601	*211-7731	*211-7732	*217-7920	*217-7921	*219-7972	*219-7973	483-15463	
CTLKVE	002144	*111-4763	154-6009							
DATARG	= 177754	*139-5634	*267-9037	*267-9038	267-9039	267-9040	267-9042	267-9047	267-9051	*267-9053
DATBUF	002240	*267-9053	*267-9056	*267-9057	267-9058	267-9059	267-9061	267-9066	267-9070	*267-9072
		*267-9072	*273-9177	*273-9178	273-9183	273-9184	273-9265	*273-9267	*273-9267	*275-9298
		*275-9299	275-9302	275-9303	275-9362	*275-9364	*275-9364	*277-9380	*277-9381	277-9386
		277-9387	*283-9518	*283-9519	283-9524	283-9525	*285-9619	*285-9620	285-9625	285-9626
		451-14514	451-14519							
DBEMSK	002254	*139-5637	*277-9384	*277-9385	277-9393	277-9395	277-9403	277-9405	279-9425	*279-9427
		*279-9427	*279-9444	279-9448	*279-9450	279-9451	*279-9456	*279-9457	*283-9522	*283-9523
		283-9531	283-9533	283-9541	283-9543	283-9573	*283-9575	*283-9575	*283-9591	283-9595
		*283-9597	283-9598	*283-9603	*283-9604	*285-9623	*285-9624	285-9631	285-9633	285-9641

CZMSPA  
SYMBOL CROSS REFERENCE  
SYMBOL VALUE

CREATED BY MACRO ON 26-APR-82 AT 09:54

PAGE 6  
CREF

SYMBOL	VALUE	REFERENCES	287-9675	*287-9677	*287-9677	*287-9693	287-9697	*287-9699	287-9700	*287-9705
DDISP	= 177570	285-9643								
DEENER	= 104421	*287-9706								
DETAIL	062274	#111-4688	141-5747	151-5951						
DETFLA	002216	#110-4611	155-6033	375-12004	377-12049					
DETPSW	002214	470-15002	#476-15150	476-15182						
DETRO	002176	#139-5624	466-14894	470-15001	*476-15151	476-15152	476-15154	*478-15252		
DETR1	002200	#139-5623	509-16394							
DETR2	002202	#139-5616	*476-15159	476-15168	509-16394					
DETR3	002204	#139-5617	476-15160	509-16394						
DETR4	002206	#139-5618	509-16394							
DETR5	002210	#139-5619	509-16394							
DETSP	002212	#139-5620	509-16394							
DET1	062764	#139-5621	509-16394							
DF1	067544	#139-5622	509-16394							
DF11	067667	478-15220	#478-15237							
DF13	067700	508-16318	#511-16413							
DF14	067710	500-16115	502-16182	502-16187	502-16192	504-16214	504-16219	504-16234	508-16328	#511-16422
DF15	067717	502-16207	506-16286	508-16358	508-16363	508-16383	#511-16423			
DF16	067726	506-16301	508-16323	508-16343	508-16353	508-16368	508-16373	#511-16424		
DF2	067545	#511-16425								
DF3	067571	508-16333	508-16338	508-16348	#511-16426					
DF4	067604	500-16120	500-16135	500-16140	500-16145	500-16150	502-16167	502-16197	504-16224	504-16229
DF5	067617	504-16239	504-16244	506-16261	506-16306	508-16313	508-16378	#511-16414		
DF6	067632	502-16157	504-16254	506-16271	506-16276	506-16291	506-16296	#511-16415		
DF7	067645	502-16162	#511-16416							
DF8	067660	502-16172	502-16202	506-16281	#511-16417					
DF9	067662	502-16177	#511-16418							
DH1	072414	504-16249	#511-16419							
DH10	072764	506-16266	#511-16420							
DH11	073062	500-16125	500-16130	#511-16421						
DH12	073100	500-16118	#515-16476							
DH13	073105	502-16165	#515-16483							
DH14	073202	504-16227	#515-16484							
DH15	073261	504-16222	#515-16485							
DH16	073400	500-16113	502-16180	502-16185	502-16190	504-16212	504-16217	504-16232	508-16326	#515-16486
DH17	073415	504-16237	#515-16488							
DH2	072451	504-16247	#515-16489							
DH2C	073422	504-16252	506-16264	#515-16491						
DH23	073501	#515-16492								
DH24	073560	508-16311	#515-16477							
DH25	073627	506-16284	508-16356	508-16361	508-16381	#515-16493				
DH26	073665	502-16205	#515-16494							
DH27	073703	506-16299	508-16321	508-16341	508-16351	508-16366	508-16371	#515-16495		
DH3	072474	500-16148	506-16259	#515-16496						
DH30	073757	502-16195	#515-16497							
DH5	072530	508-16316	#515-16498							
DH6	072607	500-16123	500-16128	#515-16478						
DH7	072654	508-16331	508-16336	508-16346	#515-16499					
		500-16133	500-16138	500-16143	504-16242	508-16376	#515-16479			
		506-16304	#515-16480							
		502-16155	502-16160	502-16170	502-16175	502-16200	506-16269	506-16274	506-16279	506-16289
		506-16294	#515-16481							

CZMSPA		CREATED BY	MACRO	ON 26-APR-82 AT 09:54	PAGE 7					
SYMBOL	CROSS REFERENCE	VALUE	REFERENCES		CREF					
DIAGFL		002002	#139-5550	*233-8324	*233-8326	296-9923				
DISPLA		002624	#141-5747	*151-5951	*151-5958	429-13447	463-14802			
DISPRE	=	000174	#113-4967	151-5958	*429-13448	*463-14803				
DISPTB		014770	187-7246	#187-7248						
DOBACK		015352	187-7257	187-7268	#191-7339					
DONE		006642	160-6261	#160-6271						
DSWR	=	177570	#111-4687	141-5746	151-5950					
DT1		067132	500-16119	#509-16386						
DT10		067252	502-16166	#509-16394						
DT11		067274	504-16228	#509-16395						
DT12		067302	504-16223	#509-16396						
DT13		067306	500-16114	502-16181	502-16186	502-16191	504-16213	504-16218	504-16233	508-16327 #509-16397
DT14		067330	*145-5824	*145-5836	*145-5842	504-16238	#509-16398			
DT16		067346	504-16248	#509-16399						
DT17		067376	504-16253	506-16265	#509-16401					
DT2		067144	#509-16387							
DT20		067404	508-16312	#509-16402						
DT22		067414	#509-16403							
DT23		067422	502-16206	506-16285	508-16357	508-16362	508-16382	#509-16404		
DT24		067444	506-16300	508-16322	508-16342	508-16352	508-16367	508-16372	#509-16405	
DT25		067464	500-16149	506-16260	#509-16406					
DT26		067476	502-16196	#509-16407						
DT27		067504	508-16317	#509-16408						
DT3		067150	500-16124	#509-16388						
DT30		067524	508-16332	508-16337	508-16347	#509-16409				
DT4		067160	500-16129	#509-16389						
DT5		067170	*145-5835	*145-5841	500-16134	500-16139	500-16144	504-16243	508-16377	#509-16390
DT6		067206	506-16305	#509-16391						
DT7		067222	502-16156	502-16161	502-16171	502-16176	502-16201	506-16270	506-16275	506-16280 506-16290
			506-16295	#509-16392						
DUMMY		002174	#139-5614	509-16392	509-16392	509-16393	509-16393	509-16393	509-16393	509-16397 509-16397
			509-16399	509-16399	509-16400	509-16400	509-16401	509-16403	509-16404	509-16404
			509-16404	509-16404	509-16405	509-16405	509-16405	509-16405	509-16405	509-16408 509-16408
			509-16408	509-16408	509-16408	509-16409	509-16409	509-16409	509-16409	
ECCDIS	=	104470	#110-4654	217-7909	240-8512	390-12367	393-12445	397-12516		
ECCINI	=	104472	#110-4656	167-6693	#167-6727	#173-6942	211-7758	#217-7926	240-8529	240-8536 240-8540
			375-12000	377-12041	#421-13253					
ECCTYP		005714	157-6097	#158-6125						
ECC1DI	=	104471	#110-4655	174-7011	213-7796	213-7819	273-9202	273-9209	273-9286	275-9312 275-9367
			277-9407	279-9459	283-9546	283-9606	285-9646	287-9708	303-10088	305-10096 305-10106
			325-10714							
ECC1IN	=	104473	#110-4657	213-7846	213-7854	282-9493	282-9496	303-10053	305-10099	
EMTVEC	=	000030	#111-4749	*147-5876	*147-5877					
EM11		070235	502-16154	502-16159	502-16169	502-16174	#513-16441			
EM12		070257	502-16164	#513-16442						
EM13		070303	502-16179	508-16355	#513-16443					
EM14		070335	502-16184	#513-16444						
EM15		070401	502-16189	#513-16445						
EM17		070447	502-16199	#513-16446						
EM19		070507	504-16211	508-16350	#513-16447					
EM2		067735	500-16117	#513-16435						
EM20		070561	504-16216	508-16340	508-16345	508-16365	#513-16448			

CZMSPA      CREATED BY      MACRO      ON 26-APR-82 AT 09:54      PAGE 8  
 SYMBOL      CROSS REFERENCE      CREF  
 SYMBOL      VALUE      REFERENCES

EM21	070640	504-16226 #513-16449							
EM22	070674	504-16231 508-16370	508-16380	#513-16450					
EM23	070721	504-16241 #513-16451							
EM24	070750	500-16112 #513-16452							
EM25	071027	504-16246 508-16330	#513-16453						
EM26	071054	504-16251 #513-16454							
EM27	071125	506-16263 #513-16455							
EM29	071215	506-16273 508-16325	#513-16456						
EM3	067773	500-16122 #513-16436							
EM30	071277	506-16278 508-16360	#513-16457						
EM32	071407	506-16288 #513-16459							
EM33	071514	506-16293 #513-16460							
EM35	071622	506-16268 #513-16461							
EM36	071707	506-16303 #513-16462							
EM4	070025	500-16127 #513-16437							
EM40	071756	508-16310 #513-16463							
EM5	070073	500-16132 #513-16438							
EM50	072030	502-16204 #513-16464							
EM51	072064	506-16298 #513-16465							
EM52	072134	506-16258 #513-16466							
EM53	072161	500-16147 #513-16467							
EM55	072210	502-16194 #513-16468							
EM56	072231	508-16315 #513-16469							
EM57	072263	508-16335 #513-16470							
EM6	070150	500-16137 #513-16439							
EM60	072331	506-16283 #513-16471							
EM61	072373	508-16320 #513-16472							
EM62	100300	508-16375 #517-16623							
EM7	070175	500-16142 #513-16440							
ENASBE	= 104506	#110-4668 167-6725	173-6940	217-7924	421-13251				
ENA1SB	= 104507	#110-4669 303-10057	303-10090	305-10101	#325-10716				
END	= 100354	#517-16631							
ENERGI	= 104420	#110-4610 155-6060	189-7321						
ENEXBK	047436	371-11928 371-11951	371-11957	#371-11959					
ERRADD	002454	#139-5703 213-7814	213-7839	*355-11422					
ERRGEN	= 104512	#110-4672 213-7813	213-7838	273-9230	273-9256	279-9422	315-10334		
ERRMAX	002550	#141-5721 458-14642							
ERROR	= 104000	#111-4682 154-5994	154-6013	154-6016	157-6103	160-6210	160-6220	160-6247	#160-6249
		160-6268 167-6715	167-6719	174-7020	186-7238	273-9225	275-9330	277-9418	282-9484
		282-9491 282-9505	291-9785	291-9797	303-10062	305-10120	305-10132	315-10329	315-10344
		316-10384 317-10419	321-10496	321-10506	322-10542	322-10552	323-10598	323-10612	324-10661
		324-10670 325-10724	325-10746	325-10751	326-10782	326-10792	337-11079	339-11104	339-11108
		339-11111 382-12175	384-12191	384-12200	421-13248	454-14589	#454-14591	454-14601	#454-14603
		456-14612 458-14659	458-14664	458-14667	458-14670	460-14681	460-14692	460-14698	#460-14700
		460-14709 460-14716	462-14745	476-15170	476-15176				
ERRPC	002016	#139-5559 *465-14839	*465-14840	465-14843	*465-14862	*465-14863	465-14868	468-14936	470-15011
		509-16386 509-16387	509-16388	509-16389	509-16390	509-16391	509-16392	509-16395	509-16397
		509-16399 509-16401	509-16402	509-16403	509-16404	509-16405	509-16406	509-16407	509-16408
		509-16409							
ERRPSW	002026	#139-5563 *465-14842	*465-14865	476-15167					
ERRSP	002022	#139-5561 *465-14841	*465-14864	476-15166					
ERRVEC	= 000004	#111-4742 *147-5886	*147-5887	462-14765	*462-14766	*462-14768	*462-14774		

CZMSPA		CREATED BY	MACRO	ON 26-APR-82 AT 09:54	PAGE 9						
SYMBOL	CROSS REFERENCE	VALUE	REFERENCES	REF							
EUFLAG		002130	#139-5595	222-8076	355-11382	355-11397					
EVEN		002360	#139-5674	*296-9894	296-9895	*297-9957	297-9957				
EXBANK		047020	170-6861	172-6901	182-7042	191-7342	193-7358	195-7388	197-7418	199-7455	201-7494
			203-7532	205-7577	207-7615	238-8444	238-8473	240-8515	240-8534	363-11618	365-11725
			365-11729	*371-11883	390-12332	390-12429	393-12462	393-12476	397-12532	397-12546	
EXCMD3		051066	386-12235	386-12238	386-12242	*386-12244					
EXCMD4		051406	388-12278	388-12282	*388-12296						
EXIT		047632	*377-12022	462-14756	466-14900	486-15611					
EXIT2		047636	*377-12024								
E1		005666	*157-6111								
E2		006054	*159-6155								
E3		006176	*159-6176								
E31		032474	*282-9508								
E32		034402	*291-9808								
E33		034370	*291-9806								
E4		006564	*160-6260								
E43		036724	*316-10395								
E44		036712	*316-10393								
E45		037172	*319-10460								
E46		037424	*321-10512								
E47		037622	*322-10556								
E5		006554	*160-6259								
E50		040132	*323-10634								
E51		040116	*323-10632								
E52		040322	*324-10675								
E53		040606	*325-10737								
E54		040576	*325-10736								
E55		040564	*325-10734								
FASTCI	=	177640	*111-4796	169-6776	169-6810	255-8847	344-11204	379-12076			
FATALS		002062	*139-5576	*167-6715	*167-6719	*303-10062	*337-11079	*339-11104	*339-11108	*339-11111	466-14898
			468-14953								
FCMD10		053370	380-12130	*399-12551							
FCMD11		053416	380-12131	*399-12561							
FCMD12		053440	380-12132	*399-12566							
FCMD13		053460	380-12133	*399-12572							
FCMD14		053502	380-12134	*399-12579							
FCMD15		053520	380-12139	*400-12598							
FCMD16		053604	380-12140	*400-12618							
FCMD17		053646	380-12141	*402-12632							
FCMD18		053662	380-12142	*404-12638							
FIELDS		050102	*380-12097	483-15451							
FINT		007174	162-6332	*163-6339							
FIRST	=	060000	*113-4974	169-6765	169-6798	169-6828	172-6900	211-7682	219-7965	219-7966	220-8014
			220-8027	220-8051	222-8080	222-8116	222-8134	224-8162	232-8273	234-8339	236-8389
			238-8448	238-8477	240-8519	242-8553	242-8558	242-8559	244-8585	244-8615	244-8622
			246-8647	246-8650	246-8651	246-8652	246-8653	246-8670	249-8751	282-9476	289-9723
			296-9907	296-9926	297-9962	297-9964	297-9965	315-10322	319-10434	323-10594	323-10610
			328-10838	*377-12061	*377-12062	*377-12063	386-12223	388-12264	390-12359	390-12360	430-13477
			430-13478	430-13479	*430-13480	*430-13481	430-13482	430-13484	*432-13518	*432-13519	432-13524
			473-15089								
FLIPLO		002602	*141-5734	*147-5865	222-8106	*327-10810	*327-10811	327-10812	327-10814	327-10816	
FLIPWA		041074	222-8079	*327-10808							



CZMSPA CREATED BY MACRO ON 26-APR-82 AT 09:54 PAGE 10  
SYMBOL CROSS REFERENCE CREF

SYMBOL	VALUE	REFERENCES
FLUSH	015062	#187-7273
FSCMD0	050300	#380-12120 #382-12147
FSCMD1	050402	380-12121 #382-12169
FSCMD2	050512	380-12122 #384-12184
FSCMD3	050660	380-12123 #386-12208
FSCMD4	051134	380-12124 #388-12250
FSCMD5	051454	380-12125 #390-12302
FSCMD6	052372	380-12126 #391-12433
FSCMD7	052400	380-12127 #393-12439
FSCMD8	052672	380-12128 #395-12480
FSCMD9	053076	380-12129 #397-12510
FSINFL	002436	#139-5696 *189-7284
FSPAT	052154	390-12377 #390-12381
FSSTAC	002302	#139-5648 *382-12171 382-12179 *384-12186 384-12203 *390-12304 390-12422 *393-12441 393-12471
		*397-12512 397-12541
FS1	050166	#380-12112 380-12118 380-12144
FS7FLA	002442	#139-5698 238-8467
FULLRE	002536	#139-5713 *240-8502 *240-8530 *240-8541 363-11657
GBLENG	= 000076	246-8647 246-8650 #313-10304
GETCSR	041324	227-8220 231-8259 247-8722 248-8737 249-8749 250-8760 #329-10854
GETDAT	054256	#423-13267 454-14585 454-14597
GETDA1	054354	*423-13269 423-13277 #423-13280
GETDIS	060450	255-8823 257-8856 462-14787 #463-14791 465-14828
GOOD	002042	#139-5568 *160-6208 *160-6218 *160-6266 *282-9480 282-9488 *282-9503 *291-9783 *291-9795
		*317-10410 *319-10455 *321-10495 *321-10504 *322-10540 *322-10550 *323-10595 *323-10608 *324-10649
		324-10667 *324-10673 *326-10780 *326-10790 *421-13220 *421-13222 *421-13240 *423-13258 *451-14481
		*451-14487 *451-14492 *451-14499 *451-14502 *451-14509 *451-14514 *451-14519 *451-14524 *451-14529
		*453-14535 *453-14540 *453-14545 *453-14550 *453-14555 *453-14560 *453-14565 *453-14570 *453-14575
		*453-14580 456-14625 *458-14650 *458-14652 *460-14679 *460-14689 *460-14708 *460-14713 509-16386
		509-16392 509-16399 509-16402 509-16404 509-16406 509-16409
GOOD2	002044	#139-5569 *456-14615 *456-14619 509-16399
GOOD3	002046	#139-5570 *456-14616 *456-14620 509-16399
GTSWR	= 104407	#110-4597 155-6053
HEADER	002576	#141-5732 *147-5866 *187-7263 *187-7265 *191-7344 *191-7347 *209-7656 *209-7671 *211-7727
		*273-9239 *273-9242 *273-9252 *273-9255 *275-9328 *275-9331 *275-9342 *275-9345 *277-9416
		*277-9419 *282-9489 *282-9502 *291-9794 *305-10121 *305-10133 *321-10505 *322-10549 *323-10597
		*323-10611 *324-10659 *324-10669 *325-10723 *325-10745 *325-10750 *326-10781 *326-10791 *390-12353
		*393-12459 *397-12529 *458-14655 *458-14672 *460-14680 *460-14682 *460-14691 *460-14693 *460-14715
		*460-14717 *466-14910 468-14951 468-14963 *470-14999 476-15156 *476-15157 476-15172 *476-15173
		*476-15180
HIPAT	047500	205-7573 207-7611 #373-11982
HT	= 000011	#111-4692 434-13568
I	002446	#139-5700 *149-5923 *149-5929 149-5929 *166-6517 *166-6519 166-6567 166-6575 166-6579
		166-6606 166-6610 166-6656 166-6661 166-6663 166-6667 *184-7169 *184-7175 184-7186
		*184-7186 184-7200 *211-7726 211-7729 *211-7735 211-7735 *238-8442 238-8452 238-8458
		*238-8466 238-8466 *238-8471 238-8481 238-8487 *238-8495 238-8495
IBSAVE	061316	*465-14822 465-14847 *465-14855 *465-14858 466-14913 #466-14917
IIII	= 177777	157-6111 157-6111 #157-6111 159-6155 159-6155 159-6155 #159-6155 159-6176
		159-6176 159-6176 #159-6176 160-6259 160-6259 160-6259 #160-6259 160-6260 160-6260
		160-6260 #160-6260 282-9508 282-9508 282-9508 #282-9508 291-9806 291-9806 291-9806
		#291-9806 291-9808 291-9808 291-9808 #291-9808 316-10393 316-10393 316-10393 #316-10393
		316-10395 316-10395 316-10395 #316-10395 319-10460 319-10460 319-10460 #319-10460 321-10512



CZMSPA SYMBOL		CREATED BY CROSS REFERENCE VALUE	MACRO ON 26-APR-82 AT 09:54	PAGE 11 CREF						
REFERENCES										
321-10512 321-10512 #321-10512 322-10556 322-10556 322-10556 #322-10556 323-10632 323-10632										
323-10632 #323-10632 323-10634 323-10634 323-10634 #323-10634 324-10675 324-10675 324-10675										
#324-10675 325-10734 325-10734 325-10734 #325-10734 325-10736 325-10736 325-10736 #325-10736										
325-10737 325-10737 325-10737 #325-10737										
ILLCSR	014124		182-7081 182-7098 #182-7138							
IMPTES	013064		172-6921 173-6929 #174-6947							
INCBNK	047510		193-7371 195-7401 197-7433	199-7470	201-7504	203-7544	205-7587	207-7627	#373-11986	
INCPAT	047464		193-7367 197-7429 #373-11974							
INCRPT	047464		201-7507 203-7547 #373-11973							
INHBAN	002534		#139-5712 *363-11651 363-11659							
INHECC	002532		#139-5711 211-7680 343-11163	363-11649	*363-11650	363-11658	*363-11665	*363-11673	*365-11733	
INTFLA	002134		#139-5597 170-6863 172-6906	173-6923	182-7053	211-7693	355-11411	*371-11908	*371-11952	
INT64K	002136		#139-5598 170-6864 355-11414	361-11555	*371-11908	*371-11955				
INVALI	= 104511		#110-4671 211-7709 217-7918	219-7970	238-8446	388-12270	390-12331	393-12464	397-12534	
IOTVEC	= 000020		#111-4747 *147-5874 *147-5875							
JMPRL1	045740		363-11684 #363-11694							
KAMIKA	002004		#139-5551 253-8814 380-12109	*380-12111	*382-12154	*399-12563	*399-12568			
KAMITE	026774		233-8312 233-8320 234-8330	242-8546	244-8575	246-8639	#253-8813			
KDIAG	= 000010		#296-9893 296-9909 297-9930	297-9956						
KDPAR0	= 172360		#111-4887 222-8098 222-8100	222-8124	222-8125	222-8144	222-8145	234-8357	234-8360	
			236-8413 236-8416 236-8421							
KDPAR6	= 172374		#111-4893 *222-8101 *234-8361							
KDPAR7	= 172376		#111-4894 *222-8126 *222-8146	*236-8414						
KERNEL	= 104417		#110-4608 169-6777 169-6790	169-6811	169-6834	174-7024	213-7845	213-7853	242-8554	
			244-8610 246-8648 255-8849	257-8886	363-11682	365-11741	367-11800	367-11816	377-12064	
			386-12233 386-12244 388-12274	388-12292	388-12296	421-13217	421-13233	421-13244	423-13263	
			423-13275 426-13328 426-13334	429-13431	429-13435	430-13494	432-13528			
KERSTK	= 002000		#111-4679							
KFLAG	002524		#139-5709 273-9222 275-9322							
KIPAR0	= 172340		#111-4877 361-11515 361-11591	369-11824	369-11844	430-13483	432-13501			
KIPAR4	= 172350		#111-4881 *162-6281 *163-6342	*163-6357	*163-6407	*163-6411	*163-6415	*164-6429	*164-6438	
			164-6439 *166-6497 *166-6505	166-6591	166-6639	*166-6678	166-6679	166-6681	369-11859	
			*369-11860 369-11868 *369-11877							
KIPAR5	= 172352		#111-4882 *166-6498 *166-6506	166-6642	*166-6659	*166-6665	*166-6671	*166-6679	*321-10477	
			*369-11862 *369-11879							
KIPAR6	= 172354		#111-4883							
KIPDR0	= 172300		#111-4857 361-11593 369-11826	425-13317	428-13393	428-13417	432-13502			
KMAP	= 104422		#110-4612 155-6058							
KPFLAG	002112		#139-5588 *371-11905 *371-11915							
KSIZE	002372		#139-5679							
KSTACK	002560		#141-5725 144-5794 166-6507	189-7317	478-15210					
LAST	= 157776		#113-4975 211-7683 220-8049	232-3271	234-8340	242-8560	244-8586	244-8614	246-8654	
			289-9725 296-9908 297-9929	297-9966	315-10346	386-12226	388-12266			
LASTBA	002552		#141-5722 *144-5799 *145-5840	166-6490	166-6492	169-6752	170-6875	173-6936	182-7071	
			*182-7113 184-7152 191-7349	234-8351	238-8465	238-8472	240-8523	331-10880	331-10889	
			361-11572 361-11579 363-11630	363-11637	373-11989	386-12221	388-12263	390-12309	393-12467	
			395-12504 397-12537 400-12623							
LASTBL	002554		#141-5723 *166-6488 *166-6495	166-6681						
LASTER	002014		#139-5558 *189-7290							
LBS0	= 000404		157-6111 159-6155 159-6176	160-6260	282-9508	291-9808	316-10395	319-10460	321-10512	
			322-10556 323-10634 324-10675	325-10737						
LBS1	= 000403		160-6259 291-9806 316-10393	323-10632	325-10736					

CZMSPA		CREATED BY	MACRO	ON 26-APR-82 AT 09:54	PAGE 12						
SYMBOL	CROSS REFERENCE	VALUE	REFERENCES		CREF						
LBLS2	=	000402	325-10734								
LF	=	000012	#111-4693	434-13646							
LINK1	=	002516	#139-5706	*169-6745	*169-6749	169-6764	169-6803	*238-8436	*238-8440	238-8456	238-8485
			*240-8508	*240-8510	240-8520	*242-8560	*242-8568	*244-8584	*244-8597	*246-8654	*246-8662
			309-10205	311-10252	313-10295						
LINK2	=	002520	#139-5707	*169-6746	*169-6750	169-6789	*244-8587	*244-8598	244-8616	244-8623	
LKS	=	177546	#111-4689								
LOADBA	=	002426	#139-5692	*363-11621	365-11719	365-11724					
LOADCS	=	104425	#110-4617	282-9479	291-9776	315-10324	315-10339	316-10376	321-10487	321-10499	322-10533
			323-10581	323-10589	324-10653	325-10711	326-10769	326-10772	346-11228	346-11236	346-11244
			346-11254	352-11335	352-11345	354-11356	355-11430	384-12196	428-13411		
LOADHO	=	002562	#141-5726	189-7322	363-11622	365-11720	365-11728	377-12047			
LOOP	=	014742	#187-7242	189-7336							
LOWMAP	=	046344	363-11692	365-11751	#365-11761	428-13390					
LSIZE	=	002374	#139-5680	*184-7151	*184-7164	184-7188	184-7190	186-7237	509-16391		
LST\$\$	=	*****	157-6111	157-6111	157-6114	157-6114	159-6155	159-6155	159-6176	159-6176	160-6259
			160-6259	160-6260	160-6260	282-9508	282-9508	291-9806	291-9806	291-9808	291-9808
			316-10393	316-10393	316-10395	316-10395	319-10460	319-10460	321-10512	321-10512	322-10556
			322-10556	323-10632	323-10632	323-10634	323-10634	324-10675	324-10675	325-10734	325-10734
			325-10736	325-10736	325-10737	325-10737					
MAINT	=	177750	#111-4761	145-5819	154-6007						
MAPHO	=	170202	#113-4902	*363-11691	*365-11750	426-13361	*428-13389				
MAPKER	=	046706	227-8218	231-8257	247-8726	248-8736	250-8759	251-8768	251-8776	251-8784	251-8792
			251-8800	#369-11855							
MAPLO	=	170200	#113-4901	*363-11689	*365-11750	365-11763	426-13361	*428-13389			
MAPL1	=	170204	#113-4903	365-11764							
MAPPER	=	044534	155-6059	169-6761	173-6937	174-6948	211-7708	242-8551	244-8580	246-8644	255-8822
			257-8855	#361-11514	377-12059	386-12230	388-12269	390-12330	390-12363	390-12428	393-12475
			397-12545	430-13475	432-13516						
MASK	=	002314	#139-5653								
MBERR	=	014002	*182-7076	*182-7092	182-7096	*182-7100	*182-7105	#182-7115			
MEMDON	=	015010	187-7247	#187-7257							
MFPT	=	000007	#111-4696	145-5845							
MJPAT	=	020620	149-5914	219-7969	219-7969	219-7974	#219-7982				
MJTEST	=	020514	209-7669	#219-7962							
MKCNT	=	017666	*211-7706	*211-7751	211-7751	#211-7763					
MKCONT	=	017046	209-7664	#211-7675							
MKCSRT	=	020204	149-5900	#215-7867							
MKFLAG	=	002116	#139-5590	172-6904	182-7050	209-7659	240-8516	*371-11905	*371-11931	373-11969	
MKLOOP	=	017230	#211-7704	211-7757							
MKPAT	=	020434	149-5907	217-7917	217-7917	217-7922	#217-7934				
MKTEST	=	020274	191-7345	209-7667	#217-7907						
MMRO	=	177572	#111-4770	*341-11136	*341-11140	*363-11694	*363-11703	*365-11742	*365-11745	426-13349	*428-13402
			430-13493	*432-13512	509-16390	509-16398					
MMR1	=	177574	#111-4771	426-13349	*428-13402	430-13492	*432-13511	509-16390	509-16398		
MMR2	=	177576	#111-4772	426-13349	*428-13402	430-13491	*432-13510	509-16390	509-16398		
MMR3	=	172516	#111-4773	145-5826	*145-5827	*145-5828	145-5829	*189-7320	*363-11685	*363-11693	*365-11749
			*365-11752	*375-12007	*377-12052	426-13352	*428-13401	430-13490	*432-13509	509-16390	509-16398
MMTRAP	=	042570	147-5888	#339-11107							
MMVEC	=	000250	#111-4757	*147-5888	*147-5889						
MONFLG	=	002272	#139-5644	*144-5792	*154-5993	*154-6012	*154-6015	465-14878			
MPT	=	*****	187-7269	311-10216	339-11113	361-11601	377-12038	377-12053	380-12135	399-12585	406-12664

CZMSPA		CREATED BY	MACRO	ON 26-APR-82 AT 09:54	PAGE 13						
SYMBOL	CROSS REFERENCE	VALUE	REFERENCES		CREF						
			425-13288	428-13375	430-13462	434-13596	434-13639	434-13658	462-14746	463-14799	470-14987
			473-15107	479-15256	483-15435	485-15525	485-15541	493-15952			
MSEEDH	002572		#141-5730	*147-5867	147-5869						
MSEEDL	002574		#141-5731	*147-5868	147-5870						
MSG012	100014		*182-7144	#517-16614							
MSG034	076031		211-7744	#517-16556							
MSG034	076067		211-7746	#517-16557							
MSG000	100226		154-5996	#517-16621							
MSG001	074036		331-10877	#517-16503							
MSG002	074120		331-10878	#517-16504							
MSG003	074175		331-10879	#517-16505							
MSG004	074302		331-10887	331-10907	#517-16507						
MSG005	074410		333-10934	#517-16509							
MSG006	074422		494-15990	#517-16510							
MSG007	074457		335-10949	#517-16511							
MSG008	100174		159-6142	#517-16620							
MSG009	074471		159-6156	335-10980	#517-16512						
MSG010	074503		335-11032	#517-16513							
MSG011	074515		331-10893	331-10906	#517-16514						
MSG012	074603		331-10896	331-10908	#517-16515						
MSG013	074700		333-10937	#517-16516							
MSG014	074702		159-6153	159-6173	331-10905	333-10939	335-11042	465-14833	472-15035	472-15039	473-15105
			#517-16517								
MSG015	074704		*159-6151	159-6152	*159-6161	*159-6163	*159-6167	*159-6170	159-6172	*335-10956	*335-10958
			*335-10968	335-10969	*335-10990	*335-10992	*335-10995	*335-10998	335-10999	*335-11011	*335-11021
			335-11022	*335-11035	*335-11039	335-11040	#517-16518				
MSG016	074706		159-6143	335-11010	#517-16519						
MSG017	074720		331-10892	331-10895	#517-16520						
MSG018	074731		470-14994	473-15103	478-15200	478-15215	478-15230	478-15245	#517-16521		
MSG019	074734		473-15073	#517-16522							
MSG020	074740		380-12099	#517-16523							
MSG021	074761		380-12117	#517-16524							
MSG022	075547		406-12651	#517-16545							
MSG023	075571		384-12193	#517-16546							
MSG025	075605		382-12178	384-12202	#517-16547						
MSG026	075631		380-12112	#517-16548							
MSG027	075643		384-12189	#517-16549							
MSG028	075660		384-12198	#517-16550							
MSG029	075674		386-12211	#517-16551							
MSG030	075714		390-12306	400-12603	#517-16552						
MSG031	075733		386-12212	388-12254	#517-16553						
MSG032	075773		386-12237	388-12277	#517-16554						
MSG033	076012		386-12241	388-12281	#517-16555						
MSG035	076115		189-7295	#517-16558							
MSG036	076120		388-12253	#517-16559							
MSG037	076137		388-12284	#517-16560							
MSG038	076156		388-12293	#517-16561							
MSG039	076174		388-12286	#517-16562							
MSG040	076216		390-12305	#517-16563							
MSG041	076242		390-12315	#517-16564							
MSG042	076267		390-12319	#517-16565							
MSG043	076305		390-12324	#517-16566							

CZMSPA      CREATED BY MACRO ON 26-APR-82 AT 09:54      PAGE 14  
 SYMBOL CROSS REFERENCE      CREF  
 SYMBOL VALUE      REFERENCES

MSG046	076327	390-12337 393-12451 397-12522 #517-16567
MSG047	076362	#517-16568
MSG048	076401	380-12102 #517-16569
MSG049	076441	390-12334 #517-16570
MSG051	076474	429-13451 #517-16572
MSG055	076513	393-12442 #517-16573
MSG056	076534	393-12449 397-12520 #517-16574
MSG058	076567	478-15191 #517-16575
MSG061	076611	473-15082 #517-16576
MSG062	076620	487-15711 487-15770 #517-16577
MSG063	076640	487-15712 #517-16578
MSG064	076651	487-15713 487-15772 #517-16579
MSG065	076661	487-15771 #517-16580
MSG066	076673	466-14903 #517-16581
MSG067	076742	468-14955 #517-16582
MSG070	076751	184-7201 #517-16583
MSG073	077002	397-12513 #517-16584
MSG075	077020	363-11632 363-11669 #517-16585
MSG076	077052	395-12495 #517-16586
MSG077	077073	189-7293 #517-16587
MSG079	077107	395-12487 #517-16588
MSG085	077133	399-12553 #517-16589
MSG088	077160	478-15209 #517-16590
MSG089	077176	478-15225 #517-16591
MSG090	077220	478-15240 #517-16592
MSG091	077234	478-15234 478-15249 #517-16593
MSG092	077246	485-15535 #517-16594
MSG093	077262	485-15537 #517-16595
MSG095	077270	485-15539 #517-16596
MSG101	077300	399-12562 #517-16597
MSG102	077330	399-12567 #517-16598
MSG103	077357	382-12148 #517-16599
MSG104	077401	470-15004 #517-16600
MSG105	077403	400-12599 #517-16601
MSG106	077456	399-12573 #517-16602
MSG107	077474	399-12580 #517-16603
MSG110	077551	400-12614 #517-16604
MSG111	077615	400-12619 #517-16605
MSG112	077647	184-7191 #517-16606
MSG113	077664	184-7195 #517-16607
MSG114	077701	184-7199 #517-16608
MSG117	077716	154-6010 #517-16609
MSG119	077730	154-6029 #517-16610
MSG120	077737	154-6030 #517-16611
MSG121	077760	154-6024 #517-16612
MSG122	100000	182-7145 #517-16613
MSG125	100046	395-12485 #517-16616
MSG126	100070	163-6416 #517-16617
MSG127	100135	402-12633 #517-16618
MSG128	100154	404-12639 #517-16619
MSG129	100275	159-6177 #517-16622
MSIZE	002376	#139-5681 *184-7151 *184-7161 184-7192 184-7194 186-7237 509-16391

CZMSPA SYMBOL	CREATED BY CROSS REFERENCE VALUE	MACRO REFERENCES	ON 26-APR-82 AT 09:54	PAGE 15 CREF						
MTA030	024600	#240-8503	240-8535							
MTEST	016752	193-7365	195-7395	197-7427	199-7464	201-7501	203-7541	205-7584	207-7624	#209-7655
MTLA11	030412	273-9172	#273-9177	273-9272						
MTLB11	030424	#273-9180	273-9268							
MTLC11	030436	#273-9183	273-9262							
MTLD11	030532	#273-9200	273-9284							
MTPA03	027510	222-8038	222-8094	#261-8939						
MTPA04	027646	222-8120	222-8123	222-8141	#265-8998	265-9018				
MTPA21	034430	232-8282	232-8296	#292-9818						
MTPA24	035254	234-8355	#299-9982	301-10041						
MTPA25	035664	303-10052	303-10087	305-10095	#305-10106					
MTPA26	036014	236-8399	236-8400	236-8412	#307-10138	307-10181				
MTPB03	027550	222-8091	222-8097	#261-8962	263-8995					
MTPB04	027702	222-8124	*222-8139	*222-8141	222-8144	265-9002	#265-9011	265-9024		
MTPB21	034460	232-8284	232-8299	#292-9835						
MTPB24	035314	234-8349	234-8356	299-10023	#301-10026					
MTPB25	035706	303-10089	305-10097	#305-10113						
MTPB26	036030	236-8403	236-8404	236-8420	#307-10145					
MTPC03	027610	222-8098	261-8964	#263-8980	263-8994					
MTPC21	034514	232-8287	232-8303	#292-9852						
MTPC24	035330	234-8357	301-10032	#301-10034						
MTPC25	035746	303-10091	305-10100	305-10102	#305-10125					
MTPC26	036064	236-8413	307-10139	307-10148	#307-10163					
MTPD03	027626	222-8099	263-8986	#263-8988						
MTPD21	034550	232-8289	232-8306	#294-9870						
MTPD25	035612	303-10064	303-10066	303-10069	303-10071	#303-10087				
MTPD26	036104	*236-8399	*236-8403	236-8415	#307-10176					
MTPE25	035634	303-10075	303-10077	303-10080	303-10082	#305-10095				
MTP000	027400	220-8019	220-8022	#259-8902	259-8904	*328-10842	328-10843	*328-10845	328-10848	
MTP001	027424	220-8041	220-8044	#259-8913						
MTP002	027456	220-8067	220-8070	#259-8925						
MTP005	027722	222-8138	222-8139	222-8143	#265-9020					
MTP006	027756	224-8155	#267-9034							
MTP007	030156	224-8165	#269-9077							
MTP010	030256	224-8172	#271-9117							
MTP011	030364	226-8183	#273-9153							
MTP012	031162	226-8198	#275-9294							

CZMSPA CREATED BY MACRO ON 26-APR-82 AT 09:54

PAGE 16

SYMBOL CROSS REFERENCE

CREF

SYMBOL	VALUE	REFERENCES
MTP036	036526	247-8727 #316-10351
MTP037	036752	248-8738 #317-10404
MTP041	037024	249-8750 #319-10425
MTP042	037176	250-8761 #321-10466
MTP043	037432	251-8770 #322-10519
MTP044	037626	251-8778 #323-10560
MTP045	040142	251-8786 #324-10640
MTP046	040330	251-8794 #325-10680
MTP047	040670	251-8803 #326-10756
MTST3	012422	169-6745 169-6746 169-6748 169-6774 169-6808 #169-6846 169-6847
MT0000	020700	#220-8012 390-12381
MT0001	020760	217-7938 219-7987 #220-8025 390-12382
MT0002	021100	217-7939 219-7988 #220-8047 390-12383
MT0003	021240	219-7989 #222-8075 390-12384
MT0004	021472	217-7940 219-7990 #222-8111 390-12385
MT0005	021614	217-7941 219-7991 #222-8129 390-12386
MT0006	021750	215-7867 219-7984 #224-8151 390-12387
MT0007	022004	217-7937 219-7986 #224-8158 390-12388
MT0010	022046	215-7884 #224-8168 390-12389
MT0011	022102	215-7890 #226-8177 390-12390
MT0012	022160	215-7891 #226-8186 390-12391
MT0013	022264	215-7892 #226-8201 390-12392
MT0014	022350	215-7873 #227-8212 390-12393
MT0015	022440	215-7893 #229-8226 390-12394
MT0016	022516	215-7894 #229-8235 390-12395
MT0017	022574	217-7936 219-7985 #229-8244 390-12396
MT0020	022616	215-7876 #231-8251 390-12397
MT0021	022706	217-7942 219-7992 #232-8264 390-12398
MT0022	023160	217-7943 219-7994 #233-8311 390-12399 397-12535
MT0023	023212	219-7995 #233-8319 390-12400
MT0024	023256	217-7945 219-7997 #234-8329 390-12401
MT0025	023522	215-7889 #234-8371 390-12402
MT0026	023600	217-7944 219-7996 #236-8382 390-12403
MT0027	024102	187-7264 #238-8429 390-12404
MT0030	024566	187-7274 #240-8500 375-12003 377-12044 390-12405
MT0031	025070	217-7946 219-7998 #242-8545 390-12406 393-12465
MT0032	025260	217-7947 219-7999 #244-8574 390-12407
MT0033	025612	217-7948 219-8000 #246-8638 390-12408
MT0034	026000	217-7917 217-7935 217-7949 219-7969 219-7983 219-8001 #246-8668 390-12409
MT0035	026152	219-7993 #246-8698 390-12410
MT0036	026264	215-7875 #247-8719 390-12411
MT0037	026336	215-7877 #248-8731 390-12412
MT0040	026404	#248-8743 390-12413
MT0041	026406	215-7878 #249-8746 390-12414
MT0042	026450	215-7879 #250-8755 390-12415
MT0043	026514	215-7880 #251-8765 390-12416
MT0044	026554	215-7872 #251-8773 390-12417
MT0045	026614	215-7874 #251-8781 390-12418
MT0046	026654	215-7881 #251-8789 390-12419
MT0047	026714	215-7882 #251-8797 390-12420
MT0999	026760	149-5926 215-7898 215-7899 215-7900 215-7901 215-7902 217-7917 217-7951 217-7952 217-7953 217-7954 217-7955 217-7956 217-7957 217-7958 217-7959 219-7969 219-8003

CZMSPA SYMBOL SYMBOL	CREATED BY CROSS REFERENCE VALUE	MACRO REFERENCES	ON 26-APR-82 AT 09:54	PAGE 17 CREF						
MT1	017026	219-8004	219-8005	219-8006	219-8007	#253-8808				
MT2	017032	209-7660	#209-7669							
MUT	002106	209-7668	#209-7670							
		#139-5586	*187-7263	*187-7266	*191-7344	*191-7346	*209-7657	*209-7670	*390-12353	*393-12459
		*397-12529	470-14997	476-15156	*476-15158	476-15172	*476-15174	476-15180		
NC	056430	434-13624	434-13628	434-13634	#434-13638					
NEMCNT	002066	#139-5578	*167-6703	167-6716	*169-6760	169-6784	169-6796	169-6818	169-6842	*339-11092
		339-11093	*339-11097	509-16389						
NEWBAN	002304	#139-5649	240-8533	*363-11677	363-11686	363-11696	*365-11737	369-11832	369-11845	
NEWKER	046640	363-11695	365-11743	#369-11842						
NEWLOA	046742	363-11624	365-11722	#369-11865						
NOCH	063560	483-15441	#483-15445							
NOERRO	002424	#139-5691	*382-12174	*384-12190	*384-12199	465-14823	465-14860	466-14883	*466-14912	468-14949
		468-14961	470-15001	*476-15169	*476-15175					
NOFSMO	002422	#139-5690	*238-8441	*238-8468	*238-8496	*240-8513	*240-8530	*240-8541	380-12101	
NONEM	002076	#139-5582	*167-6704	*169-6741	*169-6827	*169-6836	339-11090			
NONEXI	042512	167-6705	169-6742	#339-11090						
NOOJ	041464	331-10881	#331-10884							
NOPAR	002074	#139-5581	*144-5804	*144-5810	*157-6086	*164-6428	*167-6702	*169-6740	*169-6782	*211-7725
		*217-7913	*217-7928	*219-7963	*219-7975	*226-8208	*240-8504	*242-8550	*244-8579	*246-8643
		*282-9475	*303-10049	*303-10084	*315-10320	*315-10348	*325-10696	337-11063	337-11067	337-11074
		386-12209	*386-12210	*386-12245	388-12251	*388-12252	*388-12297	*390-12373	*390-12378	393-12440
		*393-12474	397-12511	*397-12544						
NORES	003664	144-5790	#144-5792							
NOSCOP	002434	#139-5695	*232-8265	*232-8308	*234-8332	*234-8369	*240-8513	*240-8530	*240-8541	*242-8548
		*242-8570	*244-8577	*244-8632	*246-8641	*246-8665	462-14758	466-14888		
NOSUPE	002452	#139-5702	*145-5834	*145-5853	155-6034	220-8029	220-8053	242-8561	244-8588	246-8655
		257-8878	355-11376	361-11518	361-11535	361-11560	361-11581	425-13309	426-13330	426-13350
		428-13399	429-13426	430-13488	432-13507	478-15219				
NOTAB	002366	#139-5677	335-10970	335-11000	335-11023	*472-15034	*472-15037	*472-15048	*472-15050	*473-15072
		*473-15075								
NOTRCE	060220	462-14737	#462-14739							
NO22BI	002450	#139-5701	*145-5837							



CZMSFA SYMBOL	CREATED BY CROSS REFERENCE VALUE	MACRO	ON 26-APP-82 AT 09:54	PAGE 18 CREF						
		REFERENCES								
		*211-7728	*240-8501	240-8524	*240-8525	*247-8723	273-9185	273-9257	273-9269	*273-927
		*273-9273	277-9388	279-9423	279-9435	*279-9437	*279-9440	283-9526	283-9571	283-958
		*283-9585	*283-9587	285-9627	287-9673	287-9685	*287-9687	*287-9689	*291-9758	*291-9761
		291-9764	291-9800	291-9808	*316-10357	*316-10360	316-10363	316-10387	316-10395	*319-10435
		*319-10439	319-10460	*321-10478	*321-10485	321-10512	*322-10527	*322-10531	322-10556	*323-10568
		*323-10573	323-10615	323-10634	*325-10691	*325-10694	325-10698	325-10728	325-10736	*390-12358
PASSNO	002264	#139-5641	*282-9494	*282-9497	282-9508	*323-10576	*323-10578	323-10632	*324-10648	*324-10652
		324-10675	*325-10696	*325-10689	325-10713	325-10737	325-10743			
PATERR	002072	#139-5580	*169-6758	169-6792	169-6814	*169-6852				
PATPLU	004606	149-5902	149-5905	149-5909	149-5912	149-5916	149-5919	#149-5922		
PATTER	002110	#139-5587	*191-7340	*193-7363	*195-7397	*197-7425	*199-7466	*201-7490	*203-7528	*205-7590
		*207-7630	217-7915	219-7967	*373-11977	373-11978	*373-11983	390-12303	*390-12321	390-12322
		390-12323	390-12375	*390-12427	393-12440	*393-12474	397-12511	*397-12544		
PCBUMP	002322	#139-5656	*217-7914	*219-7964	*222-8078	*222-8113	*222-8131	*224-8153	*224-8170	*236-8384
		337-11071	390-12303	*390-12374	*390-12426					
PCONF1	041352	184-7203	#331-10868	391-12434						
PCONFS	041652	*331-10870	331-10911	#331-10917						
PCONF1	041562	331-10883	#331-10902							
PCONF2	041620	331-10871	331-10890	331-10900	#331-10911					
PDP110	042602	147-5884	#339-11110							
PD1	054476	425-13310	#425-13317							
PERA05	056644	#451-14503	451-14510							
PERBNK	057476	211-7747	315-10330	315-10345	421-13247	454-14584	454-14596	456-14611	#458-14632	460-14705
		465-14869								
PERECC	057556	#458-14648	458-14662							
PERRAB	057314	451-14525	451-14530	#454-14595						
PERRAW	057242	421-13224	423-13265	451-14482	451-14488	451-14494	451-14500	451-14515	451-14520	453-14536
		453-14541	453-14546	453-14551	453-14556	453-14561	453-14571	453-14576	453-14581	#454-14583
PERRA3	054044	#421-13226	456-14617	456-14621						
PERRA7	057366	453-14566	#456-14608							
PERR01	= 104427	#110-4620	259-8921	311-10256						
PERR02	= 104430	#110-4621	259-8909	259-8934						



CZMSPA	CREATED BY	MACRO	ON 26-APR-82 AT 09:54	PAGE 19
SYMBOL	CROSS REFERENCE	VALUE	REFERENCES	REF
PERR27	= 104453	#110-4640		
PERR30	= 104454	#110-4641	309-10197	
PERR31	= 104455	#110-4642	273-9214	273-9220 283-9561 283-9567 285-9659 285-9666 319-10456
PERR32	= 104456	#110-4643	273-9235	273-9248
PERR33	= 104457	#110-4644	275-9336	
PERR34	= 104460	#110-4645	273-9241	273-9254 275-9344
PERR35	= 104461	#110-4646	313-10273	313-10280
PERR36	= 104462	#110-4647	355-11421	
PERR37	= 104463	#110-4648		
PERR40	= 104464	#110-4649		
PERR41	= 104465	#110-4650		
PERR42	= 104466	#110-4651		
PERR43	= 104467	#110-4652		
PERXOR	057452	454-14587	454-14599	456-14610 #456-14623 458-14654 460-14690
PFECDF	061740	470-15007	#470-15012	
PFECDH	061700	470-15007	#470-15009	
PFECDT	061730	470-15007	#470-15011	
PFCEM	061644	470-15007	#470-15008	
PFECWS	061634	468-14940	#470-15007	
PFLAG	002120	#139-5591	172-6918	363-11619 *371-11907 *371-11937
PGMCSR	002526	#139-5710	*162-6280	*162-6326 162-6331 *163-6374 *163-6404 *163-6405 *163-6417 246-8706
		343-11164	343-11166	343-11170 344-11190 344-11192 344-11196 363-11678 *363-11678 *363-11704
		*363-11711	365-11744	*365-11744
PHEBE	014004	#182-7116	*182-7118	*182-7123 182-7126 182-7128 182-7130 182-7132 *182-7134 *182-7136
		273-9171	273-9228	
PHYADD	002036	#139-5567	*473-15088	*473-15089 *473-15090 *473-15091 *473-15093 473-15094
PMEMFL	002140	#139-5599	226-8181	226-8190 226-8205 227-8216 229-8230 229-8236 231-8256 234-8375
		247-8720	248-8732	249-8747 250-8756 251-8766 251-8774 251-8782 251-8790 251-8798
		*371-11905	*371-11934	
PROTYP	003752	#145-5815	*145-5850	145-5851 145-5860 151-5956 164-6446 167-6721 169-6743 169-6772
		169-6781	169-6806	220-8017 220-8039 220-8065 222-8085 222-8118 222-8136 232-8274
		232-8276	234-8341	234-8343 236-8391 236-8393 238-8434 238-8453 238-8459 238-8482
		238-8488	240-8505	244-8606 244-8626 246-8676 246-8682 246-8688 328-10840 344-11199
		363-11683	365-11747	382-12176 382-12180 384-12192 384-12204 386-12246 388-12298 426-13359
		428-13387	462-14771	466-14895 476-15177
PSIZE	002400	#139-5682	*184-7151	*184-7159 184-7196 184-7198 186-7237
PSW	= 177776	#111-4684	*155-6038	*155-6039 *155-6045 *169-6771 *169-6788 *169-6805 *169-6831 *174-6951
		*213-7794	*242-8552	*244-8600 *246-8646 *255-8844 *257-8877 *331-10874 *333-10933 *335-10951
		*344-11215	*344-11217	*363-11680 *365-11739 *367-11798 *367-11814 *377-12060 *386-12231 *388-12271
		*388-12289	*390-12349	*393-12455 *397-12526 *421-13215 *421-13231 *421-13241 *423-13261 *423-13273
		*426-13326	*426-13332	*429-13429 *429-13433 *430-13476 *432-13517 *478-15221 *478-15222 *478-15237
PTABLE	036732	291-9772	316-10372	#316-10402 325-10707
PWRVEC	= 000024	#111-4748	*147-5880	*147-5881 *153-5975 *425-13300 *425-13301 *426-13367 *428-13378 *429-13450
		430-13477	430-13478	*430-13480 *430-13481 *432-13518 *432-13519
QUICK	002432	#139-5694	*153-5974	377-12043
QVFLAG	002342	#139-5664	*153-5974	*153-5979 189-7294 *189-7296 246-8713 273-9264 275-9361 279-9430
		283-9578	287-9680	
RANODD	036044	*236-8402	*236-8419	#307-10153 *307-10157
RDCHR	= 104411	#110-4600	486-15601	

CZMSPA SYMBOL	CREATED BY CROSS REFERENCE VALUE	MACRO	ON 26-APR-82 AT 09:54	PAGE 20 CREF							
READCS	= 104426	REFERENCES 400-12604 #110-4618 315-10333 324-10664 384-12188	166-6534 319-10447 325-10720 384-12197	273-9223 321-10491 326-10776 421-13234	282-9486 321-10500 326-10786 426-13342	291-9780 322-10537 348-11266 478-15198	291-9788 322-10546 348-11285	303-10061 323-10590 350-11301	305-10118 323-10603 350-11320	305-10130 324-10657 382-12173	
READON REALPA	002404 002274	#139-5644 #139-5645 *224-8169 *232-8266 *244-8578 *251-8775 458-14666	*220-8013 *226-8182 *233-8314 *246-8642 *251-8783 458-14669	*220-8026 *226-8191 *233-8322 *246-8669 *251-8791 463-14793	*220-8048 *226-8206 *234-8333 *246-8699 *251-8799 472-15057	*222-8077 *227-8217 *234-8376 *247-8721 *253-8809 485-15540	*222-8112 *229-8231 *236-8383 *248-8733 363-11660	*222-8130 *229-8240 *238-8432 *249-8748 363-11667	*224-8152 *229-8245 *240-8503 *250-8757 458-14658	*224-8159 *231-8255 *242-8549 *251-8767 458-14663	
REFRES REFSUB REGCOP RELENT RELOC A	035154 035224 041064 045612 045172	296-9923 297-9963 220-8016 363-11661 193-7377 #363-11611 #363-11677	#297-9960 297-9967 220-8038 463-11662 195-7407	#297-9971 #327-10802 #363-11674 197-7444	199-7481	201-7519	203-7564	205-7602	207-7647	240-8527	
RELOC1 RESREG	= 045626 = 104416	#110-4606 367-11810 *137-5538 #111-4713 382-12158 382-12172 384-12187 #139-5593 *363-11712 #139-5592 207-7622 390-12333 157-6100 #139-5604 #139-5643 #139-5642 #110-4605 367-11782 #139-5636 *275-9301 277-9393 *279-9453 283-9579 *285-9622 287-9695 213-7799 213-7842 291-9790 211-7722 #111-4683 #111-4847 #111-4852 #111-4853	222-8090 380-12103 *137-5540 *147-5884 382-12163 #382-12178 #384-12202 193-7374 *365-11746 191-7343 211-7712 393-12463 #160-6191 *390-12338 *144-5793 *369-11859 222-8087 380-12098 *273-9180 275-9306 277-9395 279-9454 *283-9581 285-9631 *287-9702 213-7801 #213-7851 #291-9815 #213-7766 157-6063 222-8099 *222-8103 *236-8417	222-8104 382-12166 #141-5738 *147-5885 #382-12166 195-7404 371-11938 193-7361 238-8445 397-12533 390-12430 465-14879 369-11877 222-8095 468-14930 *273-9181 275-9308 277-9399 279-9456 *283-9581 285-9633 287-9703 213-7806 167-6694 222-8101 *234-8360	234-8365 470-15000 151-5937 197-7441 375-12002 195-7391 238-8474 197-7478 377-12042 197-7423 240-8517 369-11879 234-8348 491-15851 273-9198 275-9356 279-9431 *283-9521 283-9531 *283-9590 285-9637 287-9705 213-7817 169-6731 234-8356 172-6879 234-8358 187-7242 234-8359 187-7259 236-8415	236-8405 236-8422 236-8422 380-12101 199-7460 201-7499 *371-11907 203-7561 425-13315 201-7499 *371-11937 205-7599 463-14795 203-7539 *371-11940 207-7644 485-5534 205-7582 *371-11948 207-7644 485-5534 205-7582 *371-11948	205-7602 207-7647 240-8527 328-10846 328-10851 367-11781 328-10837 367-11779	328-10846 328-10851 367-11781 328-10837 367-11779	328-10837 367-11779	328-10837 367-11779	328-10837 367-11779
SBEMSK	002250	213-7806 213-7808 213-7817 213-7823 213-7825 213-7831 213-7833	213-7806 213-7808 213-7817 213-7823 213-7825 213-7831 213-7833	213-7806 213-7808 213-7817 213-7823 213-7825 213-7831 213-7833	213-7806 213-7808 213-7817 213-7823 213-7825 213-7831 213-7833	213-7806 213-7808 213-7817 213-7823 213-7825 213-7831 213-7833	213-7806 213-7808 213-7817 213-7823 213-7825 213-7831 213-7833	213-7806 213-7808 213-7817 213-7823 213-7825 213-7831 213-7833	213-7806 213-7808 213-7817 213-7823 213-7825 213-7831 213-7833	213-7806 213-7808 213-7817 213-7823 213-7825 213-7831 213-7833	
SBESYN SBETES SCOPE SDPAR0 SDPAR5 SDPAR6	034410										

CZMSPA		CREATED BY	MACRO	ON 26-APR-82 AT 09:54	PAGE 21										
SYMBOL	CROSS REFERENCE	VALUE	REFERENCES												
SDPAR7	= 172276	#111-4854	*222-8102												
SEEDHI	002566	#141-5728	*147-5869	236-8386	*236-8425	489-15824	489-15831	*489-15836							
SEEDLO	002570	#141-5729	*147-5870	236-8385	*236-8424	489-15823	489-15829	*489-15835							
SELONL	002000	#139-5549	187-7262	211-7679	371-11947	*400-12615	*400-12620								
SETPAT	047500	195-7393	199-7462	#373-11981											
SHADL1	012452	#170-6861	170-6876												
SHUTUP	047664	189-7305	377-12023	#377-12031											
SIPARO	= 172240	#111-4837	361-11516												
SIPAR3	= 172246	#111-4840	273-9173	*273-9174	355-11378	361-11534									
SIPAR5	= 172252	#111-4842	220-8031	220-8055	242-8563	244-8590	246-8657	273-9174	*273-9175	361-11564					
		361-11585													
SIPAR6	= 172254	#111-4843	220-8031	220-8055	242-8563	244-8590	246-8657	*361-11585							
SIPDRO	= 172200	#111-4817	361-11517	425-13319	425-13419										
SIZE	= 040000	#113-4976	167-6707	169-6767	169-6800	220-8015	220-8028	220-8050	222-8117	222-8135					
		236-8390	238-8450	238-8479	240-8518	244-8583	244-8620	246-8671	328-10839	363-11681					
		355-11740	367-11799	367-11815											
SKIPKA	002006	#139-5552	382-12150	*382-12152	*399-12563	*399-12569									
SKIPMK	002336	#139-5662	172-6905	209-7663	*371-11908	*371-11958									
SKJ	060246	462-14741	462-14744	#462-14754											
SKPERR	002064	#139-5577	*213-7812	*213-7837	355-11419	*355-11423									
SKUB	045602	363-11668	#363-11671												
SKUJ	014006	182-7114	#182-7117												
SOBK	002556	#141-5724	242-8555												
SOBLEN	= 000056	242-8553	242-8558	#309-10212											
SOFTPA	002604	#141-5735	246-8672												
SOURCE	002306	#139-5650	*273-9191	*275-9304	*277-9397	*283-9535	*285-9635	*303-10047	357-11445						
SPLTCS	002236	#139-5633	*172-6908	*173-6930	*211-7685	*211-7695	211-7717	*211-7750	*211-7753	*211-7760					
		361-11542	361-11547	*390-12357	*390-12362										
		#111-4701	*155-6042	*255-8845	*257-8882	426-13333	*429-13430	478-15223							
SSP	=X000006	#498-16094													
ST	= 177776	#111-4678	111-4679	137-5531	137-5542	141-5725	153-5978	189-7301							
STACK	= 002000	137-5539	137-5541	#144-5788	377-12028										
START	003654	137-5535	#137-5538												
START1	000300	137-5536	#137-5540												
START2	000310	#137-5535	517-16636												
START3	000200	238-8437	#238-8442												
STAR27	024162	#139-5687	*174-7027	462-14754	*462-14755	*486-15610									
STOPOK	002414	#139-5675	*296-9902	296-9906	296-9925	*297-9956	297-9956								
STRIPE	002362	149-5920	#151-5933												
SUBAAA	004644	#153-5963													
SUBAAB	004774	169-6783	#170-6856												
SUBAAI	012446	182-7137	#184-7150												
SUBAAP	014166	173-6945	#174-7027												
SUBAAR	013372	166-6508	#167-6692												
SUBAAS	011400	#139-5659	*211-7711	*211-7737	211-7743	*395-12489	395-12494	*395-12496							
SUCCE	002330	#139-5638	*220-8019	*220-8041	*220-8067	*222-8088	*222-8091	*222-8120	*222-8138	*224-8155					
SUPDOA	002260	*224-8165	*224-8172	*226-8183	*226-8198	*226-8207	*229-8232	*229-8241	*229-8246	*232-8282					
		*232-8284	*232-8287	*232-8289	*233-8315	*233-8323	*234-8349	*234-8358	*234-8377	*236-8400					
		*236-8404	*238-8439	*238-8455	*238-8461	*238-8484	*238-8490	*240-8511	*242-8559	*244-8615					
		*244-8628	*246-8651	*246-8679	*246-8684	*246-8690	*246-8711	*249-8750	257-8884	*328-10843					
		220-8023	220-8045	220-8071	222-8096	232-8297	236-8418	236-8423	238-8440	240-8508					
SUPD01	027030	246-8693	#255-8821	328-10850											

CZMSPA		CREATED BY	MACRO	ON 26-APR-82 AT 09:54	PAGE 22					
SYMBOL	CROSS REFERENCE	VALUE	REFERENCES		CREF					
SUPDO2		027044	222-8105	222-8127	222-8147	232-8300	232-8304	232-8307	244-8631	#255-8823
SUPDO3		027206	220-8020	220-8042	220-8068	222-8089	224-8156	224-8166	224-8173	226-8184
			226-8209	229-8233	229-8242	229-8247	232-8283	233-8316	233-8325	234-8378
			236-8406	238-8436	238-8462	238-8491	240-8510	246-8685	246-8691	246-8712
			#257-8855	328-10844						
SUPDO4		027222	222-8092	222-8121	222-8140	232-8285	232-8288	232-8290	234-8362	234-8368
			244-8619	244-8629	246-8664	246-8716	#257-8856			242-8569
SUPDR0		002156	#139-5607	*255-8825	255-8833	*257-8858	257-8866			
SUPDR1		002160	#139-5608	255-8826	257-8859					
SUPDR2		002162	#139-5609							
SUPDR3		002164	#139-5610							
SUPDR4		002166	#139-5611							
SUPDR5		002170	#139-5612							
SUPDR6		002172	#139-5613	255-8836	257-8869					
SUPLIM		056506	#434-13656	517-16631	517-16632					
SUPSTK	=	000740	#111-4680	155-6041	255-8845	257-8882	478-15226	478-15232		
SWAPAT		002620	#141-5743	*147-5872						
SWR		002622	#141-5746	*151-5950	151-5952	*151-5957	*153-5976	155-6052	167-6724	173-6939
			187-7244	189-7294	217-7908	217-7923	273-9263	275-9360	279-9429	283-9577
			331-10883	363-11612	363-11631	363-11666	390-12366	393-12444	397-12515	421-13250
			428-13385	462-14754	462-14761	462-14779	465-14830	465-14870	465-14873	466-14884
			470-15001	483-15465	483-15473	483-15495	486-15608			466-14898
SWREG	=	000176	#113-4968	151-5957	155-6052	483-15465				
SW0	=	000001	#111-4721	167-6724	173-6939	217-7908	217-7923	390-12366	393-12444	397-12515
SW1	=	000002	#111-4720							421-13250
SW10	=	002000	#111-4711	465-14830						
SW11	=	004000	#111-4710	189-7294	273-9263	275-9360	279-9429	283-9577	287-9679	
SW12	=	010000	#111-4709	363-11612						
SW13	=	020000	#111-4708	363-11631	363-11666	465-14870				
SW14	=	040000	#111-4707	462-14761						
SW15	=	100000	#111-4706							
SW2	=	000004	#111-4719							
SW3	=	000010	#111-4718							
SW4	=	000020	#111-4717	331-10883						
SW5	=	000040	#111-4716	465-14873						
SW6	=	000100	#111-4715	184-7202						
SW7	=	000200	#111-4714	470-15001						
SW8	=	000400	#111-4713	462-14754						
SW9	=	001000	#111-4712	462-14779	466-14888					
SYSSIZ		003754	145-5814	#145-5816						
TAG2\$		012042	169-6753	#169-6780						
TAG3\$		012076	169-6779	#169-6784						
TAG4\$		027124	255-8834	#255-8836						
TAG70\$		061744	470-14977	#472-15019						
TAG71\$		061754	470-14978	#472-15025						
TAG72\$		061764	470-14979	#472-15031						
TAG73\$		062034	470-14980	#472-15045						
TAG74\$		062074	470-14981	#472-15057						
TAG75\$		062106	470-14982	#472-15063						
TAG76\$		062120	470-14983	#473-15069						
TAG77\$		062164	470-14984	#473-15082						
TAG78\$		062172	470-14985	#473-15088						

CZMSPA		CREATED BY	MACRO	ON 26-APR-82 AT 09:54	PAGE 23					
SYMBOL	CROSS REFERENCE	VALUE	REFERENCES	CREF						
TAG79\$		062252	470-14986 #473-15103							
TAG9\$		011670	169-6747 #169-6751	169-6786	169-6823	169-6845				
TBG4\$		027302	257-8867 #257-8869							
TCFIG1		041726	#335-10951 335-10972	472-15036						
TCFIG2		042066	#335-10981 335-11002	473-15074						
TCFIG3		042222	#335-11011 335-11025	472-15049						
TCONF1		041654	331-10888 331-10899	331-10909	#333-10933					
TEMP		002430	#139-5693 *166-6538	166-6594	166-6596	*186-7209	186-7226	*186-7227	*395-12482	*395-12483
TESTAD		U02406	395-12484							
			#139-5685 162-6282	*162-6283	*162-6284	166-6482	*166-6512	*166-6513	*166-6658	*166-6669
			*166-6670 *211-7715	211-7716	*211-7716	*211-7719	*211-7721	*211-7740	211-7740	213-7792
			213-7793 *219-7965	*219-7966	224-8154	224-8171	273-9200	273-9201	273-9287	273-9288
			275-9310 275-9351	275-9368	277-9379	279-9460	283-9545	283-9552	283-9556	283-9607
			285-9645 285-9654	287-9709	303-10050	390-12303	*390-12359	*390-12360	*390-12364	*390-12426
			458-14649							
TESTMO		002546	#141-5720 *145-5833	*145-5854	169-6771	169-6788	169-6805	169-6831	174-6951	213-7794
			242-8552 244-8600	246-8646	257-8877	344-11215	377-12060	386-12231	388-12271	388-12289
			421-13215 421-13231	421-13241	423-13261	423-13273				
TIME		002334	#139-5661							
TIMEOU		042556	145-5860 147-5886	151-5956	164-6446	167-6721	169-6781	#339-11103	382-12176	382-12180
TKVEC	=	000060	384-12192 384-12204	386-12246	388-12298					
			#111-4751 331-10869	331-10869	*331-10871	*331-10872	*331-10914	*331-10914	390-12303	390-12303
			*390-12346 *390-12347	*390-12424	*390-12424	393-12440	393-12440	*393-12452	*393-12453	*393-12474
			*393-12474 397-12511	397-12511	*397-12523	*397-12524	*397-12544	*397-12544		
TMFLAG		002132	#139-5596 *201-7510	*203-7550	*205-7593	*207-7633	373-11967			
TOOMAN		002402	#139-5683 *458-14644	465-14873	*466-14912					
TOTCSR		002222	#139-5626 *157-6116	162-6287	163-6345	163-6377	166-6483	348-11260	350-11295	354-11351
			406-12645 426-13337	428-13404	478-15195					
TRACE		006204	#159-6179 *164-6427	*164-6433	164-6435	*164-6441	*164-6445	*402-12634	*404-12640	462-14736
TRAPVE	=	000034	#111-4750 *147-5878	*147-5879						
TSTBAN		012310	169-6763 #169-6826							
TSTDAT		002244	#139-5635 *273-9183	*273-9184	*273-9187	*273-9188	273-9189	273-9190	273-9191	*273-9197
			*273-9199 273-9203	273-9205	273-9211	273-9217	*273-9276	*273-9277	*275-9302	*275-9303
			275-9304 *275-9307	*275-9309	275-9313	275-9316	*277-9385	*277-9387	*277-9390	*277-9391
			277-9397 *277-9400	*277-9402	*277-9404	*277-9406	277-9408	277-9410	*279-9441	*279-9442
			*283-9524 *283-9525	*283-9526	*283-9529	283-9535	*283-9538	*283-9540	*283-9542	*283-9544
			283-9547 283-9550	283-9558	283-9564	*283-9588	*283-9589	*285-9625	*285-9626	*285-9629
			*285-9630 285-9635	*285-9638	*285-9640	*285-9642	*285-9644	285-9647	285-9649	285-9656
			285-9663 *287-9690	*287-9691	*303-10045	*303-10046	303-10047	*303-10063	*303-10065	*303-10067
			*303-10068 *303-10070	*303-10073	*303-10074	*303-10076	*303-10078	*303-10079	*303-10081	305-10107
			305-10109 *316-10382	*316-10383	453-14545	453-14550	458-14650	458-14652	509-16397	509-16397
			344-11201 344-11203	#344-11214						
TSTRD1		043230	#110-4670	213-7810	213-7835	273-9237	273-9250	275-9326	275-9338	
TSTREA	=	104510	#157-6063							
TST1		00522	#167-6694							
TST2		011404	#169-6731							
TST3		011570	#172-6879							
TST4		012556	#187-7242							
TST5		014742	#187-7259							
TST6		015014	#110-4594	184-7190	184-7194	184-7198	184-7200	189-7298	395-12484	395-12486
TVPDS	=	104405	472-15025							395-12501
TYPEIT	=	104401	#110-4590	154-5996	154-6010	154-6024	154-6029	154-6030	159-6142	159-6143
									159-6152	

PAGE 24  
CREF

CZMSPA SYMBOL SYMBOL	CREATED BY CROSS REFERENCE VALUE
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40
41	41
42	42
43	43
44	44
45	45
46	46
47	47
48	48
49	49
50	50
51	51
52	52
53	53
54	54
55	55
56	56
57	57
58	58
59	59
60	60
61	61
62	62
63	63
64	64
65	65
66	66
67	67
68	68
69	69
70	70
71	71
72	72
73	73
74	74
75	75
76	76
77	77
78	78
79	79
80	80
81	81
82	82
83	83
84	84
85	85
86	86
87	87
88	88
89	89
90	90
91	91
92	92
93	93
94	94
95	95
96	96
97	97
98	98
99	99
100	100

## REFERENCES

SYMBOL	VALUE
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40
41	41
42	42
43	43
44	44
45	45
46	46
47	47
48	48
49	49
50	50
51	51
52	52
53	53
54	54
55	55
56	56
57	57
58	58
59	59
60	60
61	61
62	62
63	63
64	64
65	65
66	66
67	67
68	68
69	69
70	70
71	71
72	72
73	73
74	74
75	75
76	76
77	77
78	78
79	79
80	80
81	81
82	82
83	83
84	84
85	85
86	86
87	87
88	88
89	89
90	90
91	91
92	92
93	93
94	94
95	95
96	96
97	97
98	98
99	99
100	100

ТҮҮС = 104402

**TYPOS = 104403**

**TYP50 = 000000**

**TYP51 = 000000**

TYP2 = 000000

T12A	033234
T12B	033276

UDPARO = 177660

UDPAR7 = 177676  
UDPAR0 = 133610

UIPAR0 = 177640  
UIPAR1 = 177642

UIPAR2 = 177644

UIPAR3	=	177646
UIPAR4	=	177650

```
UIPAR4 = 177650
UIPAR5 = 177652
```

133/51

```
UIPAR6      = 177654
UIPDR0      = 177600
```

UNITOP 002412

UNMAP 046774

UNRELO 046056

CHARLES	010010
MARILYN	000013

UPPFLG 002263  
USERMA 046556

```

USERMA      = 040550
USESTK      = 000700

```

USP =%000006  
HARN1 011754

WARN1	011736
WARN2	027522

WARN3 027536

1. *Journal of the American Medical Association*, 2000; 283: 2689-2693.

REFERENCES	159-6150	159-6172	159-6173	159-6177	163-6416	182-7145	184-7187	184-7191
159-6153	159-6150	159-6172	159-6173	159-6177	163-6416	182-7145	184-7187	184-7191
184-7195	184-7199	184-7201	189-7293	189-7295	211-7744	211-7746	331-10877	331-10878
331-10879	331-10887	331-10891	331-10892	331-10893	331-10894	331-10895	331-10896	331-10905
331-10906	331-10907	331-10908	333-10934	333-10937	333-10939	335-10949	335-10969	335-10980
335-10999	335-11010	335-11022	335-11032	335-11040	335-11042	363-11632	363-11669	380-12099
380-12102	380-12112	380-12117	382-12148	382-12178	384-12189	384-12193	384-12198	384-12202
386-12211	386-12212	386-12237	386-12241	388-12253	388-12254	388-12277	388-12281	388-12284
388-12286	388-12293	390-12305	390-12306	390-12315	390-12319	390-12324	390-12334	390-12337
393-12442	393-12449	393-12451	393-12468	395-12485	395-12487	395-12495	395-12502	397-12513
397-12520	397-12522	397-12538	399-12553	399-12562	399-12567	399-12573	399-12580	400-12599
400-12603	400-12614	400-12619	402-12633	404-12639	406-12651	429-13451	434-13574	465-14832
465-14833	466-14903	468-14931	468-14955	468-14956	468-14958	468-14965	468-14967	470-14994
470-15004	472-15035	472-15039	473-15073	473-15082	473-15097	473-15103	473-15105	478-15191
478-15192	478-15200	478-15209	478-15213	478-15215	478-15225	478-15228	478-15230	478-15234
478-15240	478-15243	478-15245	478-15249	478-15251	481-15352	482-15417	483-15462	483-15471
483-15472	483-15474	483-15483	483-15488	483-15497	483-15515	485-15524	485-15535	485-15537
485-15539	486-15605	486-15618	486-15624	486-15629	486-15633	486-15638	486-15639	486-15641
486-15644	486-15648	487-15709	487-15711	487-15712	487-15713	487-15768	487-15770	487-15771
487-15772	494-15990							
#110-4591	468-14936	472-15019	478-15199	478-15214	478-15216	478-15229	478-15231	478-15244
478-15246	483-15473							
#110-4592	211-7745	386-12234	388-12285	388-12294	395-12498	472-15057	472-15063	473-15104
485-15538	485-15540							
157-6111	159-6155	159-6176	160-6260	282-9508	291-9808	316-10395	319-10460	321-10512
322-10556	323-10634	324-10675	325-10737					
160-6259	291-9806	316-10393	323-10632	325-10736				
325-10734								
#285-9619	287-9688							
#285-9623	287-9684							
#111-4807	234-8361							
#111-4814	*234-8359							
111-4796	#111-4797	160-6192	*160-6227	*160-6234	160-6259	*160-6272	361-11520	369-11823
#111-4798	*222-8100	*236-8416	*236-8421					
#111-4799	169-6							

CZMSPA      CREATED BY      MACRO      ON 26-APR-82 AT 09:54      PAGE 25  
 SYMBOL      CROSS REFERENCE      CREF  
 SYMBOL      VALUE      REFERENCES

WARN4	027562	#261-8968 *327-10830
WARN5	027573	#261-8973 *327-10831
WARN6	041306	#328-10849
WARN6A	041216	#328-10842
WARN6B	041300	328-10841 #328-10848
WARN7	024140	#238-8439
WASDBE	= 104500	#110-4662
WASSBE	= 104476	#110-4660
WAS1DB	= 104501	#110-4663 174-6961 174-6974 174-6989 174-70 4 277-9414
WAS1SB	= 104477	#110-4661
WHICHC	053674	382-12170 384-12185 #406-12644
WOOPEN	055704	430-13479 430-13482 430-13484 432-13500 432-13523 #432-13531 432-13534
WOOPS	055336	425-13315 #430-13472
WOOPSA	055734	*430-13477 *430-13478 430-13479 432-13518 432-13519 432-13522 #432-13534
WOOPUP	055522	430-13479 430-13479 430-13480 430-13482 430-13482 430-13482 #432-13499 432-13523 432-13524
		432-13534
WGRST	002564	#141-5727 *147-5864 *197-7436 *199-7473 *203-7556 *207-7639 371-11944
XOCHAR	056276	#434-13588 *434-13622 *434-13625 *434-13626 434-13627 *434-13631 *434-13632 434-13633 483-15440
		483-15442 *483-15443
XXDPCH	002350	#139-5667 *153-5983 466-14902
ZEROS	002332	#139-5660 145-5824 296-9896 206-9900
\$APTHD	065740	137-5533 #493-15959
\$AUTO	002010	#139-5575 *153-5974 *153-5979 155-6051 483-15469
\$BANK	002011	#139-5555 *463-14792
\$BASE	065714	#492-15931
\$BELL	002637	#141-5753 393-12468 397-12538 465-14832
\$CACHF	042720	#341-11151 496-16027
\$CACHN	042674	#341-11144 496-16026
\$CBCSR	043360	#346-11247 496-16070
\$CBREG	044350	#355-11427 496-16065
\$CB1CS	043402	#346-11252 496-16071
\$CDW1	065720	#492-15934
\$CDW2	065722	#492-15935
\$CHARC	056464	*434-13576 *434-13586 *434-13644 #434-13649
\$CHKDI	043754	#352-11339 496-16078
\$CHK1D	043770	#352-11344 496-16079
\$CKSWR	063536	#483-15434 496-16010
\$CLRCS	043732	#352-11330 496-16076
\$CLR1C	043744	#352-11334 496-16077
\$CMTAG	002000	#139-5548 144-5795
\$CMTGE	002540	#139-5714 144-5797
\$CNTLC	064730	483-15483 486-15605 #486-15660
\$CNTLG	064742	483-15471 #486-15662
\$CNTLK	064136	483-15462 #483-15517
\$CNTLU	064735	483-15488 486-15633 #486-15661
\$CPJOP	065666	#492-15904
\$CRLF	002644	#141-5755 184-7187 331-10891 331-10891 395-12502 434-13575 468-14931 468-14958 468-14967
		478-15192 478-15213 478-15228 478-15243 478-15251 483-15497 485-15524 486-15638
\$OBLK	063526	482-15387 482-15417 #482-15425
\$O820	065510	473-15095 #491-15851
\$ODW0	065714	149-5898 #493-15946
\$ODW1	065726	#493-15947



CZMSPA CREATED BY MACRO ON 26-APR-82 AT 09:54

PAGE 26  
CREF

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
\$DDW2		065730	#493-15948
\$DDW3		065732	#493-15949
\$DDW4		065734	#493-15950
\$DDW5		065736	#493-15951
\$DEENE		042664	#341-11140 496-16022
\$DEVCT		065650	*189-7329 *462-14730 *462-14732 #492-15891
\$DEVN		065716	#492-15932
\$DOAGA		015346	189-7300 189-7302 #189-7336
\$DOAGN		015242	#189-7312
\$DOWN		054726	#426-13368 426-13369
\$DTBL		063516	482-15390 #482-15421
\$ECCDI		043256	#346-11223 496-16066
\$ECCIN		043304	#346-11231 496-16068
\$ECCID		043272	#346-11227 496-16067
\$ECCII		043320	#346-11235 496-16067
\$ENASB		043332	#346-11239 496-16080
\$ENAIS		043346	#346-11243 496-16081
\$ENDAD		015232	137-5524 153-5980 154-5991 #189-7308
\$ENERG		042654	#341-11136 496-16021
\$ENV		065660	144-5789 153-5973 #492-15896
\$ENVN		065661	153-5967 153-5970 #492-15900
\$EOP		015066	#189-7284
\$ERFLG		002012	#139-5556 462-14777 *462-14783 *465-14826
\$ERRGE		044100	#355-11373 496-16084
\$ERROR		060522	147-5876 #465-14822
\$ERRTB		066262	468-14946 #500-16111
\$ERRTY		061324	465-14877 #468-14930
\$ERTTL		002614	#141-5739 189-7290 395-12486 395-12488 *465-14834 *465-14836 466-14902
\$ESCAP		002356	#139-5673 *462-14786 466-14891 466-14893
\$ETABL		065660	#492-15895
\$ETEND		065740	#493-15955 493-15965
\$EXHAL		047656	#377-12027
\$ES	=	000001	#157-6114
\$FATAL		065642	#465-14868 #492-15888
\$FILLC		002636	#141-5752 434-13579
\$FILLS		002353	#139-5670 *399-12557
\$GTSWR		063712	#483-15472 496-16009
\$HALT		061106	#466-14886
\$HALT2		066014	#494-15991
\$HIBTS		065740	#493-15960
\$HIOCT		065140	386-12214 388-12256 *487-15704 #487-15715
\$ILLUP		055330	425-13300 428-13378 #429-13457 429-13458
\$INVAL		044050	#354-11364 496-16083
\$ITEMB		002013	#139-5557 *465-14843 465-14845 465-14855 *465-14856 468-14933
\$KERNE		042644	#341-11132 496-16020
\$KMAP		045100	#361-11589 496-16024
\$LF		002645	#141-5756 486-15648
\$LOADC		042736	#343-11161 496-16029
\$LPADR		002606	#141-5736 255-8824 *255-8834 255-8835 *255-8851 257-8857 *257-8867 257-8868 *257-8888
			*462-14781 *462-14784 462-14788
\$LPERR	002610		#141-5737 255-8824 *255-8835 *255-8851 257-8857 *257-8868 *257-8888 462-14781 *462-14785
			466-14889



CZMSPA SYMBOL SYMBOL SL\$	CREATED BY CROSS REFERENCE VALUE = 000000	MACRO REFERENCES	ON 26-APR-82 AT 09:54	PAGE 27 CREF	SEQ
		#157-6111 #159-6155 #159-6176 #160-6259 #160-6260 #282-9508 #291-9806 #291-9808 #316-10393			
		#316-10395 #319-10460 #321-10512 #322-10556 #323-10632 #323-10634 #324-10675 #325-10734 #325-10736			
		#325-10737			
SMADR1	065672	#492-15918			
SMADR2	065676	#492-15922			
SMADR3	065702	#492-15925			
SMADR4	065706	#492-15928			
SMAIL	065647	#492-15886 493-15961 493-15965			
SMAMS1	065670	186-7210 #492-15911			
SMAMS2	065674	#492-15920			
SMAMS3	065700	#492-15923			
SMAMS4	065704	#492-15926			
SMBADR	065742	#493-15961			
SMNFW	064760	483-15474 #486-15664			
SMSGAD	065654	#492-15893			
SMSGLG	065656	#492-15894			
SMSGTY	065640	*466-14899 #492-15887			
SMSWR	064747	483-15472 #486-15663			
SMTYP1	065671	#492-15912			
SMTYP2	065675	#492-15921			
SMTYP3	065701	#492-15924			
SMTYP4	065705	#492-15927			
SNOTRA	066010	#494-15990 496-16005 496-16007 496-16060 496-16061 496-16062 496-16063 496-16064 496-16086			
		496-16087 496-16088 496-16089 496-16090 496-16091			
SNULL	002352	#139-5669 434-13581			
SNWTST	= 000001	#157-6063 157-6063 #157-6063 #167-6074 167-6694 #167-6694 #169-6731 169-6731 #169-6731			
		#172-6879 172-6879 #172-6879 #187-7242 187-7242 #187-7242 #187-7259 187-7259 #187-7259			
SOCNT	063306	*481-15324 *481-15353 #481-15366			
SOCTVL	065622	491-15853 #491-15878 491-15381			
SOCT8	= 065626	473-15097 #491-15881			
SOMODE	063310	*481-15319 *481-15323 481-15328 *481-15331 *481-15342 #481-15368			
SOVER	060436	462-14761 462-14782 #462-14787			
SPASS	065646	*153-5966 *189-7291 *189-7292 189-7294 189-7298 189-7326 *189-7332 226-8179 226-8188			
		226-8203 227-8214 229-8228 229-8238 231-8253 234-8373 363-11614 395-12482 #492-15890			
SPASTM	065746	#493-15963			
SPATMA	002010	#139-5554 429-13447 429-13448 *463-14793 *463-14797 463-14802 463-14803 465-14829			
SPER01	056506	#451-14478 496-16032			
SPER02	056534	#451-14484 496-16033			
SPER03	056562	#451-14490 496-16034			
SPER04	056612	#451-14496 496-16035			
SPER07	056674	#451-14512 496-16036			
SPER10	056716	#451-14517 496-16037			
SPER11	056746	#451-14522 496-16038			
SPER12	056766	#451-14527 496-16039			
SPER13	057010	#453-14533 496-16040			
SPER14	057030	#453-14538 496-16041			
SPER15	057052	#453-14543 496-16042			
SPER16	057074	#453-14548 496-16043			
SPER17	057114	#453-14553 496-16044			
SPER20	057132	#453-14558 496-16045			
SPER21	057150	#453-14563 496-16046			
SPER22	057170	#453-14568 496-16047			

CZMSPA CREATED BY MACRO ON 26-APR-82 AT 09:54 PAGE 28  
 SYMBOL CROSS REFERENCE CREF

SYMBOL	VALUE	REFERENCES
\$PER23	057206	#453-14573 496-16048
\$PER24	057224	#453-14578 496-16049
\$PER25	053762	#421-13213 496-16050
\$PER26	057414	#456-14615 496-16051
\$PER27	057434	#456-14619 496-16052
\$PER30	054210	#423-13258 496-16053
\$PER31	057624	#458-14658 496-16054
\$PER32	057722	#460-14676 496-16055
\$PER33	057770	#460-14685 496-16056
\$PER34	060050	#460-14696 496-16057
\$PER35	060102	#460-14705 496-16058
\$PER36	060136	#460-14713 496-16059
\$PWDRN	054356	147-5880 #425-13287 429-13450
\$PWDRUP	054732	426-13367 #428-13374 432-13530
\$QUES	002643	#141-5754 483-15515 486-15641
\$RAND	065424	#489-15822
\$RDCHR	064256	#486-15567 496-16012
\$RDDEC	065142	#487-15730 496-16015
\$RDLIN	064406	#486-15596 496-16013
\$RDOCT	064772	#487-15679 496-16014
\$READC	043032	#343-11179 496-16030
\$RESRE	065366	#488-15803 496-16018
\$SAVRE	065330	#488-15792 496-16017
\$SAVR6	055334	*426-13365 428-13380 *428-13381 *428-13382 #429-13459 432-13513
\$SCOPE	060166	147-5874 #462-14730
\$STN	= 000001	#157-6063 #167-6694 #169-6731 #172-6879 #187-7242 #187-7259
\$SVLAD	060422	462-14769 462-14778 #462-14784
\$SWR	= 163000	#108-4554 157-6063 167-6694 169-6731 172-6879 187-7242 187-7259
\$SWREG	065662	153-5976 #492-15902
\$TESTN	065644	*465-14829 470-15011 #492-15889
\$TKB	002630	#141-5749 331-10873 331-10913 390-12348 390-12425 393-12454 393-12473 397-12525 397-12543
		434-13625 434-13631 483-15447 483-15479 485-15551 486-15571 486-15577
\$TKS	002626	#141-5748 331-10875 331-10912 390-12350 390-12423 393-12456 393-12472 397-12527 397-12542
		434-13623 434-13629 487-15445 483-15477 485-15549 486-15569 486-15575
\$TN	= 000007	#108-4555 157-6063 157-6063 #157-6063 167-6694 167-6694 #167-6694 169-6731 169-6731
		#169-6731 172-6879 172-6879 #172-6879 187-7242 187-7242 #187-7242 187-7259 187-7259
		#187-7259
\$TPB	002634	#141-5751 434-13638
\$TPFLG	002354	#139-5671 *153-5968 434-1355P
\$TPS	002632	#141-5750 434-13620
\$TRAP	065754	147-5878 #494-15975
\$TRAP2	065776	#494-15986 496-16001
\$TRPAD	066016	494-15980 #496-16001
\$TSTM	065744	#493-15962
\$TSTRD	043052	#344-11186 496-16082
\$TTYIN	064704	486-15598 486-15599 486-15621 486-15639 486-15653 #486-15657
\$TYPDS	063312	#482-15380 496-16006
\$TYPE	056152	#434-13558 496-16002
\$TYPEC	056300	434-13578 434-13585 #434-13589 483-15501
\$TYPEX	056466	434-13645 434-13647 #434-13650
\$TYPOC	063110	#481-15322 496-16003
\$TYPON	063124	481-15321 #481-15324

CZMSPA		CREATED BY	MACRO	ON 26-APR-82 AT 09:54	PAGE 29										
SYMBOL	CROSS REFERENCE	VALUE	REFERENCES		CREF										
\$TYPOS		063064	#481-15317	496-16004											
\$T2	=	000053	157-6111	159-6155	159-6176	160-6259	160-6260	282-9508	291-9806	291-9808	316-10393				
			316-10395	319-10460	321-10512	322-10556	323-10632	323-10634	324-10675	325-10734	325-10736				
			325-10737												
\$UNIT		065652	*189-7330	*462-14733	#492-15892										
\$UNITM		065750	#493-15964												
\$USWR		065664	189-7326	#492-15903											
\$VECT1		065710	#492-15929												
\$VECT2		065712	#492-15930												
\$WASDB		043566	#350-11294	496-16074											
\$WASSB		043422	#348-11259	496-16072											
\$WAS1D		043702	#350-11320	496-16075											
\$WAS1S		043536	#348-11285	496-16073											
\$XTSTR		060314	#462-14763												
\$ZAP42		015212	#189-7301	466-14906											
\$ST	=	000404	#157-6111	#157-6111	#159-6155	#159-6155	#159-6176	#159-6176	#160-6259	#160-6259	#160-6260				
			#160-6260	#282-9508	#282-9508	#291-9806	#291-9806	#291-9808	#291-9808	#291-9808	#316-10393	#316-10393			
			#316-10395	#316-10395	#319-10460	#319-10460	#321-10512	#321-10512	#322-10556	#322-10556	#323-10632				
			#323-10632	#323-10634	#323-10634	#324-10675	#324-10675	#325-10734	#325-10734	#325-10736	#325-10736				
			#325-10737	#325-10737											
\$OFILL		063307	*481-15318	*481-15322	481-15332	#481-15367									

CZMSPA CREATED BY MACRO ON 26-APR-82 AT 09:54

PAGE 30  
CREFMACRO CROSS REFERENCE  
MACRO NAME

REFERENCES

AND	#108-4544									
ANDB	#108-4547									
BEGIN	#108-4545	#182-7052	#209-7661	#211-7710	#348-11262	#350-11297	#354-11352	#363-11616	#363-11636	#400-12601
BMOV	#426-13338	#428-13406	#478-15194							
	#126-5283	169-6748	220-8022	220-8044	220-8070	222-8094	222-8097	222-8098	222-8099	222-8123
	222-8124	222-8143	222-8144	232-8296	232-8299	232-8303	232-8306	234-8355	234-8356	234-8357
	236-8412	236-8413	236-8415	236-8420	238-8438	240-8507	242-8553	244-8609	246-8647	246-8678
	328-10848	344-11203	363-11601	365-11740	367-11799	367-11815	430-13479	430-13482		
CASE	#108-4544	#215-7862	#380-12119							
CLEAR	#108-4542	#144-5792	#184-7151	#213-7797	#213-7804	#213-7852	#240-8530	#240-8541	#273-9289	#279-9461
	#283-9608	#327-10821	#331-10886	#331-10904	#352-11330	#352-11334	#363-11673	#365-11733	#365-11750	#371-11905
	#371-11907	#371-11908	#466-14912							
CLEARB	#108-4542									
DLEFT	#135-5485	267-9053	267-9072	273-9261	273-9267	275-9358	275-9364	279-9427	279-9433	283-9575
	283-9581	287-9677	287-9683							
DOWNT0	#108-4545									
ELSE	#108-4544	153-5977	153-5982	154-5995	158-6135	159-6162	159-6169	160-6199	160-6230	160-6248
	160-6255	167-6726	170-6867	170-6872	172-6909	173-6941	174-6979	174-6994	182-7064	184-7160
	184-7163	186-7217	217-7910	217-7925	226-8195	246-8686	253-8816	291-9766	291-9802	296-9898
	296-9914	297-9941	316-10365	316-10389	323-10618	325-10700	325-10715	325-10730	325-10748	335-10991
	335-10994	344-11209	348-11279	350-11314	377-12026	382-12153	382-12161	390-12368	393-12446	397-12517
	421-13221	421-13252	434-13635	454-14590	454-14602	458-14651	460-14699	466-14909	478-15233	478-15248
	485-15555									
END	#108-4545	149-5927	149-5929	151-5942	151-5959	153-5969	153-5972	153-5984	153-5985	153-5986
	154-5997	154-5998	155-6054	155-6055	157-6098	157-6104	157-6110	158-6134	158-6137	159-6164
	159-6165	159-6168	159-6171	160-6201	160-6212	160-6222	160-6232	160-6237	160-6250	160-6252
	160-6258	160-6270	167-6728	169-6841	169-6844	170-6870	170-6874	172-6911	172-6920	173-6931
	173-6932	173-6933	173-6934	173-6935	173-6936	173-6943	174-6981	174-6996	174-7005	174-7006
	174-7007	174-7110	#182-7063	#182-7066	182-7068	182-7069	182-7070	182-7071	184-7162	184-7165
	184-7166	184-7167	184-7176	184-7180	184-7181	184-7204	186-7219	186-7230	186-7233	186-7235
	186-7236	186-7239	186-7240	187-7267	189-7297	189-7334	189-7335	191-7348	191-7349	209-7665
	209-7666	211-7735	#211-7739	211-7740	211-7741	211-7742	211-7748	211-7749	211-7751	213-7818
	213-7843	215-7903	217-7912	217-7919	217-7927	219-7971	#226-8180	#226-8189	226-8197	#226-8204
	#227-8215	#229-8229	#229-8239	#231-8254	#234-8374	238-8457	238-8463	238-8464	238-8465	238-8466
	238-8470	238-8486	238-8492	238-8493	238-8494	238-8495	240-8521	240-8522	240-8523	240-8532
	240-8539	246-8695	253-8818	273-9226	273-9227	282-9485	282-9492	282-9506	291-9768	291-9786
	291-9798	291-9805	296-9901	296-9917	296-9920	297-9936	297-9940	297-9945	297-9949	297-9950
	297-9953	297-9956	297-9957	297-9964	297-9969	315-10346	316-10367	316-10385	316-10392	317-10420
	319-10457	321-10497	321-10507	322-10543	322-10553	323-10599	323-10613	323-10630	324-10662	324-10671
	325-10702	325-10717	325-10725	325-10733	325-10747	325-10752	325-10753	326-10783	326-10793	335-10993
	335-10996	344-11211	#348-11270	348-11271	348-11273	348-11274	348-11281	#350-11305	350-11306	350-11308
	350-11309	350-11316	354-11357	354-11359	354-11360	#363-11615	#363-11629	363-11630	363-11633	363-11635
	#363-11648	363-11652	363-11653	363-11654	363-11655	363-11656	363-11663	363-11664	363-11670	363-11672
	363-11690	371-11919	371-11949	377-12029	377-12045	380-12105	380-12143	382-12155	382-12165	390-12317
	390-12327	390-12336	390-12365	390-12372	393-12448	393-12466	393-12467	395-12497	395-12503	395-12504
	395-12505	397-12519	397-12536	397-12537	#400-12608	400-12612	400-12613	400-12628	421-13218	421-13223
	421-13254	423-13264	426-13344	426-13346	426-13347	428-13412	428-13414	428-13415	434-13637	454-14592
	454-14604	458-14653	458-14660	458-14665	458-14668	458-14671	460-14701	462-14734	462-14757	462-14760
	465-14837	465-14838	465-14867	465-14872	465-14875	465-14876	465-14881	466-14890	466-14901	466-14907
	466-14908	466-14911	470-15003	478-15201	478-15203	478-15204	478-15217	478-15232	478-15233	478-15247
	478-15250	485-15536	485-15557							
FATAL	#115-4987	167-6715	167-6719	303-10062	337-11079	339-11104	339-11108	339-11111		

CZMSPA CREATED BY MACRO ON 26-APR-82 AT 09:54

PAGE 31

MACRO CROSS REFERENCE

CREF

MACRO NAME  
FOR

REFERENCES

GOTO  
IF

IFB

JUMPTO  
LEAVE  
LET

#108-4545	149-5923	172-6899	182-7041	184-7155	184-7170	184-7177	186-7211	191-7341	211-7706
211-7707	211-7715	211-7726	238-8442	238-8443	238-8471	238-8472	240-8514	296-9894	296-9902
297-9962	315-10322	#348-11263	#350-11298	#354-11353	#363-11617	363-11640	393-12461	395-12490	397-12531
400-12626	#426-13339	#428-13407	#478-15195	478-15212	478-15227	478-15242			
#108-4546	#90-12316	393-12469	397-12539	465-14874					
#108-4544	151-5937	151-5952	#153-5978	153-5980	154-5991	154-5992	155-6051	155-6052	157-6096
157-6101	158-6128	158-6132	159-6158	159-6159	159-6160	159-6166	160-6197	160-6207	160-6217
160-6228	160-6235	160-6243	160-6246	160-6253	160-6261	160-6265	167-6724	169-6838	169-6842
170-6863	170-6864	172-6903	172-6904	172-6905	172-6906	172-6916	172-6918	173-6923	173-6939
174-6976	174-6991	182-7045	182-7050	#182-7053	182-7059	#184-7156	184-7157	184-7158	#184-7178
184-7186	184-7202	186-7208	186-7215	186-7237	187-7262	189-7294	189-7325	189-7326	191-7343
#209-7662	209-7663	211-7679	#211-7680	211-7712	211-7743	213-7811	213-7817	213-7836	213-7842
217-7908	217-7917	217-7923	219-7969	222-8076	226-8178	226-8179	226-8181	226-8187	226-8188
226-8190	226-8193	226-8202	226-8203	226-8205	227-8213	227-8214	227-8216	229-8227	229-8228
229-8230	229-8236	#229-8237	229-8238	231-8252	231-8253	231-8256	234-8372	234-8373	234-8375
238-8445	238-8452	238-8458	238-8467	238-8474	238-8481	238-8487	240-8516	240-8517	240-8524
246-8680	246-8713	247-8720	248-8732	249-8747	250-8756	251-8766	251-8774	251-8782	251-8790
251-8798	253-8814	273-9222	273-9224	273-9263	273-9264	275-9360	275-9361	279-9429	279-9430
282-9483	282-9488	282-9501	283-9577	283-9578	287-9679	287-9680	291-9782	291-9793	296-9895
#296-9909	#296-9910	#296-9911	296-9923	#297-9930	#297-9931	#297-9932	297-9934	297-9938	297-9943
297-9947	316-10379	317-10416	319-10453	321-10493	321-10502	322-10539	322-10548	323-10593	323-10607
324-10653	324-10667	325-10713	325-10721	325-10743	325-10744	#325-10749	326-10778	326-10788	331-10883
335-10970	335-10988	335-10989	335-11000	335-11023	343-11163	344-11207	348-11267	348-11272	350-11302
350-11307	354-11358	363-11612	#363-11613	363-11614	363-11619	363-11631	#363-11641	363-11642	363-11643
363-11644	363-11645	363-11649	363-11657	363-11658	363-11666	371-11917	371-11947	375-12002	377-12024
377-12042	#377-12043	380-12101	382-12150	382-12159	388-12263	388-12265	388-12266	390-12309	390-12314
390-12322	390-12323	390-12333	390-12361	390-12366	393-12444	393-12463	395-12488	395-12494	397-12515
397-12533	400-12606	421-13214	421-13219	421-13227	421-13250	423-13260	425-13315	426-13345	428-13413
434-13633	454-14585	#454-14586	454-14588	454-14597	#454-14598	454-14600	456-14609	458-14649	458-14658
458-14661	458-14663	458-14666	458-14669	460-14676	460-14685	460-14696	#460-14697	462-14731	462-14754
462-14758	462-14761	465-14823	465-14835	465-14860	465-14861	465-14870	465-14873	465-14878	466-14883
466-14888	466-14894	466-14898	466-14902	470-15001	478-15202	478-15226	478-15241	485-15534	485-15553
#108-4547	153-5967	153-5970	153-5973	186-7212	186-7228	186-7231	291-9764	291-9800	316-10363
316-10387	323-10615	325-10698	325-10728	395-12493					
#108-4546	157-6118								
#108-4546	#182-7062	#182-7065	#211-7738	#348-11269	#350-11304	#363-11628	#363-11647	#400-12607	
#108-4546	#157-6095	#157-6116	#158-6127	#158-6130	#158-6131	#160-6193	#160-6195	#160-6196	#160-6198
#160-6200	#160-6202	#160-6204	#160-6205	#160-6208	#160-6209	#160-6213	#160-6214	#160-6218	#160-6219
#160-6223	#160-6227	#160-6229	#160-6231	#160-6238	#160-6239	#160-6241	#160-6244	#160-6245	#160-6262
#160-6263	#160-6266	#160-6267	#160-6271	#184-7159	#184-7161	#184-7164	#184-7179	#211-7729	#227-8219
#231-8258	#238-8447	#238-8475	#249-8750	#249-8751	#251-8769	#251-8777	#251-8785	#251-8793	#251-8801
#251-8802	#282-9474	#282-9475	#282-9476	#282-9478	#282-9480	#282-9481	#282-9490	#282-9499	#282-9503
#282-9504	#291-9761	#291-9762	#291-9763	#291-9765	#291-9767	#291-9771	#291-9772	#291-9775	#291-9777
#291-9783	#291-9784	#291-9790	#291-9795	#291-9796	#296-9896	#296-9897	#296-9899	#296-9900	#296-9906
#296-9907	#296-9912	#296-9913	#296-9915	#296-9916	#296-9918	#296-9919	#296-9925	#296-9926	#297-9933
#297-9937	#297-9942	#297-9946	#297-9951	#297-9952	#297-9965	#297-9968	#316-10360	#316-10361	#316-10362
#316-10364	#316-10366	#316-10371	#316-10372	#316-10375	#316-10377	#316-10380	#316-10381	#316-10382	#316-10383
#317-10410	#317-10411	#317-10413	#317-10415	#317-10417	#317-10418	#319-10430	#319-10433	#319-10434	#319-10435
#319-10440	#319-10441	#319-10442	#319-10444	#319-10448	#319-10450	#319-10454	#319-10455	#319-10458	#321-10473
#321-10478	#321-10479	#321-10486	#321-10488	#321-10494	#321-10495	#321-10498	#321-10503	#321-10504	#321-10508
#321-10509	#322-10528	#322-10529	#322-10532	#322-10534	#322-10536	#322-10540	#322-10541	#322-10550	#322-10551
#322-10555	#323-10568	#323-10569	#323-10571	#323-10573	#323-10576	#323-10578	#323-10579	#323-10580	#323-10582

PAGE 32  
CREF

**CREF**

## REFERENCES

BACKLOG NAME	1	2	3	4	5	6	7	8	9	10
	#323-10588	#323-10591	#323-10594	#323-10595	#323-10596	#323-10605	#323-10608	#323-10609	#323-10610	#323-10631
	#323-10633	#324-10648	#324-10649	#324-10650	#324-10654	#324-10660	#324-10668	#324-10673	#324-10674	#325-10689
	#325-10694	#325-10695	#325-10696	#325-10697	#325-10699	#325-10701	#325-10704	#325-10706	#325-10707	#325-10710
	#325-10712	#325-10722	#326-10763	#326-10768	#326-10770	#326-10771	#326-10773	#326-10779	#326-10780	#326-10789
	#326-10790	#363-11677	#365-11737	#421-13213	#421-13216	#421-13220	#421-13222	#421-13239	#423-13258	#423-13259
	#423-13262	#451-14517	#451-14518	#451-14519	#451-14522	#451-14523	#451-14524	#451-14527	#451-14528	#451-14529
	#453-14533	#453-14534	#453-14535	#453-14538	#453-14539	#453-14540	#453-14543	#453-14544	#453-14545	#453-14548
	#453-14549	#453-14550	#453-14553	#453-14554	#453-14555	#453-14558	#453-14559	#453-14560	#453-14563	#453-14564
	#453-14555	#453-14568	#453-14569	#453-14570	#453-14573	#453-14574	#453-14575	#453-14578	#453-14579	#453-14580
	#456-14615	#456-14616	#456-14620							
MAP	#123-5346	#155-6059	#169-6761	#173-6937	#174-6948	#211-7708	#242-8551	#244-8580	#246-8644	#255-8822
	#257-9855	#377-12059	#386-12230	#388-12269	#390-12330	#390-12363	#390-12428	#393-12475	#397-12545	#430-13475
	#432-13516									
NEWST ON.ERR	#117-5037	#157-6063	#167-6694	#169-6731	#172-6879	#187-7242	#187-7259			
	#108-4547	157-6108	174-7008	193-7378	195-7408	197-7445	199-7482	201-7520	203-7565	205-7603
	207-7648	233-8313	233-8321	234-8331	240-8528	242-8547	244-8576	246-8640	348-11265	348-11277
	350-11300	350-11312	354-11355	363-11688	426-13341	428-13409	478-15197			
ON.NOE OR ORB POP	#108-4547	149-5925	174-6965	174-6975	174-6990	211-7723				
	#108-4544									
	#108-4547									
	#108-4546	#155-6059	#160-6272	#166-6540	#166-6637	#169-6761	#169-6837	#173-6937	#174-6948	#174-7021
	#174-7023	#182-7104	#189-7306	#211-7708	#211-7734	#211-7761	#213-7848	#213-7856	#242-8551	#244-8580
	#246-8644	#255-8822	#255-8851	#257-8855	#257-8888	#327-10832	#331-10914	#343-11174	#343-11182	#344-11206
	#348-11276	#350-11311	#354-11361	#354-11370	#355-11408	#355-11424	#357-11461	#361-11586	#361-11607	#365-11732
	#365-11758	#365-11770	#369-11852	#369-11873	#371-11959	#377-12059	#379-12091	#380-12114	#382-12154	#382-12156
	#382-12164	#384-12195	#386-12215	#386-12230	#386-12245	#388-12257	#388-12269	#388-12288	#388-12297	#390-12308
	#390-12321	#390-12326	#390-12330	#390-12363	#390-12371	#390-12424	#390-12426	#390-12427	#390-12428	#393-12474
	#393-12475	#395-12506	#397-12544	#397-12545	#399-12555	#399-12558	#400-12			





CZMSPA CREATED BY MACRO ON 26-APR-82 AT 09:54

PAGE 34  
CREF

MACRO CROSS REFERENCE

MACRO NAME

THRU

TO

TYPDEC

TYPE

REFERENCES

#108-4545

#108-4545

#125-5247

#115-5001

159-6156

184-7201

331-10892

333-10937

335-11042

384-12193

388-12281

390-12337

397-12520

400-12614

468-14931

473-15073

478-15215

481-15352

485-15524

486-15639

487-15771

TYPOCS

#123-5194

#485-15540

TYPOCT

#121-5148

483-15473

UNTIL

#108-4545

325-10734

UNTILB

#108-4547

USER

#130-5396

WHILE

#108-4544

WHILEB

#108-4547

\$CALL

#108-4548

\$\_RETUR

#119-5109

\$\_SUBTS

153-5963

166-6449

197-7415

217-7907

224-8168

233-8311

246-8668

251-8789

263-8980

277-9576

301-10026

313-10261

325-10680

354-11364

369-11842

377-12022

390-12302

400-12598

#184-7190

#154-5996

159-6172

189-7293

331-10893

333-10939

363-11632

384-12198

388-12284

393-12442

397-12522

400-12619

468-14955

473-15082

478-15225

482-15417

485-15535

486-15641

487-15772

#211-7745

468-14936

157-6111

325-10737

291-9808

363-11680

296-9908

297-9929

297-9966

#213-7849

144-5788

154-5988

167-6723

199-7452

219-7962

226-8177

233-8319

246-8698

251-8797

263-8988

282-9467

301-10034

314-10307

326-10756

355-11373

369-11855

377-12031

391-12433

400-12618

#184-7194

154-6010

159-6173

189-7295

331-10894

335-10949

363-11669

384-12202

388-12286

393-12449

397-12538

402-12633

468-14956

473-15097

#478-15228

483-15462

485-15537

486-15644

494-15990

#386-12234

472-15019

159-6155

316-10395

365-11739

367-11798

297-9929

297-9966

#213-7857

144-5800

154-6000

170-6856

201-7489

220-8012

226-8186

234-8329

247-8719

253-8808

265-8998

283-9515

303-10044

315-10319

327-10802

355-11427

369-11865

377-12058

393-12439

402-12632

#184-7198

154-6024

159-6177

211-7744

331-10895

335-10969

380-12099

386-12211

388-12293

393-12451

399-12553

404-12639

468-14958

473-15103

478-15230

483-15471

485-15539

486-15648

#388-12285

478-15199

159-6176

319-10460

367-11798

367-11814

426-13326

429-13433

#253-8815

145-5813

155-6032

182-7036

203-7527

220-8025

226-8201

234-8371

248-8731

253-8813

265-9011

285-9614

307-10138

316-10351

327-10808

357-11434

369-11876

379-12068

395-12480

404-12638

#184-7200

154-6029

163-6416

211-7746

331-10896

335-10980

380-12102

386-12212

390-12305

393-12468

399-12562

406-12651

468-14965

473-15105

#478-15234

483-15472

486-15605

487-15709

#388-12294

478-15214

160-6259

321-10512

367-11814

426-13326

429-13433

#363-11634

147-5863

155-6050

184-7150

205-7572

220-8047

227-8212

236-8382

248-8743

255-8821

267-9034

289-9716

307-10145

317-10404

328-10835

361-11503

371-11883

380-12097

397-12510

406-12644

#189-7298

154-6030

182-7145

331-10877

331-10905

335-10999

380-12112

386-12237

390-12306

395-12485

399-12567

429-13451

468-14967

478-15191

478-15240

483-15474

486-15618

487-15711

#395-12498

478-15216

160-6260

322-10556

426-13326

429-13433

#363-11671

149-5893

158-6120

186-7207

207-7610

222-8075

229-8226

238-8429

249-8746

259-8902

269-9077

291-9750

307-10163

319-10425

329-10854

363-11611

373-11963

382-12147

399-12551

421-13226

#395-12484

159-6142

#184-7187

331-10878

331-10906

335-11010

380-12117

386-12241

390-12315

395-12487

399-12573

434-13574

470-14994

478-15192

#478-15243

483-15483

486-15624

487-15712

#472-15057

478-15229

282-9508

323-10632



CZMSPA CREATED BY MACRO ON 26-APR-82 AT 09:54 PAGE 35  
 MACRO CROSS REFERENCE CREF

MACRO NAME	REFERENCES
	454-14595 456-14608 456-14623 458-14632 463-14791 476-15150 485-15522 485-15547
SEND	#108-4546 #517-16630
NEW	#119-5069 #157-6063 #167-6694 #169-6731 #172-6879 #187-7242 #187-7259
ARITH	#108-4547
EMIT	#108-4541
EMITL	#108-4541 #157-6111 #159-6155 #159-6176 #160-6259 #160-6260 #282-9508 #291-9806 #291-9808 #316-10393
	#316-10395 #319-10460 #321-10512 #322-10556 #323-10632 #323-10634 #324-10675 #325-10734 #325-10736 #325-10737
EMITN	#108-4541 157-6114
EMITR	#108-4541
GENBR	#108-4542
GOTO	#108-4544
IFARI	#108-4544
IFOPR	#108-4542
IS	#108-4542
LEAVE	#108-4544
OPADD	#108-4542
OPSUB	#108-4542
OR	#108-4544
SIMPL	#108-4547
BRAN	#108-4541
POP	#108-4541 #157-6111 #159-6155 #159-6176 #160-6259 #160-6260 #282-9508 #291-9806 #291-9808 #316-10393
	#316-10395 #319-10460 #321-10512 #322-10556 #323-10632 #323-10634 #324-10675 #325-10734 #325-10736 #325-10737
PUSH	#108-4541
TAG	#108-4541 157-6111 159-6155 159-6176 160-6259 160-6260 282-9508 291-9806 291-9808 316-10393
	316-10395 319-10460 321-10512 322-10556 323-10632 323-10634 324-10675 325-10734 325-10736 325-10737