

VT55

ACCEPTANCE TEST
MD-11-DZVTD-B

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

NOV 1976
digital
MADE IN U.S.A.

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
13						

.REM

%

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZVTD-B
PRODUCT NAME: VTSS ACCEPTANCE TEST
DATE: AUG. 21, 1976
MAINTAINER: DIAGNOSTIC GROUP

COPYRIGHT (C) 1975, 1976
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

1.0 ABSTRACT

**** THE PROGRAM WILL RUN ON NON-SWITCH REGISTER CPU TYPES ****

THIS PROGRAM IS AN ACCEPTANCE TEST OF THE VT55 VIDEO TERMINAL. THE PROGRAM CONSISTS OF 13 TEST PATTERNS DISPLAYED ON THE VT55 SCREEN. EACH PATTERN REQUIRES OPERATOR INSPECTION FOR ERROR DETECTION. A DESCRIPTION OF THE CORRECT VISUAL DISPLAY FOR EACH TEST CAN BE FOUND IN SECTION 9. THE SCREEN WILL BE COPIED IN 9 OF THE TEST PATTERNS.

THE PROGRAM IS CAPABLE OF HANDLING MULTIPLE VT55'S IN A SEQUENTIAL DL-11 FASHION, HOWEVER:

ONLY ONE VT55 IS TESTED AT ONE TIME.

2.0 REQUIREMENTS

2.1 EQUIPMENT

POP-11 FAMILY COMPUTER WITH 4K OF MEMORY.
VT55 VIDEO GRAPHIC TERMINAL
DL-11 TYPE INTERFACE

2.2 STORAGE

THIS PROGRAM USES 4K OF MEMORY.

3.0 LOADING PROCEDURE

PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED.

4.0 STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

STANDARD PDP-11 FORMAT

SW 15 = 1	HALT ON ERROR
SW 14 = 1	LOOP ON TEST
SW 13 = 1	INHIBIT ERROR TYPEOUTS
SW 12 = 1	INHIBIT PROGRAM SUB-TEST DELAY
SW 11 = 1	INHIBIT COPIER TESTING
SW 08 = 1	LOOP ON TEST IN SWR <4:0>
SW 07 = 1	KEYBOARD CONTROL OF THE TEST
	<SW 8 AND SW 7 = 1 IS AN ERROR>

4.2 STARTING ADDRESS OR ADDRESSES

200 IS THE STARTING ADDRESS OF THE ACCEPTANCE TEST
204 IS THE RESTART ADDRESS OF THE ACCEPTANCE TEST
210 IS THE SPECIAL STARTING ADDRESS FOR VT-55 PRODUCTION

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

5.0 OPERATING PROCEDURE

THE OPERATOR MUST INSERT THE CORRECT INFORMATION IN THE SWITCH REGISTER WHEN REQUIRED BY THE PROGRAM OR AN ERROR WILL OCCUR. ONCE STARTED, THE TEST WILL RUN IN ITS NORMAL MANNER WITHOUT OPERATOR INTERVENTION OR SWITCH CHANGES.

THIS PROGRAM ALLOWS THE OPERATOR TWO MODES OF TEST PATTERN SELECTION. THESE MODES ARE SELECTED BY THE STATE OF SW 07 AT THE BEGINNING OF THE PROGRAM. WHEN SW 07 IS A ZERO, THE PROGRAM IS UNDER SWITCH REGISTER CONTROL FOR TEST PATTERN SELECTION. IF SW 07 IS EQUAL TO A ONE, THE PROGRAM IS UNDER KEYBOARD CONTROL OF THE TEST PATTERN SELECTION. IN THIS MODE THE OPERATOR WILL BE REQUIRED TO TYPE IN ON THE CONSOLE TTY THE FIRST AND LAST OCTAL BASE ADDRESS OF THE DL-11'S TO WHICH VT-55'S ARE CONNECTED.

KEYBOARD CONTROL IS ASSUMED IF RUNNING ON A SWITCH REGISTER LESS CPU TYPE (I.E., 11/04 LSI-11).

IN THE KEYBOARD SELECT MODE, TWO CHARACTERS ARE USED TO SELECT THE "STARTING WITH" OR "LOOPING ON" PARTICULAR TEST PATTERN BY "/" OR "\" RESPECTFULLY.

THE "/" KEY IS USED TO SUSPEND THE CURRENT TEST AND ASK THE OPERATOR AT WHICH TEST PATTERN HE/SHE WISHES TO START. THE OPERATOR NOW DEPRESSES THE LETTER WHICH REPRESENTS THE TEST PATTERN TO BE STARTED WITH. REFER TO THE PROGRAM LISTING TABLE OF CONTENTS FOR THE TEST LETTER OF EACH PATTERN.

THE "\" KEY IS USED TO SUSPEND THE CURRENT TEST AND ASK THE OPERATOR WHICH TEST PATTERN HE/SHE WISHES TO LOOP ON. THE OPERATOR NOW DEPRESSES THE LETTER OF THE TEST TO LOOP ON.

IF DURING THE EXECUTION OF A TEST PATTERN, A KEY IS DEPRESSED AND SW 07 EQUALS A ZERO, AN ERROR WILL BE REPORTED TO THE CONSOLE TTY. IF SW 07 EQUALS A ONE, AND THE CHARACTER RECEIVED WAS NOT A "/" OR "\" AN ERROR WILL BE REPORTED. THE CODES "X-OFF" AND "X-ON" ARE THE ONLY EXCEPTIONS.

6.0 ERRORS

THIS PROGRAMS USES THE DIAGNOSTIC 'SYSMAC' PACKAGE FOR ERROR REPORTING AND TYPEOUT. REFER TO THE "ERROR POINTER TABLE" FOR TYPE AND DESCRIPTION OF ERRORS. THE ERROR INFORMATION CONSISTS OF THE FOLLOWING:

ERRPC - LOCATION AT WHICH AN ERROR WAS DETECTED
VTNOW - CURRENT DL-11 BUS ADDRESS OF VT55 UNDER TEST
TSTNUM - TEST PATTERN NUMBER OF FAILING TEST

F01

MAINDEC-11-DZVTD-8
DZVTD8.SRC

MACY11 27(732) 25-SEP-76 10:10 PAGE 6

182
183

EXPT - EXPECTED INPUT CHARACTER
RCVD - RECEIVED INPUT CHARACTER

7.0 MISCELLANEOUS

1. THE OPERATOR SHOULD SET SW 15 AND 13 IF THE VT55 UNDER TEST IS THE CONSOLE TTY.
2. ONLY ONE VT55 CAN BE TESTED AT ONE TIME.
3. EXECUTION TIME
EXECUTION TIME WILL VARY WITH THE "BAUD" RATE AND IF A "COPIER" IS CONNECTED.
4. DEVICE ADDRESS PROGRAM LOCATIONS
THE LOCATION "SBASE" CONTAINS THE FIRST DL11 ADDRESS IF SEVERAL VT-55'S ARE BEING TESTED. THE DEFAULT IS THE CONSOLE ADDRESS (177560). THE LOCATION "LAST" CONTAINS THE LAST DL11 ADDRESS IF SEVERAL VT-55'S ARE BEING TESTED. LOCATION VTNOW CONTAINS THE CURRENT DL11 BASE ADDRESS.

#NOTE: IF THESE LOCATIONS ARE CHANGED, THE OPERATOR MUST START THE TEST AGAIN AT LOC. 200. THE PROGRAM WILL USE THE BASE ADDRESS TO UPDATE THE ACTUAL PROGRAM VALUES.
5. PROGRAM IS CHAINABLE UNDER XXDP/ACT-11. APT HOOKS HAVE BEEN PROVIDED BUT NOT TESTED.

8.0 RESTRICTIONS

NONE.

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

9.0 PROGRAM DESCRIPTION

9.1 GROWING HORIZONTAL LINE

THE CORRECT VISUAL DISPLAY WILL BE A SINGLE HORIZONTAL LINE EXTENDING THE ENTIRE WIDTH OF THE SCREEN PLACED AT BASE ZERO OF THE SCREEN. ANOTHER HORIZONTAL LINE WILL SUCCESSIVELY APPEAR GIVING THE IMPRESSION OF A GROWING HORIZONTAL LINE UNTIL THE ENTIRE SCREEN HAS BEEN FILLED. IF "COPIER" TESTING IS ENABLE, THE SCREEN WILL NOW BE COPIED. THEN THE FIRST LINE AT THE BASE OF THE SCREEN WILL BE REMOVED, FOLLOWED BY EACH SUCCESSIVE LINE UNTIL THE ENTIRE BLOCK HAS DISAPPEARED. IF "COPIER" TESTING IS ENABLE, THE SCREEN WILL NOW BE COPIED.

9.2 GROWING VERTICAL LINE

THE CORRECT VISUAL DISPLAY WILL BE A SINGLE VERTICAL LINE EXTENDING THE ENTIRE HEIGHT OF THE SCREEN AND PLACED AT THE FAR RIGHT SIDE OF THE SCREEN. ANOTHER VERTICAL LINE WILL SUCCESSIVELY APPEAR, GIVING THE IMPRESSION OF A GROWING VERTICAL LINE RIGHT TO LEFT UNTIL THE ENTIRE SCREEN HAS BEEN FILLED. IF "COPIER" TESTING IS ENABLE, THE SCREEN WILL NOW BE COPIED. THEN THE FIRST LINE AT THE RIGHT WILL BE REMOVED, FOLLOWED BY EACH SUCCESSIVE LINE UNTIL THE ENTIRE BLOCK HAS DISAPPEARED. IF "COPIER" TESTING IS ENABLE, THE SCREEN WILL NOW BE COPIED.

9.3 STEPPING HORIZONTAL LINE FOR GRAPH 0

THE CORRECT VISUAL DISPLAY WILL BEGIN WITH A SINGLE HORIZONTAL LINE APPEARING NEAR THE CENTER OF THE SCREEN AND EXTENDING THE ENTIRE WIDTH OF THE SCREEN. THEN A SECOND HORIZONTAL LINE HALFWAY BETWEEN THE FIRST LINE AND THE BASE OF THE SCREEN WILL BEGIN TO GROW FROM THE LEFT SIDE OF THE SCREEN. AS THIS LINE GROWS, THE FIRST LINE DISAPPEARS AND YOU'RE LEFT WITH A SINGLE HORIZONTAL LINE ABOUT ONE FOURTH THE WAY UP THE SCREEN. THEN ANOTHER LINE BEGINS TO GROW FROM THE LEFT, HALFWAY BETWEEN THE PREVIOUS LINE AND THE BASE OF THE SCREEN. AS THIS ONE GROWS THE PREVIOUS LINE IS REMOVED. THIS PROCEDURE CONTINUES FOR A TOTAL OF EIGHT TIMES.

9.4 STEPPING HORIZONTAL LINE FOR GRAPH 1

SAME AS 9.3 EXCEPT GRAPH 1 IS ENABLED.

9.5 DIFFERENT DATA ON GRAPH 0 AND 1

GRAPH 0:

THE CORRECT VISUAL DISPLAY WILL BEGIN WITH THE APPEARANCE OF A HORIZONTAL LINE EXTENDING THE ENTIRE WIDTH OF THE SCREEN PLACED AT THE BASE ZERO. AS THIS LINE IS REMOVED FROM THE LEFT TO RIGHT, A DIAGONAL LINE BEGINNING AT THE LEFT BOTTOM CORNER, BEGINS TO GROW UNTIL IT REACHES THE TOP OF THE SCREEN. AT THIS POINT A SECOND DIAGONAL LINE BEGINS TO GROW UP FROM THE BASE LINE ABOUT IN THE MIDDLE OF THE SCREEN. THIS LINE CONTINUES TO GROW UP UNTIL IT REACHES THE TOP OF THE SCREEN.

GRAPH 1:

THEN A HORIZONTAL LINE REAPPEARS AT BASE ZERO OF THE SCREEN EXTENDING THE ENTIRE WIDTH OF THE SCREEN. NOW A DIAGONAL LINE BEGINNING AT THE TOP OF THE LEFT CORNER OF THE SCREEN BEGINS TO DECAY DOWNWARD AS THE BASE HORIZONTAL LINE DISAPPEARS. IT CONTINUES UNTIL IT REACHES THE BASE LINE OF THE SCREEN WHEN A SECOND DIAGONAL LINE BEGINS TO DECAY DOWNWARD FROM THE TOP MIDDLE SECTION OF THE SCREEN AND CONTINUES UNTIL IT REACHES THE BASE LINE OF THE SCREEN. THE END RESULT SHOULD BE TWO LARGE X'S FILLING UP THE ENTIRE SCREEN (XX). IF "COPIER" TESTING IS ENABLE, THE SCREEN WILL NOW BE COPIED.

9.6 DISPLAY A STEPPING HISTOGRAM LINE ON GRAPH 0

THE CORRECT VISUAL DISPLAY WILL BEGIN WITH THE APPEARANCE OF A HORIZONTAL LINE AT BASE ZERO ON THE SCREEN. THEN A LINE HALFWAY UP THE SCREEN WILL BEGIN TO GROW FROM THE LEFT SIDE OF THE SCREEN WITH ALL THE AREA BETWEEN THE TWO LINES SHADED. THIS SHADED AREA WILL CONTINUE TO GROW UNTIL IT REACHES THE FAR RIGHT SIDE OF THE SCREEN. THEN A LINE WHICH BISECTS THE SHADED AREA BEGINS TO GROW FROM THE LEFT SIDE. AS THIS LINE GROWS IT REMOVES THE SHADED AREA ABOVE IT, CONTINUING UNTIL IT REACHES THE RIGHT SIDE OF THE SCREEN. THE SHADED AREA IS THEN CUT IN HALF AGAIN AND AGAIN UNTIL A SINGLE HORIZONTAL LINE REMAINS.

9.7 DISPLAY A STEPPING HISTOGRAM LINE ON GRAPH 1

SAME AS 9.6 EXCEPT GRAPH 1 IS ENABLED.

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

9.8 HISTOGRAM ON GRAPH 0 AND 1

THE CORRECT VISUAL DISPLAY WILL FOLLOW THE SAME PATTERN AS THE VISUAL DISPLAY FOR GRAPH 0 AND 1, EXCEPT THAT AS EACH DIAGONAL LINE GROWS, A TRIANGULAR SHADED AREA GROWS UNDER IT. THE FINAL RESULT SHOULD CONSIST OF FOUR OVERLAPPING RIGHT TRIANGLES, TWO WITH THE RIGHT ANGLE ON THE RIGHT BOTTOM OF THE SCREEN (MADE BY THE DIAGONAL LINES WHICH STARTED FROM THE BOTTOM LEFT) AND TWO WITH THE RIGHT ANGLE ON THE LEFT BOTTOM OF THE SCREEN (MADE BY THE DIAGONAL LINES WHICH STARTED FROM THE TOP LEFT). IF "COPIER" TESTING IS ENABLE, THE SCREEN WILL NOW BE COPIED.

9.9 CURSORS ON GRAPH 0

THE CORRECT VISUAL DISPLAY WILL BEGIN WITH A SINGLE HORIZONTAL LINE EXTENDING THE ENTIRE LENGTH OF THE SCREEN PLACED AT BASE ZERO. A DIAGONAL LINE WILL THEN BEGIN TO GROW FROM THE BOTTOM LEFT CORNER AS THE BASE HORIZONTAL LINE IS REMOVED. IT CONTINUES TO GROW UNTIL IT REACHES THE TOP OF THE SCREEN WHEN A SECOND DIAGONAL LINE BEGINS TO GROW FROM THE BOTTOM OF THE MIDDLE OF THE SCREEN. THIS LINE CONTINUES TO GROW AS THE BASE HORIZONTAL LINE CONTINUES TO BE REMOVED. WHEN THE DIAGONAL LINE REACHES THE TOP OF THE SCREEN A SQUARE OF CUS GROWS AT THE BASE OF THE FIRST DIAGONAL LINE. IT IS FOLLOWED BY ANOTHER SQUARE, EVENTUALLY GIVING THE APPEARANCE OF A STAIRCASE. THIS PROCEDURE IS REPEATED ON THE SECOND DIAGONAL LINE AND WHEN THE LAST SQUARE IS DONE, THE ENTIRE PROCEDURE IS REVERSED. EACH SQUARE IS SUCCESSIVELY REMOVED STARTING AT THE TOP OF THE SECOND DIAGONAL LINE, CONTINUING DOWN IT, THEN STARTING AT THE TOP OF THE FIRST DIAGONAL LINE AND GOING DOWN IT. IF "COPIER" TESTING IS ENABLE, THE SCREEN WILL NOW BE COPIED.

9.10 CURSORS ON GRAPH 1

THE CORRECT VISUAL DISPLAY WILL BE ALMOST IDENTICAL TO THAT OF THE CURSORS ON GRAPH 0. THE ONLY DIFFERENCE IS THAT THE TWO DIAGONAL LINES BEGIN AT THE TOP LEFT OF THE SCREEN AND GO DOWN TOWARDS THE RIGHT. IF "COPIER" TESTING IS ENABLE, THE SCREEN WILL NOW BE COPIED.

9.11 STARTING COORDINATE ON GRAPH 0

THE CORRECT VISUAL DISPLAY WILL BEGIN WITH A SINGLE HORIZONTAL LINE EXTENDING THE ENTIRE WIDTH OF THE SCREEN PLACED AT BASE ZERO. A DIAGONAL LINE WILL THEN BEGIN TO DECAY FROM THE TOP LEFT CORNER AS THE HORIZONTAL LINE DISAPPEARS. IT CONTINUES TO DECAY UNTIL IT REACHES THE BOTTOM OF THE SCREEN WHEN A SECOND DIAGONAL LINE BEGINS TO DECAY FROM THE TOP OF THE MIDDLE OF THE SCREEN. THIS LINE CONTINUES TO DECAY AS THE HORIZONTAL LINE CONTINUES TO DISAPPEAR. WHEN THE SECOND DIAGONAL LINE REACHES THE BOTTOM OF THE SCREEN, A SMALL SECTION OF THE HORIZONTAL LINE SHOULD STILL BE VISIBLE. AT THIS POINT A DOTTED SINE CURVE BEGINS TO APPEAR FROM THE RIGHT EDGE. AS IT GROWS UPWARD THE DIAGONAL LINE IS REMOVED. AS THE SINE CURVE REACHES ITS PEAK, THE SECOND DIAGONAL LINE BEGINS TO DISAPPEAR. THE LINE CONTINUES TO DISAPPEAR AS THE SINE CURVE ROUNDS THE PEAK AND STARTS DOWNWARD. THEN THE SINE CURVE GROWS UPWARD AGAIN AND THE FIRST DIAGONAL LINE BEGINS TO DISAPPEAR AS THE CURVE REACHES ITS PEAK. THE SINE CURVE CONTINUES TO GROW UNTIL FOUR COMPLETE CYCLES HAVE BEEN FORMED. IF "COPIER" TESTING IS ENABLE, THE SCREEN WILL NOW BE COPIED.

9.12 STARTING COORDINATE ON GRAPH 1

SAME AS 9.11 EXCEPT GRAPH 1 IS ENABLED.

9.13 VT55 ADJUSTMENT PATTERN

THE CORRECT VISUAL DISPLAY CONSISTS OF TWELVE ROWS OF THE LETTER "H" WITH A BLANK LINE SEPARATING EACH OF THE ROWS. THEN TWELVE HORIZONTAL LINES ARE DISPLAYED, STARTING AT THE BOTTOM AND OVERLAYING THE ROWS OF H'S TOUCHING THE BOTTOM OF THE H'S. THEN FORTY-ONE VERTICAL LINES ARE DISPLAYED, RESULTING IN A CHECKERBOARD OVER THE TWELVE ROWS OF H'S. THE FIRST VERTICAL LINE IS PLACED THROUGH THE MIDDLE OF THE FIRST ROW OF H'S. EACH SUCCESSIVE VERTICAL LINE IS 13 POINTS TO THE RIGHT OF THE PREVIOUS LINE. THE VERY LAST LINE IS EXACTLY ON TOP OF THE RIGHT SIDE OF THE LAST ROW OF H'S. IF "COPIER" TESTING IS ENABLE, THE SCREEN WILL NOW BE COPIED.

```
%
.TITLE MAINDEC-11-DZVTD-B
*COPYRIGHT (C) 1976
*DIGITAL EQUIPMENT CORP.
*MAYNARD, MASS. 01754
*
*PROGRAM BY RAYMOND SHOOP
*
*THIS PROGRAM WAS ASSEMBLED USING THE POP-11 MAINDEC SYSMAC
*PACKAGE (MAINDEC-11-DZQAC-CO), MAR 21, 1976.
*
```

.SBTTL BASIC DEFINITIONS

;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***

STACK= 1100

.EQUIV EMT,ERROR

;;BASIC DEFINITION OF ERROR CALL

.EQUIV IOT,SCOPE

;;BASIC DEFINITION OF SCOPE CALL

;*MISCELLANEOUS DEFINITIONS

HT= 11

;;CODE FOR HORIZONTAL TAB

LF= 12

;;CODE FOR LINE FEED

CR= 15

;;CODE FOR CARRIAGE RETURN

CRLF= 200

;;CODE FOR CARRIAGE RETURN-LINE FEED

PS= 177776

;;PROCESSOR STATUS WORD

.EQUIV PS,PSW

STKLMT= 177774

;;STACK LIMIT REGISTER

PIRQ= 177772

;;PROGRAM INTERRUPT REQUEST REGISTER

DSWR= 177570

;;HARDWARE SWITCH REGISTER

DDISP= 177570

;;HARDWARE DISPLAY REGISTER

;*GENERAL PURPOSE REGISTER DEFINITIONS

R0= %0

;;GENERAL REGISTER

R1= %1

;;GENERAL REGISTER

R2= %2

;;GENERAL REGISTER

R3= %3

;;GENERAL REGISTER

R4= %4

;;GENERAL REGISTER

R5= %5

;;GENERAL REGISTER

R6= %6

;;GENERAL REGISTER

R7= %7

;;GENERAL REGISTER

.EQUIV R6,SP

;;STACK POINTER

.EQUIV R7,PC

;;PROGRAM COUNTER

;*PRIORITY LEVEL DEFINITIONS

PR0= 0

;;PRIORITY LEVEL 0

PR1= 40

;;PRIORITY LEVEL 1

PR2= 100

;;PRIORITY LEVEL 2

PR3= 140

;;PRIORITY LEVEL 3

PR4= 200

;;PRIORITY LEVEL 4

PR5= 240

;;PRIORITY LEVEL 5

PR6= 300

;;PRIORITY LEVEL 6

PR7= 340

;;PRIORITY LEVEL 7

;*"SWITCH REGISTER" SWITCH DEFINITIONS

SW15= 100000

SW14= 40000

SW13= 20000

SW12= 10000

SW11= 4000

SW10= 2000

SW09= 1000

SW08= 400

SW07= 200

SW06= 100

SW05= 40

SW04= 20

SW03= 10

SW02= 4

001100

000011

000012

000015

000200

177776

177774

177772

177570

177570

000000

000001

000002

000003

000004

000005

000006

000007

000000

000040

000100

000140

000200

000240

000300

000340

100000

040000

020000

010000

004000

002000

001000

000400

000200

000100

000040

000020

000010

000004

```

483      000002      SW01= 2
484      000001      SW00= 1
485      .EQUIV      SW09,SW9
486      .EQUIV      SW08,SW8
487      .EQUIV      SW07,SW7
488      .EQUIV      SW06,SW6
489      .EQUIV      SW05,SW5
490      .EQUIV      SW04,SW4
491      .EQUIV      SW03,SW3
492      .EQUIV      SW02,SW2
493      .EQUIV      SW01,SW1
494      .EQUIV      SW00,SW0
495
496      .*DATA BIT DEFINITIONS (BIT00 TO BIT15)
497      100000      BIT15= 100000
498      040000      BIT14= 40000
499      020000      BIT13= 20000
500      010000      BIT12= 10000
501      004000      BIT11= 4000
502      002000      BIT10= 2000
503      001000      BIT09= 1000
504      000400      BIT08= 400
505      000200      BIT07= 200
506      000100      BIT06= 100
507      000040      BIT05= 40
508      000020      BIT04= 20
509      000010      BIT03= 10
510      000004      BIT02= 4
511      000002      BIT01= 2
512      000001      BIT00= 1
513      .EQUIV      BIT09,BIT9
514      .EQUIV      BIT08,BIT8
515      .EQUIV      BIT07,BIT7
516      .EQUIV      BIT06,BIT6
517      .EQUIV      BIT05,BIT5
518      .EQUIV      BIT04,BIT4
519      .EQUIV      BIT03,BIT3
520      .EQUIV      BIT02,BIT2
521      .EQUIV      BIT01,BIT1
522      .EQUIV      BIT00,BIT0
523
524      .*BASIC "CPU" TRAP VECTOR ADDRESSES
525      000004      ERRVEC= 4      TIME OUT AND OTHER ERRORS
526      000010      RESVEC= 10     RESERVED AND ILLEGAL INSTRUCTIONS
527      000014      TBITVEC= 14    "T" BIT
528      000014      TRTVEC= 14     TRACE TRAP
529      000014      BPTVEC= 14     BREAKPOINT TRAP (BPT)
530      000020      IOTVEC= 20     INPUT/OUTPUT TRAP (IOT) **SCOPE**
531      000024      PWRVEC= 24     POWER FAIL
532      000030      EMTVEC= 30     EMULATOR TRAP (EMT) **ERROR**
533      000034      TRAPVEC= 34    "TRAP" TRAP
534      000060      TKVEC= 60      TTY KEYBOARD VECTOR
535      000064      TPVEC= 64      TTY PRINTER VECTOR
536      000240      PIRQVEC= 240   PROGRAM INTERRUPT REQUEST VECTOR
537      176500      ABASE=176500

```

```

538 .SBTTL OPERATIONAL SWITCH SETTINGS
539
540 SWITCH USE
541 -----
542 15 HALT ON ERROR
543 14 LOOP ON TEST
544 13 INHIBIT ERROR TYPEOUTS
545 12 INHIBIT SUB-TEST DELAY'S
546 10 ENABLE "SAVE COPIER PAPER MODE"
547 9 LOOP ON ERROR
548 8 LOOP ON TEST IN SWR(7:0)
549 .SBTTL TRAP CATCHER
550
551 .=0
552 ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
553 ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
554 ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
555 .=174
556 000174 000000 DISPREG: .WORD 0 ;:SOFTWARE DISPLAY REGISTER
557 000176 000000 SWREG: .WORD 0 ;:SOFTWARE SWITCH REGISTER
558 .SBTTL STARTING ADDRESS(ES)
559 000200 000137 001346 JMP 2*BEGIN ;:JUMP TO STARTING ADDRESS OF PROGRAM
560 000204 000137 001402 JMP RBEGIN ;:JUMP TO RESTART ADDRESS
561 000210 000137 001412 JMP BEGIN2 ;:JUMP TO ADJUSTMENT PATTERN
562
563 .SBTTL ACT11 HOOKS
564
565 ;*****
566 ;HOOKS REQUIRED BY ACT11
567
568 000214 $SVPC=. ;SAVE PC
569 000046 .=46
570 000046 006776 SENDAD ;:1)SET LOC.46 TO ADDRESS OF SENDAD IN .SEOP
571 000052 000052 .=52
572 000052 000000 .WORD 0 ;:2)SET LOC.52 TO ZERO
573 000214 .=$SVPC ;: RSTORE PC
574 001000 .=1000
575 .SBTTL APT PARAMETER BLOCK
576
577 ;*****
578 ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
579 ;*****
580 001000 .$X=. ;:SAVE CURRENT LOCATION
581 000024 .=24 ;:SET POWER FAIL TO POINT TO START OF PROGRAM
582 000024 000200 200 ;:FOR APT START UP
583 000044 .=44 ;:POINT TO APT INDIRECT ADDRESS PNTR.
584 000044 001000 $APTHDR ;:POINT TO APT HEADER BLOCK
585 001000 .=$X ;:RESET LOCATION COUNTER
586 ;*****
587 ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-POPI1 DIAGNOSTIC
588 ;INTERFACE SPEC.
589
590 001000 $APTH0:
591 001000 000000 $HIBTS: .WORD 0 ;:TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
592 001002 001174 $MBADR: .WORD $MAIL ;:ADDRESS OF APT MAILBOX (BITS 0-15)
593 001004 000300 $TSTM: .WORD 300 ;:RUN TIM OF LONGEST TEST

```

802

MAINDEC-11-DZVTD-B MACY11 27(732) 25-SEP-76 10:10 PAGE 15
DZVTDB.SAC APT PARAMETER BLOCK

594 001006 000300
595 001010 000300
596 001012 000031

SPASTM: .WORD 300 ;; RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
SUNITM: .WORD 300 ;; ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
.WORD SETEND-SMIL/2 ;; LENGTH MAILBOX-ETABLE(WORDS)

```

:*****
:THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
:USED IN THE PROGRAM.

```

;;START OF COMMON TAGS

```

: CONTAINS THE TEST NUMBER
: CONTAINS ERROR FLAG
: CONTAINS SUREST ITERATION COUNT
: CONTAINS SCOPE LOOP ADDRESS
: CONTAINS SCOPE RETURN FOR ERRORS
: CONTAINS TOTAL ERRORS DETECTED
: CONTAINS ITEM CONTRL. BYTE
: CONTAINS MAX. ERRORS PER TEST
: CONTAINS PC OF LAST ERROR INSTRUCTION
: CONTAINS ADDRESS OF 'GOOD' DATA
: CONTAINS ADDRESS OF 'BAD' DATA
: CONTAINS 'GOOD' DATA
: CONTAINS 'BAD' DATA
: RESERVED--NOT TO BE USED

: AUTOMATIC MODE INDICATOR
: INTERRUPT MODE INDICATOR

: ADDRESS OF SWITCH REGISTER
: ADDRESS OF DISPLAY REGISTER
: TTY KBD STATUS
: TTY KBD BUFFER
: TTY PRINTER STATUS REG. ADDRESS
: TTY PRINTER BUFFER REG. ADDRESS
: CONTAINS NULL CHARACTER FOR FILLS
: CONTAINS # OF FILLER CHARACTERS REQUIRED
: INSERT FILL CHARS. AFTER A "LINE FEED"
: "TERMINAL AVAILABLE" FLAG (BIT(07)=0=YES)
: CONTAINS THE ADDRESS FROM
: WHICH ($REG0) WAS OBTAINED
: CONTAINS (($REG0)+0)
: CONTAINS (($REG0)+2)
: ESCAPE ON ERROR ADDRESS
: QUESTION MARK
: CARRIAGE RETURN
: LINE FEED

```

SMTL APT MAILBOX-ETABLE

```

;*****
;EVEN
;MAIL;
;MSGTY: .WORD      MSGTY      ;;APT MAILBOX
;FATAL: .WORD      AFATAL     ;;MESSAGE TYPE CODE
;TESTN: .WORD      ATESTN    ;;FATAL ERROR NUMBER
;PASS:  .WORD      APASS     ;;TEST NUMBER
;DEVCT: .WORD      ADEVCT    ;;PASS COUNT
;                               ;;DEVICE COUNT

```

D02

MAINDEC-11-DZVTD-B MACY11 27(732) 25-SEP-76 10:10 PAGE 17
 DZVTD8.SRC APT MAILBOX-ETABLE

653	001206	000000	SUNIT: .WORD	AUNIT	:: I/O UNIT NUMBER
654	001210	000000	MSGAD: .WORD	AMSGAD	:: MESSAGE ADDRESS
655	001212	000000	MSGLG: .WORD	AMSGLG	:: MESSAGE LENGTH
656	001214		SETABLE:		:: APT ENVIRONMENT TABLE
657	001214	000	SENV: .BYTE	RENV	:: ENVIRONMENT BYTE
658	001215	000	SENVH: .BYTE	RENVH	:: ENVIRONMENT MODE BITS
659	001216	000000	SWREG: .WORD	ASWREG	:: APT SWITCH REGISTER
660	001220	000000	USUR: .WORD	AUSUR	:: USER SWITCHES
661	001222	000000	SCPUOP: .WORD	ACPUOP	:: CPU TYPE, OPTIONS
662			::		BITS 15-11=CPU TYPE
663			::		11/04=01, 11/05=02, 11/21=03, 11/40=04, 11/45=05
664			::		11/70=06, P00=07, 0=10
665			::		BIT 10=REAL TIME CLOCK
666			::		BIT 9=FLOATING POINT PROCESSOR
667			::		BIT 8=MEMORY MANAGEMENT
668	001224	000	SHMS1: .BYTE	AHMS1	:: HIGH ADDRESS, H.S. BYTE
669	001225	000	SHYP1: .BYTE	ANTYP1	:: MEM. TYPE, BLK#1
670			::		MEM. TYPE BYTE -- (HIGH BYTE)
671			::		900 NSEC CORE=001
672			::		300 NSEC BIPOLAR=002
673			::		500 NSEC MOS=003
674	001226	000000	SHADR1: .WORD	AHADR1	:: HIGH ADDRESS, BLK#1
675			::		MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
676	001230	000	SHMS2: .BYTE	AHMS2	:: HIGH ADDRESS, H.S. BYTE
677	001231	000	SHYP2: .BYTE	ANTYP2	:: MEM. TYPE, BLK#2
678	001232	000000	SHADR2: .WORD	AHADR2	:: MEM. LAST ADDRESS, BLK#2
679	001233	000	SHMS3: .BYTE	AHMS3	:: HIGH ADDRESS, H.S. BYTE
680	001234	000	SHYP3: .BYTE	ANTYP3	:: MEM. TYPE, BLK#3
681	001235	000000	SHADR3: .WORD	AHADR3	:: MEM. LAST ADDRESS, BLK#3
682	001236	000	SHMS4: .BYTE	AHMS4	:: HIGH ADDRESS, H.S. BYTE
683	001237	000	SHYP4: .BYTE	ANTYP4	:: MEM. TYPE, BLK#4
684	001238	000000	SHADR4: .WORD	AHADR4	:: MEM. LAST ADDRESS, BLK#4
685	001239	000000	SVECT1: .WORD	AVECT1	:: INTERRUPT VECTOR#1, BUS PRIORITY#1
686	001240	000000	SVECT2: .WORD	AVECT2	:: INTERRUPT VECTOR#2, BUS PRIORITY#2
687	001241	176500	BASE: .WORD	ABASE	:: BASE ADDRESS OF EQUIPMENT UNDER TEST
688	001242	000000	DEVH: .WORD	ADEVH	:: DEVICE MAP
689	001243	000000	SCON1: .WORD	ACON1	:: CONTROLLER DESCRIPTION WORD#1
690	001244	000000	SETEND:		
691			.NEXT		

.SBTTL ERROR POINTER TABLE

; THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
 ; THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
 ; LOCATION SITE#B. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
 ; NOTE1: IF SITE#B IS 0 THE ONLY PERTINENT DATA IS (SERAPC).
 ; NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 PCINTERS 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

SERRTB:

;	ITEM	1		
		EM1	;	INCORRECT OR UNEXPECTED INPUT CHARACTER
		DH1	;	ERRPC VTNOW TSTNUM EXPT RCVD
		DT1	;	SERRPC VTNOW TSTNUM SCDOAT SBDOAT
		0		

;	ITEM	2		
		EM2	;	NO HOST INPUT FLAG RECEIVED
		DH1	;	ERRPC VTNOW TSTNUM
		DT1	;	SERRPC VTNOW TSTNUM
		0		

;	ITEM	3		
		EM3	;	INVALID BUS ADDRESS, TRY AGAIN
		0		
		0		
		0		

692
 693
 694
 695
 696
 697
 698
 699
 700
 701
 702
 703
 704
 705
 706 001256
 707
 708
 709
 710 001256 011612
 711 001260 011745
 712 001262 012700
 713 001264 000000
 714
 715
 716 001266 011563
 717 001270 011714
 718 001272 012670
 719 001274 000000
 720
 721
 722 001276 011655
 723 001300 000000
 724 001302 000000
 725 001304 000000
 726

;VT-55 EQUALITIES

727			
728			
729	000354	MAXH0Z=236.	;MAX. HORIZ LINE COUNT
730	000400	MAXH7C=256.	
731	001000	MAXVRT=512.	;MAX. VERTICAL LINE COUNT
732			
733	002000	ADDLIN=BIT10	
734			
735	000033	ESC= 33	
736			
737	000100	LNO= 100	;NOP
738			
739	000101	LDE0= 101	;LOAD ENABLE REG. 0
740	000111	LDE1= 111	;LOAD ENABLE REG. 1
741			
742	000102	LDG0= 102	;LOAD GRAPH 0
743	000112	LDG1= 112	;LOAD GRAPH 1
744			
745	000103	LDC0= 103	;LOAD CURSOR ON GRAPH 0
746	000113	LDC1= 113	;LOAD CURSOR ON GRAPH 1
747			
748	000104	LHV0= 104	;LOAD HORIZONTAL LINE
749	000114	LHV1= 114	;LOAD VERTICAL LINE
750			
751	000110	LSC= 110	;LOAD STARTING COORDINATE
752			
753	000135	COPY= 135	;COPY SCREEN COMMAND

754											
755	001306	177560			FIRST:	177560					; FIRST DEVICE ADDRESS OF SEQUENTIAL DL-11-A/B TYPE DEVIC
756											; DEFAULT TO THE CONSOLE ADDRESS
757	001310	000000			LAST:	0					; LAST DEVICE ADDRESS OF DL-11-A/B TYPE
758	001312	177560			VTNOW:	177560					; CURRENT DEVICE BUSS ADDRESS
759	001314	000000			TSTNUM:	0					; ERROR PATTERN
760											
761	001316	000100			TIMED:	100					; CHARACTER FLAG TIMEOUT CONSTANT
762	001320	000012			SUBTST:	10.					; SUBTEST DELAY CONSTANT
763	001322	000000			MFTEST:	0					
764	001324	177560			VTIS:	177560					; DEVICE ADDRESSES
765	001326	177562			VTIB:	177562					; IN DATA
766	001330	177564			VTOS:	177564					; OUT STAT
767	001332	177566			VT08:	177566					; OUT DATA
768	001334	000000			SAVE4:	0					
769											
770	001336	022626			BUSSTR:	CMP	(SP)+, (SP)+				; POP STACK
771	001340	104003				ERROR	3				; INVALID BUS ADDRESS
772	001342	000240				NOP					
773	001344	000240				NOP					
774											
775	001346	012737	002064	002062	BEGIN:	MOV	#TST1, WHERE				; STARTING ACCEPTANCE TEST ADDRESS
776	001354	005037	001322			CLR	MFTEST				
777	001360	005037	001334			CLR	SAVE4				
778	001364	012737	001336	000004		MOV	#BUSSTR, #ERRVEC				
779	001372	012737	000340	000006		MOV	#340, #ERRVEC+2				
780	001400	000412				BR	GINA				
781	001402	012737	002064	002062	RBEGIN:	MOV	#TST1, WHERE				
782	001410	000403				BR	GIN				
783	001412	012737	000001	001322	BEGIN2:	MOV	#1, MFTEST				
784	001420	012737	000001	001334	GIN:	MOV	#1, SAVE4				
785	001426	000005			GINA:	RESET					

```

786 .SBTTL INITIALIZE THE COMMON TAGS
787 ;; CLEAR THE COMMON TAGS (SCHTAG) AREA
788 MOV #SCHTAG,R6 ;; FIRST LOCATION TO BE CLEARED
789 CLR (R6)+ ;; CLEAR MEMORY LOCATION
790 CMP #SWR,R6 ;; DONE?
791 BNE -6 ;; LOOP BACK IF NO
792 MOV #STACK,SP ;; SETUP THE STACK POINTER
793 ;; INITIALIZE A FEW VECTORS
794 MOV #SSCOPE, @IOTVEC ;; IOT VECTOR FOR SCOPE ROUTINE
795 MOV #340, @IOTVEC+2 ;; LEVEL 7
796 MOV #ERROR, @ENTVEC ;; ENT VECTOR FOR ERROR ROUTINE
797 MOV #340, @ENTVEC+2 ;; LEVEL 7
798 MOV #TRAP, @TRAPVEC ;; TRAP VECTOR FOR TRAP CALLS
799 MOV #340, @TRAPVEC+2 ;; LEVEL 7
800 MOV #SPWRON, @PWRVEC ;; POWER FAILURE VECTOR
801 MOV #340, @PWRVEC+2 ;; LEVEL 7
802 MOV #NOCT, @LOPCT ;; SETUP END-OF-PROGRAM COUNTER
803 CLR #ESCAPE ;; CLEAR THE ESCAPE ON ERROR ADDRESS
804 MOVB #1, #SERMAX ;; ALLOW ONE ERROR PER TEST
805 MOV #., #SLFADR ;; INITIALIZE THE LOOP ADDRESS FOR SCOPE
806 MOV #., #SLPERR ;; SETUP THE ERROR LOOP ADDRESS
807 ;; SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
808 ;; EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
809 MOV @ERRVEC, -(SP) ;; SAVE ERROR VECTOR
810 MOV #645, @ERRVEC ;; SET UP ERROR VECTOR
811 MOV #DSWR, SWR ;; SETUP FOR A HARDWARE SWITCH REGISTER
812 MOV #DISP, DISPLAY ;; AND A HARDWARE DISPLAY REGISTER
813 CMP #1, @SWR ;; TRY TO REFERENCE HARDWARE SWR
814 BNE 66$ ;; BRANCH IF NO TIMEOUT TRAP OCCURRED
815 ;; AND THE HARDWARE SWR IS NOT = -1
816 BR 65$ ;; BRANCH IF NO TIMEOUT
817 MOV #65$, (SP) ;; SET UP FOR TRAP RETURN
818 RTI
819 MOV #SWREG, SWR ;; POINT TO SOFTWARE SWR
820 MOV #DISPREG, DISPLAY
821 MOV (SP)+, @ERRVEC ;; RESTORE ERROR VECTOR
822
823 CLR #PASS ;; CLEAR PASS COUNT
824 BITB #APTSIZE, #ENVH ;; TEST USER SIZE UNDER APT
825 BEQ 67$ ;; YES, USE NON-APT SWITCH
826 MOV #SSWREG, SWR ;; NO, USE APT SWITCH REGISTER
827
828 MOV #BASE, FIRST ;; LOAD "APT" 'S BASE ADDRESS
829 CLR LOOP
830 MOV #SSCOPE, @IOTVEC
831 TST SAVE4 ;; TEST FLAG
832 BNE 11$ ;; BR IF NON-ZERO
833 TYPE
834 TITLE

```

MAINDEC-11-DZVTD-B MACY11 27(732) 25-SEP-76 10:10 PAGE 22
 DZVTD8.SRC INITIALIZE THE COMMON TAGS

```

835
836
837 001726 022737 000176 001140 ;NOW TEST IF RUNNING ON NON-SWR CPU
838 001734 001403 115: CMP #SWREG,SWR ;TEST IF NON-SWITCH REGISTER
839 001736 105777 177176 BEQ 25 ;BR IF YES - AND ASK ADDRESS
840 001742 100022 TS18 #SWR ;BR IF CLEARED
841 001744 104400 BPL RSTRTA ;FIND OUT THE DEVICE ADDRESS
842 001746 011466 25: TYPE ;FIND OUT THE DEVICE ADDRESS
843 001750 104410 MHATO ;FIND OUT THE DEVICE ADDRESS
844 001752 012637 001306 RDOCT ;FIND OUT THE DEVICE ADDRESS
845 001756 022737 160000 001306 MOV (SP)+,FIRST ;SAVE THE ADDRESS
846 001764 101360 CMP #160000,FIRST ;BR IF INVALID
847 001766 005777 177314 BHI 115 ;TEST IF VALID
848 001772 005037 001310 TST #FIRST
849 001776 012737 000006 000004 CLR LAST
850 002004 005037 000006 000004 MOV #6,#4
851 002010 013737 001306 001312 RSTRTA: CLR #6
852 002016 012707 001324 RSTRT: MOV FIRST,VTNOW ;LOAD INITIAL DEVICE ADDRESS
853 002022 013720 001312 MOV #VTIS,RO ;LOAD POINTER
854 002026 013710 001312 MOV VTNOW,(RO)+ ;LOAD INPUT STAT
855 002032 062720 000002 MOV VTNOW,(RO) ;LOAD INPUT BUFFER
856 002036 013710 001312 ADD #2,(RO)+ ;LOAD INPUT BUFFER
857 002042 062720 000004 MOV VTNOW,(RO) ;LOAD OUTPUT STAT
858 002046 013710 001312 ADD #4,(RO)+ ;LOAD OUTPUT STAT
859 002052 062720 000006 MOV VTNOW,(RO) ;LOAD OUTPUT BUFFER
860 002056 000177 000000 ADD #6,(RO)+ ;JUMP TO STARTING ADDRESS
861 002062 002064 WHERE: JMP WHERE
862

```


K02

MAINDEC-11-DZVTD-8
DZVTD8.SRC T1MACY11 27(732) 25-SEP-76 10:10 PAGE 24
A DISPLAY A GROWING HORIZONTAL LINES

911							
912	002332	012700	015562		MOV	#BUFFER, R0	;LOAD BUFFER POINTER
913	002336	112720	000100		MOVB	#LNO, (R0)+	;LOAD NOP
914	002342	112720	000104		MOVB	#LHVD, (R0)+	;LOAD HORIZONTAL LINE MODE AGAIN
915							
916	002346	012737	000000	007772	MOV	#0, BASE	;LOAD STARTING DATA VALUE TO REMOVE THE LINE
917	002354	004737	007776	25:	JSR	PC, SHUFF	;SHUFFEL THE DATA INTO VT-55 FORMAT
918	002360	010120			MOV	R1, (R0)+	;SAVE THE LSB MSB BYTE
919	002362	005237	007772		INC	BASE	;UPDATE THE DATA
920	002366	022737	000354	007772	CMP	#MAXH02, BASE	;COMPUTE TO LAST DATA LINE
921	002374	001367			BNE	25	;BR TOLL DONE
922							
923	002376	112720	000033		MOVB	#ESC, (R0)+	;LOAD "ESC" CODE
924	002402	112720	000135		MOVB	#COPY, (R0)+	;LOAD COPY SCREEN
925	002406	112720	000100		MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
926	002412	112720	000100		MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
927	002416	112720	000100		MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
928	002422	112720	000100		MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
929	002426	112720	000100		MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
930	002432	112720	000100		MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
931	002436	112720	000100		MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
932	002442	112720	000100		MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
933	002446	112720	000100		MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
934	002452	112720	000100		MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
935	002456	112720	000100		MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
936	002462	112720	000100		MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
937	002466	112720	000100		MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
938	002472	112720	000100		MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
939	002476	112720	000100		MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
940	002502	112720	000100		MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
941	002506	105020			CLRB	(R0)+	;LOAD TERMINATOR
942	002510	004737	010074		JSR	PC, XPRNT	;DISPLAY
943	002514	004737	010742		JSR	PC, DELAY	
944							

Address	Hex	Dec	Label	Instruction	Comment
002520	000004	011070	IST2:	SCOPE	
002522	004537			JSR	RS,AMSG
002526	012045			DAVL	;DISPLAY HEADER
002530	012700	015562		MOV	#BUFFER,RO
002534	112720	000033		MOVB	#ESC,(RO)+
002540	112720	000061		MOVB	#61,(RO)+
002544	112720	000101		MOVB	#LDE0,(RO)+
002550	112720	000040		MOVB	#BITS,(RO)+
002554	112720	000111		MOVB	#LDE1,(RO)+
002560	112720	000060		MOVB	#BITS!BIT4,(RO)+
002564	112720	000101		MOVB	#LDE0,(RO)+
002570	112720	000041		MOVB	#BITS!BIT0,(RO)+
002574	112720	000111		MOVB	#LDE1,(RO)+
002600	112720	000042		MOVB	#BITS!BIT1,(RO)+
002604	112720	000100		MOVB	#LNO,(RO)+
002610	112720	000114		MOVB	#LHV1,(RO)+
002614	012737	002777	0G7772	MOV	#ADDLIN!MAXVRT-1,BASE
002622	004737	007776	1S:	JSR	PC,SHUFF
002626	010120			MOV	R1,(RO)+
002630	005337	007772		DEC	BASE
002634	022737	002000	007772	CMF	#ADDLIN,BASE
002642	001367			BNE	1S
002644	112720	000033		MOVB	#ESC,(RO)+
002650	112720	000135		MOVB	#COPY,(RO)+
002654	112720	000100		MOVB	#LNO,(RO)+
002660	112720	000100		MOVB	#LNO,(RO)+
002664	112720	000100		MOVB	#LNO,(RO)+
002670	112720	000100		MOVB	#LNO,(RO)+
002674	112720	000100		MOVB	#LNO,(RO)+
002700	112720	000100		MOVB	#LNO,(RO)+
002704	112720	000100		MOVB	#LNO,(RO)+
002710	112720	000100		MOVB	#LNO,(RO)+
002714	112720	000100		MOVB	#LNO,(RO)+
002720	112720	000100		MOVB	#LNO,(RO)+
002724	112720	000100		MOVB	#LNO,(RO)+
002730	112720	000100		MOVB	#LNO,(RO)+
002734	112720	000100		MOVB	#LNO,(RO)+
002740	112720	000100		MOVB	#LNO,(RO)+
002744	112720	000100		MOVB	#LNO,(RO)+
002750	112720	000100		MOVB	#LNO,(RO)+
002754	105020			CLR2	(RO)+
002756	004737	010074		JSR	PC,XPRNT
002762	004737	010742		JSR	PC,DELAY

MAINDEC-11-DZVTD-B
DZVTD8.SRC T2

MACY11 27(732) 25-SEP-76 10:10 PAGE 26
B DISPLAY A GROWING VERTICAL LINES

992	002766	012700	015562	MOV	#BUFFER,RO	
993	002772	112720	000100	MOVB	#LNO,(RO)+	;LOAD NOP
994	002775	112720	000114	MOVB	#LHV1,(RO)+	;LOAD VERTIACL LINE MODE AGAIN
995						
996	003002	012737	001000	MOV	#MAXVRT,BASE	;LOAD STARTING DATA VALUE TO REMOVE THE LINE
997	003010	004737	007776	JSR	PC,SHUFF	;SHUFFEL THE DATA INTO VT-55 FORMAT
998	003014	010120		MOV	R1,(RO)+	;SAVE THE LSB MCB BYTE
999	003016	005337	007772	DEC	BASE	;UPDATE THE DATA
1000	003022	001372		BNE	25	;BR TOLL DONE
1001						
1002	003024	112720	000333	MOVB	#ESC,(RO)+	;LOAD "ESC" CODE
1003	003030	112720	000135	MOVB	#COPY,(RO)+	;LOAD COPY SCREEN
1004	003034	112720	000100	MOVB	#LNO,(RO)+	;LOAD NOP TO LOAD SILO
1005	003040	112720	000100	MOVB	#LNO,(RO)+	;LOAD NOP TO LOAD SILO
1006	003044	112720	000100	MOVB	#LNO,(RO)+	;LOAD NOP TO LOAD SILO
1007	003050	112720	000100	MOVB	#LNO,(RO)+	;LOAD NOP TO LOAD SILO
1008	003054	112720	000100	MOVB	#LNO,(RO)+	;LOAD NOP TO LOAD SILO
1009	003060	112720	000100	MOVB	#LNO,(RO)+	;LOAD NOP TO LOAD SILO
1010	003064	112720	000100	MOVB	#LNO,(RO)+	;LOAD NOP TO LOAD SILO
1011	003070	112720	000100	MOVB	#LNO,(RO)+	;LOAD NOP TO LOAD SILO
1012	003074	112720	000100	MOVB	#LNO,(RO)+	;LOAD NOP TO LOAD SILO
1013	003100	112720	000100	MOVB	#LNO,(RO)+	;LOAD NOP TO LOAD SILO
1014	003104	112720	000100	MOVB	#LNO,(RO)+	;LOAD NOP TO LOAD SILO
1015	003110	112720	000100	MOVB	#LNO,(RO)+	;LOAD NOP TO LOAD SILO
1016	003114	112720	000100	MOVB	#LNO,(RO)+	;LOAD NOP TO LOAD SILO
1017	003120	112720	000100	MOVB	#LNO,(RO)+	;LOAD NOP TO LOAD SILO
1018	003124	112720	000100	MOVB	#LNO,(RO)+	;LOAD NOP TO LOAD SILO
1019	003130	112720	000100	MOVB	#LNO,(RO)+	;LOAD NOP TO LOAD SILO
1020	003134	105020		CLRB	(RO)+	;LOAD TERMINATOR
1021	003136	004737	010074	JSR	PC,XPRNT	;DISPLAY
1022	003142	004737	010742	JSR	PC,DELAY	
1023						

N02

MAINDEC-11-DZVTD-B
DZVTD8.SRC T3

MACY11 27(732) 25-SEP-76 10:10 PAGE 27
C GRAPH 0 DISPLAY A STEPPING HORIZONTAL LINE

```

1024                                     ;*****
1025                                     ;*TEST 3          C      GRAPH 0 DISPLAY A STEPPING HORIZONTAL LINE
1026                                     ;*****
1027 003146 000004                      TST3: SCOPE
1028 003150 004537 011070                JSR      RS,AMSG                      ;DISPLAY HEADER
1029 003154 012076                      SHLO
1030 003156 012700 015562                MOV      #BUFFER,RO                      ;LOAD OUTPUT BUFFER POINTER
1031 003162 112720 000033                MOV      #ESC,(RO)+                      ;ENTER VT-55 MODE
1032 003166 112720 000061                MOV      #61,(RO)+
1033 003172 112720 000101                MOV      #LDE0,(RO)+                      ;LOAD ENABLE 0
1034 003176 112720 000040                MOV      #BITS,(RO)+                      ;DISABLE DISPLAY
1035 003202 112720 000111                MOV      #LDE1,(RO)+                      ;LOAD ENABLE 1
1036 003206 112720 000060                MOV      #BITS!BIT4,(RO)+          ;CLEAR GRAPH, LILES, AND CURSORS
1037 003212 105020                      CLR      (RO)+                      ;LOAD TERMINATOR
1038 003214 004737 010074                JSR      PC,XPRNT                      ;EXECUTE
1039
1040 003220 012737 000200 003400          MOV      #BIT7,100$                      ;LOAD STARTING BASE LINE
1041 003226 012700 015562 2$:            MOV      #BUFFER,RO                      ;LOAD OUTPUT BUFFER POINTER
1042 003232 112720 000033                MOV      #ESC,(RO)+                      ;ENTER VT-55 FORMAT
1043 003236 112720 000061                MOV      #61,(RO)+                      ;LOAD ENTER '01' CODE
1044 003242 112720 000101                MOV      #LDE0,(RO)+                      ;LOAD ENABLE 0
1045 003246 112720 000040                MOV      #BITS,(RO)+                      ;DISABLE DISPLAY
1046 003252 112720 000111                MOV      #LDE1,(RO)+                      ;LOAD ENABLE 1
1047 003256 112720 000060                MOV      #BITS!BIT4,(RO)+          ;CLEAR GRAPH, LINES AND CURSORS
1048
1049 003262 112720 000101                MOV      #LDE0,(RO)+                      ;LOAD ENABLE 0
1050 003266 112720 000043                MOV      #BITS!BIT1!BIT0,(RO)+        ;LOAD DISPLAY ENABLE AND GRAPH 0 ON
1051 003272 112720 000100                MOV      #LNO,(RO)+                      ;NOP
1052 003276 112720 000110                MOV      #LSC,(RO)+                      ;LOAD STARTING COORD.
1053 003302 012737 000000 007772          MOV      #0,BASE                      ;GET BASE LINE
1054 003310 004737 007776                JSR      PC,SHUFF                      ;CONVERT
1055 003314 010120                      MOV      R1,(RO)+                      ;SAVE COORD.
1056 003316 112720 000100                MOV      #LNO,(RO)+                      ;LOAD NOP
1057 003322 112720 000102                MOV      #LDG0,(RO)+                      ;LOAD "LOAD GRAPH"
1058 003326 013737 003400 007772          MOV      100$,BASE                      ;LOAD THE STARTING DATA VALUE
1059 003334 004737 007776                JSR      PC,SHUFF                      ;SHUFFEL THE DATA INTO VT-55 FORMAT
1060 003340 012737 001000 003402          MOV      #MAXVRT,101$                  ;LOAD COUNTER
1061 003346 010120 1$:                    MOV      R1,(RO)+                      ;SAVE THE LSB MSB BYTE
1062 003350 005337 003402                DEC      101$                          ;DONE FULL GRAPH
1063 003354 001374                      BNE      1$
1064
1065
1066 003356 105020                      CLR      (RO)+                      ;LOAD TERMINATOR
1067 003360 004737 010074                JSR      PC,XPRNT                      ;DISPLAY
1068 003364 006037 003400                ROR      100$                          ;CHANGE DATA VALUE
1069 003370 001316                      BNE      2$                          ;NO
1070 003372 004737 010742                JSR      PC,DELAY
1071 003376 000402                      BR       TST4                      ;;NEXT TEST
1072
1073 003400 000000                      100$: 0
1074 003402 000000                      101$: 0

```

```

1075      ::*****
1076      ;TEST 4      D      GRAPH 1 DISPLAY A STEPPING HORIZONTAL LINE
1077      ;*****
1078      ;ST4:  SCOPE
1079      JSR      RS,AMSG      ;DISPLAY HEADER
1080      SHL 1
1081      MOV      #BUFFER,RO      ;LOAD OUTPUT BUFFER POINTER
1082      MOV      #ESC,(RO)+      ;ENTER VT-55 MODE
1083      MOV      #61,(RO)+
1084      MOV      #LDE0,(RO)+      ;LOAD ENABLE 0
1085      MOV      #BITS,(RO)+      ;DISABLE DISPLAY
1086      MOV      #LDE1,(RO)+      ;LOAD ENABLE 1
1087      MOV      #BITS!BIT4,(RO)+      ;CLEAR GRAPH, LINES, AND CURSORS
1088      CLRB      (RO)+      ;LOAD TERMINATOR
1089      JSR      PC,XPRNT      ;EXECUTE
1090
1091      003456 012737 000200 003636      MOV      #BIT7,100$      ;LOAD STARTING BASE LINE
1092      003464 012700 015562 2$:      MOV      #BUFFER,RO      ;LOAD OUTPUT BUFFER POINTER
1093      003470 112720 000033      MOV      #ESC,(RO)+      ;ENTER VT55 MODE
1094      003474 112720 000061      MOV      #61,(RO)+      ;LOAD ENTER "01" CODE
1095      003500 112720 000101      MOV      #LDE0,(RO)+      ;LOAD ENABLE 0
1096      003504 112720 000040      MOV      #BITS,(RO)+      ;DISABLE DISPLAY
1097      003510 112720 000111      MOV      #LDE1,(RO)+      ;LOAD ENABLE 1
1098      003514 112720 000060      MOV      #BITS!BIT4,(RO)+      ;CLEAR GRAPH,LINES AND CURSORS
1099      003520 112720 000101      MOV      #LDE0,(RO)+      ;LOAD ENABLE 0
1100      003524 112720 000045      MOV      #BITS!BIT2!BIT0,(RO)+      ;LOAD DISPLAY ENABLE AND GRAPH 1 ON
1101      003530 112720 000100      MOV      #LNO,(RO)+      ;NOP
1102      003534 112720 000110      MOV      #LSC,(RO)+      ;LOAD STARTING COORD.
1103      003540 012737 000000 007772      MOV      #0,BASE      ;GET BASE LINE
1104      003546 004737 007776      JSR      PC,SHUFF      ;CONVERT
1105      003552 010120      MOV      R1,(RO)+      ;SAVE COORD.
1106      003554 112720 000100      MOV      #LNO,(RO)+      ;LOAD NOP
1107      003560 112720 000112      MOV      #LDG1,(RO)+      ;LOAD "LOAD GRAPH"
1108      003564 013737 003636 007772      MOV      #100$,BASE      ;LOAD THE STARTING DATA VALUE
1109      003572 004737 007776      JSR      PC,SHUFF      ;SHUFFLE THE DATA INTO VT-55 FORMAT
1110      003576 012737 001000 003640      MOV      #MAXVRT,101$      ;LOAD COUNTER
1111      003604 010120 1$:      MOV      R1,(RO)+      ;SAVE THE LSB MSB BYTE
1112      003606 005337 003640      DEC      101$      ;DONE FULL GRAPH
1113      003612 001374
1114
1115
1116      003614 105020      CLRB      (RO)+      ;LOAD TERMINATOR
1117      003616 004737 010074      JSR      PC,XPRNT      ;DISPLAY
1118      003622 006037 003636      ROR      100$
1119      003626 001316      BNE      2$      ;NO
1120      003630 004737 010742      JSR      PC,DELAY
1121      003634 000402      BR      TS1$      ;;NEXT TEST
1122
1123      003636 000000      100$: 0
1124      003640 000000      101$: 0

```

```

1125      ;*****
1126      ;TEST 5      E      GRAPH 0 AND 1
1127      ;*****
1128      TST5:  SCOPE
1129      JSR      RS,AMSG      ;DISPLAY HEADER
1130      GORAI
1131
1132      MOV      #BUFFER,RO      ;LOAD OUTPUT BUFFER POINTER
1133      MOV      #ESC,(RO)+      ;ENTER VTSS MODE
1134      MOV      #61,(RO)+      ;ENTER "01" CODE
1135      MOV      #LDEL,(RO)+      ;LOAD ENABLE 0
1136      MOV      #BITS,(RO)+      ;DISABLE DISPLAY
1137      MOV      #LDEL,(RO)+      ;LOAD ENABLE 1
1138      MOV      #BITS!BIT4,(RO)+ ;CLEAR GRAPH LINES AND CURSORS
1139      CLRB      (RO)+      ;LOAD TERMINATOR
1140      JSR      PC,XPRNT      ;EXECUTE
1141
1142      JSR      RS,UPDOWN      ;LOAD DATA PATTERN
1143      .BYTE      BIT1,LDG0      ;GRAPH 0 INC. PAT.
1144      .WORD      0
1145      JSR      PC,XPRNT
1146
1147      JSR      RS,UPDOWN      ;LOAD DATA PATTERN
1148      .BYTE      BIT2:BIT1,LDG1 ;GRAPH 0 DEC. PAT.
1149      .WORD      BIT15:MAXH02
1150      ;
1151      MOV      #ESC,(RO)+      ;LOAD "ESC" CODE
1152      MOV      #COPY,(RO)+      ;LOAD COPY SCREEN
1153      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1154      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1155      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1156      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1157      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1158      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1159      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1160      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1161      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1162      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1163      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1164      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1165      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1166      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1167      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1168      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1169      CLRB      (RO)+      ;LOAD TERMINATOR
1170      JSR      PC,XPRNT      ;EXECUTE
1171      JSR      PC,DELAY

```

```

1173 .....
1174 .....
1175 .....
1176 004062 000004 011070
1177 004064 004537 012276
1178 004070 012700 015562
1179 004076 112 20 000033
1180 004102 112720 000061
1181 004106 112720 000101
1182 004112 112720 000040
1183 004116 112720 000111
1184 004122 112720 000060
1185 004126 105020
1186 004130 004737 010074
1187 .....
1188 004134 012737 000200 004274
1189 .....
1190 004142 012700 015562
1191 004146 112720 000033
1192 004152 112720 000061
1193 004156 112720 000101
1194 004162 112720 000053
1195 004166 112720 000100
1196 004172 112720 000110
1197 004176 012737 000000 007772
1198 004204 004737 007776
1199 004210 010120
1200 004212 112720 000100
1201 004216 112720 000102
1202 004222 013737 004274 007772
1203 004230 004737 007776
1204 004234 012737 001000 004276
1205 004242 010120
1206 004244 005337 004276
1207 004250 001374
1208 .....
1209 .....
1210 004252 105020
1211 004254 004737 010074
1212 004260 006037 004274
1213 004264 001326
1214 004266 004737 010742
1215 004272 000402
1216 .....
1217 004274 000000
1218 004276 000000

;*****
;TEST 6 F GRAPH 0 DISPLAY A STEPPING HISTOGRAM LINE
;*****
TST6: SCOPE
      JSR RS,AMSG ;DISPLAY HEADER
      SHGLD
      MOV #BUFFER,RO ;LOAD OUTPUT BUFFER POINTER
      MOVB #ESC,(RO)+ ;ENTER VT55 MODE
      MOVB #61,(RO)+ ;ENTER "01" CODE
      MOVB #LDE0,(RO)+ ;LOAD ENABLE 0
      MOVB #BIT5,(RO)+ ;DISABLE DISPLAY
      MOVB #LDE1,(RO)+ ;LOAD ENABLE 1
      MOVB #BIT5:BIT4,(RO)+ ;CLEAR GRAPH, LINES AND CURSORS
      CLRB (RO)+ ;LOAD TERMINATOR
      JSR PC,XPRNT ;EXECUTE

      MOV #BIT7,100$ ;LOAD STARTING BASE LINE

2$: MOV #BUFFER,RO ;LOAD THE STARTING ADDRESS
   MOVB #ESC,(RO)+ ;LOAD "ESC" CODE
   MOVB #61,(RO)+ ;LOAD "01" ENTER CODE
   MOVB #LDE0,(RO)+ ;LOAD ENABLE 0
   MOVB #BIT5:BIT3:BIT1:BIT0,(RO)+ ;LOAD DISP. ENABLE, GRAPH 0, HISTO 1 ON
   MOVB #LNO,(RO)+ ;LOAD NOP
   MOVB #LSC,(RO)+ ;LOAD STARTING COORD.
   MOV #0,BASE ;GET BASE LINE
   JSR PC,SHUFF ;CONVERT
   MOV R1,(RO)+ ;SAVE COORD.
   MOVB #LNO,(RO)+ ;LOAD NOP
   MOVB #LDE0,(RO)+ ;LOAD "LOAD GRAPH"
   MOV #100$,BASE ;LOAD THE STARTING DATA VALUE
   JSR PC,SHUFF ;SHUFFLE THE DATA INTO VT-55 FORMAT
   MOV #MAXVRT,101$ ;LOAD COUNTER
   MOV R1,(RO)+ ;SAVE THE LSB MSB BYTE
   DEC 101$ ;DONE FULL GRAPH
   BNE 1$

1$: CLRB (RO)+ ;LOAD TERMINATOR
   JSR PC,XPRNT ;DISPLAY
   ROR 100$ ;CHANGE DATA VALUE
   BNE 2$ ;NO
   JSR PC,DELAY
   BR TST7 ;;NEXT TEST

100$: 0
101$: 0

```

E03

MAINDEC-11-DZVTD-8
DZVTD8.SAC T7MACY11 271732) 25-SEP-76 10:10 PAGE 31
G GRAPH 1 DISPLAY A STEPPING HISTOGRAM LINE

```

1219          ::*****
1220          ::TEST 7          G          GRAPH 1 DISPLAY A STEPPING HISTOGRAM LINE
1221          ::*****
1222          TST7:  SCOPE
1223          JSR      RS,AMSG          ;DISPLAY HEADER
1224          SHGL1
1225          MOV      @BUFFER,RO      ;LOAD OUTPUT BUFFER POINTER
1226          MOVB     @ESC,(RO)+      ;ENTER VT55 MODE
1227          MOVB     @B1,(RO)+      ;ENTER "01" CODE
1228          MOVB     @LDE0,(RO)+     ;LOAD ENABLE 0
1229          MOVB     @BIT5,(RO)+     ;DISABLE DISPLAY
1230          MOVB     @LDE1,(RO)+     ;LOAD ENABLE 1
1231          MOVB     @BIT5!BIT4,(RO)+ ;CLEAR GRAPH LINES AND CURSORS
1232          CLRB     (RO)+           ;LOAD TERMINATOR
1233          JSR      PC,XPRNT        ;EXECUTE
1234
1235          004352 012737 000200 004512      MOV      @BIT7,100$      ;LOAD STARTING BASE LINE
1236
1237          004360 012700 015562      2$:      MOV      @BUFFER,RO      ;LOAD THE STARTING ADDRESS
1238          004364 112720 000033      MOVB     @ESC,(RO)+      ;LOAD "ESC" CODE
1239          004370 112720 000061      MOVB     @B1,(RO)+      ;LOAD "01" ENTER CODE
1240          004374 112720 000101      MOVB     @LDE0,(RO)+     ;LOAD ENABLE 0
1241          004400 112720 000065      MOVB     @BIT5!BIT4!BIT2!BIT0,(RO)+ ;LOAD DISPLAY ENABLE AND GRAPH 1 ON
1242          004404 112720 000100      MOVB     @LNO,(RO)+      ;LOAD NOP
1243          004410 112720 000110      MOVB     @LSC,(RO)+      ;LOAD STARTING COORD.
1244          004414 012737 000000 007772      MOV      @J,BASE      ;GET BASE LINE
1245          004418 004737 007776      JSR      PC,SHUFF      ;CONVERT
1246          004422 010120      MOV      R1,(RO)+      ;SAVE COORD.
1247          004426 112720 000100      MOVB     @LNO,(RO)+      ;LOAD NOP
1248          004430 112720 000112      MOVB     @LDG1,(RO)+     ;LOAD "LOAD GRAPH"
1249          004434 013737 004512 007772      MOV      100$,BASE      ;LOAD THE STARTING DATA VALUE
1250          004438 004737 007776      JSR      PC,SHUFF      ;SHUFFLE THE DATA INTO VT-55 FORMAT
1251          004442 012737 001000 004514      MOV      @MAXVRT,101$    ;LOAD COUNTER
1252          004446 010120      MOV      R1,(RO)+      ;SAVE THE LSB MSB BYTE
1253          004450 005337 004514      DEC      101$      ;DONE FULL GRAPH
1254          004454 001374      BNE     1$
1255
1256          004470 105020      CLRB     (RO)+      ;LOAD TERMINATOR
1257          004472 004737 010074      JSR      PC,XPRNT      ;DISPLAY
1258          004476 006037 004512      ROR      100$
1259          004480 001326      BNE     2$      ;NO
1260          004484 004737 010742      JSR      PC,DELAY
1261          004488 000402      BR      TST10      ;;NEXT TEST
1262
1263          100$: 0
1264          101$: 0
1265
1266

```

F03

MAINCC-11-DZVTD-8
DZVTD8.SRC T7

MACY11 27(732) 25-SEP-76 10:10 PAGE 32
G GRAPH 1 DISPLAY A STEPPING HISTOGRAM LINE

```

1267
1268
1269
1270
1271 004516 000004
1272 004520 004537 011070
1273 004524 012420
1274 004526 012700 015562
1275 004532 112720 000033
1276 004536 112720 000061
1277 004542 112720 000101
1278 004546 112720 000040
1279 004552 112720 000111
1280 004556 112720 000060
1281 004562 105020
1282 004564 004737 010074
1283
1284 004570 004537 007242
1285 004574 012 102
1286 004576 000000
1287 004600 004737 010074
1288
1289 004604 004537 007242
1290 004610 036 112
1291 004612 100354
1292
1293 004614 112720 000033
1294 004620 112720 000135
1295 004624 112720 000100
1296 004630 112720 000100
1297 004634 112720 000100
1298 004640 112720 000100
1299 004644 112720 000100
1300 004650 112720 000100
1301 004654 112720 000100
1302 004660 112720 000100
1303 004664 112720 000100
1304 004670 112720 000100
1305 004674 112720 000100
1306 004700 112720 000100
1307 004704 112720 000100
1308 004710 112720 000100
1309 004714 112720 000100
1310 004720 112720 000100
1311 004724 105020
1312 004726 004737 010074
1313 004732 004737 010742
1314

```

```

*****
*TEST 10      H      HISTOGRAM ON GRAPH 0 AND 1
*****
TST10: SCOPE
      JSR      RS,AMSG          ;DISPLAY HEADER
      HGCAL
      MOV      @BUFFER,RO      ;LOAD OUTPUT BUFFER POINTER
      MOVB     @ESC,(RO)+      ;LOAD "ESC" CODE
      MOVB     @BLNO,(RO)+     ;LOAD COPY SCREEN
      MOVB     @BLNO,(RO)+     ;LOAD NOP TO LOAD SILO
      MOVB     @BLNO,(RO)+     ;LOAD NOP TO LOAD SILO
      MOVB     @BLNO,(RO)+     ;LOAD NOP TO LOAD SILO
      MOVB     @BLNO,(RO)+     ;LOAD NOP TO LOAD SILO
      MOVB     @BLNO,(RO)+     ;LOAD NOP TO LOAD SILO
      MOVB     @BLNO,(RO)+     ;LOAD NOP TO LOAD SILO
      MOVB     @BLNO,(RO)+     ;LOAD NOP TO LOAD SILO
      MOVB     @BLNO,(RO)+     ;LOAD NOP TO LOAD SILO
      MOVB     @BLNO,(RO)+     ;LOAD NOP TO LOAD SILO
      MOVB     @BLNO,(RO)+     ;LOAD NOP TO LOAD SILO
      MOVB     @BLNO,(RO)+     ;LOAD NOP TO LOAD SILO
      MOVB     @BLNO,(RO)+     ;LOAD NOP TO LOAD SILO
      MOVB     @BLNO,(RO)+     ;LOAD NOP TO LOAD SILO
      MOVB     @BLNO,(RO)+     ;LOAD NOP TO LOAD SILO
      MOVB     @BLNO,(RO)+     ;LOAD NOP TO LOAD SILO
      MOVB     @BLNO,(RO)+     ;LOAD NOP TO LOAD SILO
      MOVB     @BLNO,(RO)+     ;LOAD NOP TO LOAD SILO
      CLRB     (RO)+           ;TERMINATOR
      JSR      PC,XPRNT        ;EXECUTE
      JSR      PC,DELAY

```

```

1315      ;*****
1316      ;*TEST 11      I      CURSORS ON GRAPH 0
1317      ;*****
1318      ;ST11:  SCCPE
1319      ;          JSR      RS,AMSG      ;DISPLAY HEADER
1320      ;          CURGRD
1321      ;          MOV      #BUFFER,RO   ;LOAD OUPUT BUFFER POINTER
1322      ;          MOV      #ESC,(RO)+   ;LOAD VTSS MODE
1323      ;          MOV      #61,(RO)+   ;ENTER "01" CODE
1324      ;          MOV      #LDE0,(RO)+  ;LOAD ENABLE 0
1325      ;          MOV      #BITS,(RO)+  ;DISABE DISPLAY
1326      ;          MOV      #LDE1,(RO)+  ;LOAD ENABLE 1
1327      ;          MOV      #BITS:BIT4,(RO)+ ;CLEAR GRAPH, LINES MAD CURSRS
1328      ;          CLRB      (RO)+       ;LOAD TERMINATOR
1329      ;          JSR      PC,XPRNT     ;EXECUTE
1330
1331      ;          JSR      RS,UPDWN     ;LOAD DATA PATTERN
1332      ;          .BYTE     BIT1,LDC0   ;GRAPH 0
1333      ;          .WORD     0           ;DATA TO BE LOADED
1334      ;          JSR      PC,XPRNT     ;EXECUTE IT
1335
1336      ;          JSR      RS,CURSOR    ;ENABLE CURSORS
1337      ;          .BYTE     BIT1,LDC0   ;
1338      ;          .WORD     ADDLIN      ;SC = 0  GRAPH 0
1339
1340      ;          MOV      #ESC,(RO)+   ;LOAD "ESC" CODE
1341      ;          MOV      #COPY,(RO)+   ;LOAD COPY SCREEN
1342      ;          MOV      #LNO,(RO)+   ;LOAD NOP TO LOAD SILO
1343      ;          MOV      #LNO,(RO)+   ;LOAD NOP TO LOAD SILO
1344      ;          MOV      #LNO,(RO)+   ;LOAD NOP TO LOAD SILO
1345      ;          MOV      #LNO,(RO)+   ;LOAD NOP TO LOAD SILO
1346      ;          MOV      #LNO,(RO)+   ;LOAD NOP TO LOAD SILO
1347      ;          MOV      #LNO,(RO)+   ;LOAD NOP TO LOAD SILO
1348      ;          MOV      #LNO,(RO)+   ;LOAD NOP TO LOAD SILO
1349      ;          MOV      #LNO,(RO)+   ;LOAD NOP TO LOAD SILO
1350      ;          MOV      #LNO,(RO)+   ;LOAD NOP TO LOAD SILO
1351      ;          MOV      #LNO,(RO)+   ;LOAD NOP TO LOAD SILO
1352      ;          MOV      #LNO,(RO)+   ;LOAD NOP TO LOAD SILO
1353      ;          MOV      #LNO,(RO)+   ;LOAD NOP TO LOAD SILO
1354      ;          MOV      #LNO,(RO)+   ;LOAD NOP TO LOAD SILO
1355      ;          MOV      #LNO,(RO)+   ;LOAD NOP TO LOAD SILO
1356      ;          MOV      #LNO,(RO)+   ;LOAD NOP TO LOAD SILO
1357      ;          MOV      #LNO,(RO)+   ;LOAD NOP TO LOAD SILO
1358      ;          CLRB      (RO)+       ;LOAD TERMINATOR
1359      ;          JSR      PC,XPRNT     ;EXECUTE
1360
1361      ;          JSR      RS,CURSOR    ;REMOVE CURSORS
1362      ;          .BYTE     BIT1,LDC0   ;ON GRAPH 0
1363      ;          .WORD     BIT15
1364      ;          JSR      PC,XPRNT     ;EXECUTE
1365      ;          JSR      PC,DELAY
1366

```

1367			
1368			
1369			
1370			
1371	005172	000004	
1372	005174	000037	011070
1373	005200	012512	
1374	005202	012700	015562
1375	005206	112720	000033
1376	005212	112720	000061
1377	005216	112720	000101
1378	005222	112720	000040
1379	005226	112720	000111
1380	005232	112720	000060
1381	005236	105020	
1382	005240	004737	010074
1383			
1384	005244	004537	007242
1385	005250	004	112
1386	005252	100354	
1387	005254	004737	010074
1388			
1389	005260	004537	007574
1390	005264	004	113
1391	005266	002000	
1392			
1393	005270	112720	000033
1394	005274	112720	000135
1395	005300	112720	000100
1396	005304	112720	000100
1397	005310	112720	000100
1398	005314	112720	000100
1399	005320	112720	000100
1400	005324	112720	000100
1401	005330	112720	000100
1402	005334	112720	000100
1403	005340	112720	000100
1404	005344	112720	000100
1405	005350	112720	000100
1406	005354	112720	000100
1407	005360	112720	000100
1408	005364	112720	020100
1409	005370	112720	000100
1410	005374	112720	000100
1411	005400	105020	
1412	005402	004737	010074
1413	005406	004537	007574
1414	005412	004	113
1415	005414	100000	
1416	005416	004737	010074
1417	005422	004737	010742

[illegible]

MAINDEC-11-DZVTD-B
 DZVTD8.SRC T13

 MACY11 27(732) 25-SEP-76 10:10 PAGE 36
 K TEST STARTING COORDINATE ON GRAPH 0

1474	005676	112720	000100	MOV	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
1475	005702	112720	000100	MOV	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
1476	005706	112720	000100	MOV	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
1477	005712	105020		CLRB	(R0)+	;LOAD TERMINATOR
1478	005714	004737	010074	JSR	PC, XPRNT	;EXECUTE
1479	005720	004737	010742	JSR	PC, DELAY	
1480						
1481						
1482						
1483						
1484	005724	000004				
1485	005726	004537	011070	JSR	RS, MSG	;DISPLAY HEADER
1486	005732	012615		SCORD1		
1487	005734	012700	015562	MOV	#BUFFER, R0	;LOAD OUTPUT BUFFER POINTER
1488	005740	112720	000033	MOV	#ESC, (R0)+	;ENTER VT-55 MODE
1489	005744	112720	000061	MOV	#61, (R0)+	
1490	005750	112720	000101	MOV	#LDE0, (R0)+	;LOAD ENABLE 0
1491	005754	112720	000040	MOV	#BITS, (R0)+	;DISABLE DISPLAY
1492	005760	112720	000111	MOV	#LDE1, (R0)+	;LOAD ENABLE 1
1493	005764	112720	000060	MOV	#BITS!BIT4, (R0)+	;CLEAR GRAPH, LILES, AND CURSORS
1494	005770	105020		CLRB	(R0)+	;LOAD TERMINATOR
1495	005772	004737	010074	JSR	PC, XPRNT	;EXECUTE
1496						
1497	005776	004537	007242	JSR	RS, UPDOWN	;LOAD GRAPH 0 DATA
1498	006002	004	112	.BYTE	BIT2, LDG1	
1499	006004	100354		.WORD	BIT15!MAXHOZ	
1500	006006	004737	010074	JSR	PC, XPRNT	;EXECUTE
1501						
1502	006012	012703	000004	MOV	#4, R3	;LOAD SINE COUNT
1503	006016	012701	000777	MOV	#MAXVRT-1, R1	;LOAD STARTING CORD.
1504	006022	012700	007201	MOV	#SINEND-40, R0	;LOAD SA POINTER
1505	006026	000402		BR	1\$	
1506	006030	012700	007241	3\$: MOV	#SINEND, R0	;LOAD SINE POINTER
1507	006034	114037	006054	1\$: MOV	-(R0), 10\$	;LOAD SINE DATA WORD
1508	006040	001413		BEQ	2\$	;BR IF NO MORE DATA
1509	006042	010137	006056	MOV	R1, 11\$	;LOAD STARTING COORDINATE
1510						
1511	006046	004537	007436	JSR	RS, STCORD	;LOAD DATA INTO BUFFER
1512	006052	004	112	.BYTE	BIT2, LDG1	;FOR GRAPH 1
1513	006054	000000		10\$: .WORD	0	;DATA TO BE LOADED
1514	006056	000000		11\$: .WORD	0	;STARTING COORD.
1515						
1516	006060	004737	010074	JSR	PC, XPRNT	;EXECUTE
1517	006064	005301		DEC	R1	;DONE ALL COORD. ?
1518	006066	000762		BR	1\$	
1519	006070	005303		2\$: DEC	R3	;DONE ALL LINES ?
1520	006072	001356		BNE	3\$	;BR IF NOT
1521	006074	012700	015562	MOV	#BUFFER, R0	
1522	006100	112720	000033	MOV	#ESC, (R0)+	;LOAD "ESC" CODE
1523	006104	112720	000135	MOV	#COPY, (R0)+	;LOAD COPY SCREEN
1524	006110	112720	000100	MOV	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
1525	006114	112720	000100	MOV	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
1526	006120	112720	000100	MOV	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
1527	006124	112720	000100	MOV	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
1528	006130	112720	000100	MOV	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
1529	006134	112720	000100	MOV	#LNO, (R0)+	;LOAD NOP TO LOAD SILO

K03

MAINDEC-11-DZVTD-8
DZVTD8.SRC T14MACY11 27(732) 25-SEP-76 10:10 PAGE 37
L TEST STARTING COORDINATE ON GRAPH 1

1530	006140	112720	000100	MOVB	#LNO, (R3);	;LOAD NOP TO LOAD SILO
1531	006144	112720	000100	MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
1532	006150	112720	000100	MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
1533	006154	112720	000100	MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
1534	006160	112720	000100	MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
1535	006164	112720	000100	MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
1536	006170	112720	000100	MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
1537	006174	112720	000100	MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
1538	006200	112720	000100	MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
1539	006204	112720	000100	MOVB	#LNO, (R0)+	;LOAD NOP TO LOAD SILO
1540	006210	105020		CLRB	(R0)+	
1541	006212	004737	010074	JSR	PC, XPRNT	;EXECUTE
1542	006216	004737	010742	JSR	PC, DELAY	
1543						

MAINDEC-11-DZVTD-B
DZVTD8.SRC T14

MACY11 27(732) 25-SEP-76 10:10 PAGE 38
L TEST STARTING COORDINATE ON GRAPH 1

```

1544
1545
1546
1547
1548 006222 000004
1549
1550
1551 006224 012702 000014
1552 006230 012700 015562
1553 006234 112720 000033
1554 006240 112720 000061
1555 006244 112720 000101
1556 006250 112720 000040
1557 006254 112720 000111
1558 006260 112720 000060
1559 006264 112720 000033
1560 006270 112720 000062
1561 006274 105020
1562 006276 004737 010074
1563 006302 012700 015562
1564 006306 112720 000015
1565 006312 112720 000012
1566 006316 112720 000015
1567 006322 112720 000012
1568 006326 012701 000050
1569 006332 012720 044110
1570 006336 005301
1571 006340 001374
1572 006342 105020
1573 006344 004737 010074
1574 006350 005302
1575 006352 001374
1576
1577
1578
1579 006354 012700 015562
1580 006360 112720 000033
1581 006364 112720 000061
1582 006370 112720 000101
1583 006374 112720 000041
1584 006400 112720 000111
1585 006404 112720 000043
1586 006410 112720 000100
1587 006414 112720 000104
1588 006420 012737 002002 007772
1589 006426 004737 007776
1590 006432 010120
1591 006434 062737 000024 007772
1592 006442 022737 002340 007772
1593 006450 100366
1594 006452 105020
1595 006454 004737 010074

;*****
;TEST 15 M VT55 ADJUSTMENT PATTERN
;*****
;ST15: SCOPE
;FILL THE SCREEN WITH THE H CHARACTER

MOV #12, R2
MOV #BUFFER, R0
MOVB #ESC, (R0)+
MOVB #61, (R0)+
MOVB #LDE0, (R0)+
MOVB #BIT5, (R0)+
MOVB #LDE1, (R0)+
MOVB #BIT5:BIT4, (R0)+
MOVB #ESC, (R0)+
MOVB #62, (R0)+
CLRB (R0)+
JSR PC, XPRNT
MOV #BUFFER, R0
MOVB #15, (R0)+
MOVB #12, (R0)+
MOVB #15, (R0)+
MOVB #12, (R0)+
MOV #40, R1
15: MOV #44110, (R0)+
DEC R1
BNE 15
CLRB (R0)+
25: JSR PC, XPRNT
DEC R2
BNE 25

;NOW INSTALL THE HORIZONTAL LINES

MOV #BUFFER, R0
MOVB #33, (R0)+
MOVB #61, (R0)+
MOVB #LDE0, (R0)+
MOVB #BIT5:BIT0, (R0)+
MOVB #LDE1, (R0)+
MOVB #BIT5:BIT1:BIT0, (R0)+
MOVB #LNO, (R0)+
MOVB #LHVD, (R0)+
MOV #ADDLIN+2, BASE
35: JSR PC, SHUFF
MOV R1, (R0)+
ADD #24, BASE
CMP #ADDLIN+340, BASE
BPL 35
CLRB (R0)+
JSR PC, XPRNT

;LOAD COUNT
;LOAD BUFFER POINTER
;ENTER VT55 FORMAT
;LOAD ENTER "01" CODE
;LOAD ENABLE 0
;DISABLE DISPLAY
;LOAD ENABLE 1
;CLEAR GRAPH, LINES, AND CURSORS
;ENTER VT55 FORMAT
;EXIT "02" CODE
;LOAD TERMINATOR
;EXECUTE
;LOAD BUFFER POINTER
;LOAD CR
;LOAD LF
;LOAD CR
;LOAD LF
;LOAD SCREEN WIDTH
;LOAD ASCII H
;FINISHED ?
;BR IF NOT
;LOAD TERM.
;XFER TO SCREEN
;FINISHED ALL LINES
;BR IF NOT

;LOAD BUFFER POINTER
;ENABLE CHART MODE
;LOAD ENABLE 0
;LOAD DISPLAY ENABLE
;LOAD ENABLE 1
;ENABLE HORIZ.+ VERT LINES
;LOAD HORIZ INST
;LOAD BASE LINE VALUE
;SHUFFLE THE DATA
;SAVE THE DATA
;UPDATE BASE LINE VALUE
;TEST FOR GREATER THAN VALID
;BR IF OK
;LOAD TERM
;XMIT TO THE SCREEN

```

```

1596
1597
1598
1599
1600 006460 012700 015562      MOV      #BUFFER,RO      ;LOAD OUTPUT BUFFER POINTER
1601 006464 112720 000033      MOV      #33,(RO)+      ;LOAD ESC
1602 006470 112720 000061      MOV      #61,(RO)+      ;ENTER CHART MODE
1603 006474 112720 000100      MOV      #LNO,(RO)+      ;LOAD NOP
1604 006500 112720 000114      MOV      #LHV1,(RO)+      ;LOAD VERT. LINE INST.
1605 006504 012737 002000 007772      MOV      #ADDLIN,BASE      ;LOAD STARTING LINE #
1606 006512 004737 007776      JSR      PC,SHUFF      ;SHUFFLE THE DATA
1607 006516 010120      MOV      R1,(RO)+      ;SAVE DATA
1608 006520 062737 000015 007772      ADD      #15,BASE      ;UPDATE DATA
1609 006526 022737 003000 007772      CMP      #ADDLIN!100L,BASE      ;TEST FOR LAST DATA LINE
1610 006534 100366      BPL      4$      ;BR IF NOT DONE
1611 006536 012737 002777 007772      MOV      #ADDLIN!777,BASE      ;LOAD LAST LINE
1612 006544 004737 007776      JSR      PC,SHUFF      ;SHUFFLE THE DATA
1613 006550 010120      MOV      R1,(RO)+      ;LOAD LAST DATA LINE
1614 006552 112720 000033      MOV      #ESC,(RO)+      ;LOAD "ESC" CODE
1615 006556 112720 000135      MOV      #COPY,(RJ)+      ;LOAD COPY SCREEN
1616 006562 112720 000100      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1617 006566 112720 000100      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1618 006572 112720 000100      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1619 006576 112720 000100      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1620 006602 112720 000100      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1621 006606 112720 000100      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1622 006612 112720 000100      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1623 006616 112720 000100      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1624 006622 112720 000100      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1625 006626 112720 000100      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1626 006632 112720 000100      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1627 006636 112720 000100      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1628 006642 112720 000100      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1629 006646 112720 000100      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1630 006652 112720 000100      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1631 006656 112720 000100      MOV      #LNO,(RO)+      ;LOAD NOP TO LOAD SILO
1632 006662 105020      CLRB      (RO)+      ;LOAD TERM.
1633 006664 004737 010074      JSR      PC,XPRNT      ;SEND TO SCREEN
1634
1635 006670 004737 010742      JSR      PC,DELAY      ;DELAY

```

;NOW LOAD THE VERTICAL LINES

4\$:

MAINDEC-11-DZVTD-B
DZVTD8.SRC T15

MACY11 27(732) 25-SEP-76 10:10 PAGE 40
M VT55 ADJUSTMENT PATTERN

```

1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647 006674
1648 006674 000004
1649 006676 005737 001310
1650 006702 001411
1651 006704 023737 001310 001312
1652 006712 001405
1653 006714 062737 000010 001312
1654 006722 000137 002016
1655 006726 000240
1656 006730 005037 001102
1657 006734 005237 001202
1658 006740 042737 100000 001202
1659 006746 005327
1660 006750 000001
1661 006752 003015
1662 006754 012737
1663 006756 000001
1664 006760 006750
1665 006762 104400 007015
1666 006766 013700 000042
1667 006772 001405
1668 006774 000005
1669 006776 004710
1670 007000 000240
1671 007002 000240
1672 007004 000240
1673 007006
1674 007006 000137
1675 007010 002010
1676 007012 377 377 000
1677 007015 015 042412 042116
1678 007022 050040 051501 000123
1679

.SBTTL END OF PASS ROUTINE

*****
; INCREMENT THE PASS NUMBER ($PASS)
; TYPE "END PASS"
; IF THERES A MONITOR GO TO IT
; IF THERE ISN'T JUMP TO RSTRTA
; IF IT IS DESIRED TO HAVE A BELL INDICATE THE "END OF PASS" LOCATION
; SENDMG CAN BE CHANGED TO 7.

SEOP:
SCOPE
TST LAST ; TEST IF MORE
BEQ IS ; BR IF NONE
CMP LAST,VTNOW ; IS THIS THE LAST ONE
BEQ IS ; BR IF YES
ADD #10,VTNOW
JMP RSTAT ; TEST NEXT ONE

IS:
NOP
CLR $STNM ; ZERO THE TEST NUMBER
INC $PASS ; INCREMENT THE PASS NUMBER
BIC #100000,$PASS ; DON'T ALLOW A NEG. NUMBER
DEC (PC)+ ; LOOP?

SEOPCT: .WORD 1
BGT $DOAGN ; YES
MOV (PC)+,2(PC)+ ; RESTORE COUNTER

SENDCT: .WORD 1
TYPE SENDMG ; TYPE "END PASS"
MOV #42,R0 ; GET MONITOR ADDRESS
BEQ $DOAGN ; BRANCH IF NO MONITOR
RESET ; CLEAR THE WORLD
SENDAD: JSR PC,(R0) ; GO TO MONITOR
NOP ; SAVE ROOM
NOP ; FOR
NOP ; ACT11

$DOAGN: JMP 2(PC)+ ; RETURN

$RTNAD: .WORD RSTRTA
$ENULL: .BYTE -1,-1,0 ; NULL CHARACTER STRING
$ENDMG: .ASCIZ <15><12>/END PASS/

```

MAINDEC-11-DZVTD-B
DZVTD8.SACEND OF MACY11 27(732)
PASS ROUTINE

25-SEP-76 10:10 PAGE 41

1680	007030	000	003	004	.BYTE	000,003,004,004,005
1681	007033	004	005			
1682	007035	006	007	010	.BYTE	006,007,010,012,014,015,017,021,024,027,032,035,040
1683	007040	012	014	015		
1684	007043	017	021	024		
1685	007046	027	032	035		
1686	007051	040				
1687	007052	043	047	053	.BYTE	043,047,053,056,062,066,073,077,103,110,115,121,126,133
1688	007055	056	062	066		
1689	007060	073	077	103		
1690	007063	110	115	121		
1691	007066	126	133			
1692	007070	137	144	151	.BYTE	137,144,151,156,162,167,174,201,206,213,217,223,230,235
1693	007073	156	162	167		
1694	007076	174	201	206		
1695	007101	213	217	223		
1696	007104	230	235			
1697	007106	241	245	251	.BYTE	241,245,251,255,261,265,271,274,277,303,305,310,313
1698	007111	255	261	265		
1699	007114	271	274	277		
1700	007117	303	305	310		
1701	007122	313				
1702	007123	316	317	321	.BYTE	316,317,321,323,324,326,327,330,331,331,331,331,331
1703	007126	323	324	326		
1704	007131	327	330	331		
1705	007134	331	331	331		
1706	007137	331				
1707	007140	330	327	326	.BYTE	330,327,326,325,323,322,320,317,314,312,307
1708	007143	325	323	322		
1709	007146	323	317	314		
1710	007151	312	307			
1711	007153	304	301	275	.BYTE	304,301,275,272,267,263,257,253,247,243,237,232,225
1712	007156	272	267	263		
1713	007161	257	253	247		
1714	007164	243	237	232		
1715	007167	225				
1716	007170	221	214	207	.BYTE	221,214,207,203,176,171,164,157,153,145,140
1717	007173	203	176	171		
1718	007176	164	157	153		
1719	007201	145	140			
1720	007203	134	127	121	.BYTE	134,127,121,115,111,104,077,073,066,063,057,053,047
1721	007206	115	111	104		
1722	007211	077	073	066		
1723	007214	063	057	053		
1724	007217	047				
1725	007220	043	040	034	.BYTE	043,040,034,031,027,023,021,017,015,013,011,010,007
1726	007223	031	027	023		
1727	007226	021	017	015		
1728	007231	013	011	010		
1729	007234	007				
1730	007235	006	005	004	.BYTE	006,005,004,003
1731	007240	003				
1732	007241	000				
1733					SINEND: .BYTE	0
1734					.EVEN	
1735					;UP-DOWN SUBROUTINE	

MAINDEC-11-DZVTD-B MACY11 27(732) 25-SEP-76 10:10 PAGE 42
 DZVTD9.SRC END OF PASS ROUTINE

1736	007242	012537	007432	UPDOWN:	MOV	(R5)+,11\$	:GET GRAPH BIT AND INC/DEC WORD
1737	007246	012537	007434		MOV	(R5)+,12\$	:GET STARTING COORD.
1738	007252	012700	015562		MOV	#BUFFER,R0	:LOAD THE POINTER
1739	007256	112720	000033		MOVB	#ESC,(R0)+	:LOAD 'ESCAPE'
1740	007262	112720	000061		MOVB	#61,(R0)+	:ENABLE GRAPHIC MODE
1741	007266	112720	000101		MOVB	#LOEO,(R0)+	:LOAD ENABLE 0
1742	007272	112710	000041		MOVB	#BITS,BITO,(R0)	:ENABLE DISPLAY
1743	007276	153720	00743		BISB	11\$,(R0)+	:LOAD GRAPH ENABLE BIT
1744	007302	012737	000001	007430	MOV	#1,10\$	
1745	007310	112720	000100		MOVB	#LNO,(R0)+	:LOAD NOP
1746	007314	112720	000110		MOVB	#LSC,(R0)+	:LOAD STARTING CORD.
1747	007320	005037	007772		CLR	BASE	:LOAD START. CORD.
1748	007324	004737	007776		JSR	PC,SHUFF	
1749	007330	010120			MOV	R1,(R0)+	:LOAD INTO BUFFER
1750	007332	112720	0001J0		MOVB	#LNO,(R0)+	:LOAD NOP
1751	007336	113720	007433		MOVB	11\$+1,(R0)+	:LOAD LOAD GRAPH X
1752	007342	013737	007434	007772	MOV	12\$,BASE	:LOAD DATA
1753	007350	004737	007776	4\$:	JSR	PC,SHUFF	:SHUFFEL DATA
1754	007354	010120		1\$:	MOV	R1,(R0)+	:LOAD DATA
1755	007356	005737	007434		TST	12\$	:TEST FOR UP OR DOWN DATA
1756	007362	100007			BPL	2\$	:BR IF INC.
1757	007364	005337	007772		DEC	BASE	:DEC. DATA
1758	007370	042737	177400	007772	BIC	#177400,BASE	
1759	007376	001407			BZQ	5\$	
1760	007400	00076			JR	1\$	
1761	007402	005237	007772	2\$:	INC	BASE	:CHANGE DATA
1762	007406	022737	000354	007772	CMP	#MAXNOZ,BASE	:TEST FOR LAST
1763	007414	001355			BNE	1\$	:BR IF NOT
1764	007416	005337	007430	5\$:	DEC	10\$	
1765	007422	100347			BPL	4\$	:BR IF NOT
1766	007424	105010			CLRB	(R0)	:LOAD TERMINATOR
1767	007426	000205			RTS	RS	:EXIT
1768	007430	000000		10\$:	0		
1769	007432	000000		11\$:	0		
1770	007434	000000		12\$:	0		

```

1771                                     ;STARTING COORDINATE SUBROUTINE
1772
1773 007436 012537 007570      STCORD: MOV      (RS)+,11$      :GET GRAPH BIT AND INC/DEC WORD
1774 007442 012537 007572      MOV      (RS)+,12$      :GET STARTING CORD.
1775 007446 012537 007566      MOV      (RS)+,10$      :GET ARG. WORD
1776 007452 010046      MOV      RO,-(SP)      :SAVE RO
1777 007454 010146      MOV      R1,-(SP)      :SAVE R1
1778 007456 012700 015562      MOV      @BUFFER,RO      :LOAD THE POINTER
1779 007462 112720 000033      MOV      @ESC,(RO)+      :LOAD 'ESCAPE'
1780 007466 112720 000061      MOV      @61,(RO)+      :ENABLE GRAPHIC MODE
1781 007472 112720 000101      MOV      @LNO,(RO)+      :LOAD ENABLE 0
1782 007476 112710 000041      MOV      @BIT0,(RO)+      :ENABLE DISPLAY
1783 007502 153720 007570      BISH      11$,(R1)+      :LOAD GRAPH ENABLE BIT
1784 007506 112720 000100      MOV      @LNO,(R1)+      :LOAD NOP
1785 007512 112720 000110      MOV      @LSC,(R1)+      :LOAD STARTING CORD.
1786 007516 013737 007566 007772      MOV      10$,BASE      :LOAD START. CORD.
1787 007524 004737 007776      JSR      PC,SHUFF      :
1788 007530 010120      MOV      R1,(RO)+      :LOAD INTO BUFFER
1789 007532 112720 000100      MOV      @LNO,(R1)+      :LOAD NOP
1790 007536 113720 007571      MOV      11$+1,(R1)+      :LOAD LOAD GRAPH X
1791 007542 013737 007572 007772 4$:      MOV      12$,BASE      :LOAD DATA
1792 007550 004737 007776 1$:      JSR      PC,SHUFF      :SHUFFLE DATA
1793 007554 010120      MOV      R1,(RO)+      :LOAD DATA
1794 007556 105010      CLRB      (RO)      :LOAD TERMINATOR
1795 007560 012601      MOV      (SP)+,R1
1796 007562 012600      MOV      (SP)+,RO
1797 007564 000205      RTS
1798 007566 000000      10$:      0
1799 007570 000000      11$:      0
1800 007572 000000      12$:      0

```

;EXIT

```

1801
1802
1803
1804 007574 012537 007766
1805 007600 012537 007770
1806 007604 012700 015562
1807 007610 112720 000033
1808 007614 112720 000061
1809 007620 112720 000101
1810 007624 112710 000041
1811 007630 153720 007766
1812 007634 112720 000111
1813 007640 013701 007766
1814 007644 006301
1815 007646 112710 000040
1816 007652 150120
1817 007654 112720 000100
1818 007660 112720 000110
1819 007664 005037 007772
1820 007670 004737 007776
1821 007674 010120
1822 007676 112720 000100
1823 007702 113720 007767
1824 007706 013737 007770 007772
1825 007714 004737 007776
1826 007720 010120
1827 007722 005737 007770
1828 007726 103007
1829 007730 005337 007772
1830 007734 042737 177000 007772
1831 007742 001407
1832 007744 000763
1833 007746 005237 007772
1834 007752 022737 003000 007772
1835 007760 001355
1836 007762 105010
1837 007764 000205
1838 007766 000000
1839 007770 000000
1840 007772 000000
1841 007774 000000
1842

```

;CURSOR SUBROUTINE

```

CURSOR: MOV (RS)+,11$ ;GET ARG. WORD
MOV (RS)+,12$ ;GET ARG. WORD
MOV $BUFFER,R0 ;LOAD THE POINTER
MOVB $ESC,(R0)+ ;LOAD 'ESCAPE'
MOVB $51,(R0)+ ;ENABLE GRAPHIC MODE
MOVB $LDE0,(R0)+ ;LOAD ENABLE 0
MOVB $BITS!BIT0,(R0)+ ;ENABLE DISPLAY
BISB 11$,(R0)+ ;LOAD GRAPH ENABLE BIT
MOVB $LDE1,(R0)+ ;LOAD ENABLE
MOV 11$,R1
RSL R1
MOVB $BITS,(R0)
BISB R1,(R0)+
MOVB $LNO,(R0)+ ;LOAD NOP
MOVB $LSC,(R0)+ ;LOAD STARTING CORD.
CLR BASE ;LOAD START. CORD.
JSR PC,SHUFF
MOV R1,(R0)+ ;LOAD INTO BUFFER
MOVB $LNO,(R0)+ ;LOAD NOP
MOVB 11$+1,(R0)+ ;LOAD "LOAD CURSOR ON GRAPH X"
MOV 12$,BASE ;LOAD CURSOR POSITION DATA
JSR PC,SHUFF ;SHUFFLE DATA
MOV R1,(R0)+ ;LOAD DATA
TST 12$ ;TEST FOR UP OR DOWN DATA
BPL 2$ ;BR IF INC.
DEC BASE
BIC $177000,BASE
BEQ 4$
BR 1$
2$: INC BASE ;CHANGE DATA
3$: CMP $ADDLIN!MAXVRT,BASE ;TEST FOR LAST
BNE 1$ ;BR IF NOT
4$: CLRB (R0) ;LOAD TERMINATOR
RTS RS ;EXIT
11$: 0
12$: 0
BASE: 0
BASE1: 0

```

F04

MAINDEC-11-DZVTD-8 MACY11 27(732) 25-SEP-76 10:10 PAGE 45
 DZVTD8.SRC END OF PASS ROUTINE

```

1843                                     ;SUBROUTINE TO SHUFFLE DATA INTO VT-55 DATA BYTE FORMAT
1844
1845 007776 013702 007772 SHUFF: MOV     BASE,R2          ;LOAD VALUE TO BE SHUFFLED
1846 010002 010237 007774      MOV     R2,BASE1
1847 010006 042737 177740 007774      BIC     #177740,BASE1
1848 010014 010201      MOV     R2,R1
1849 010016 006001      ROR     R1
1850 010020 006001      ROR     R1
1851 010022 006001      ROR     R1
1852 010024 006001      ROR     R1
1853 010026 006001      ROR     R1
1854 010030 110137 007775      MOV     R1,BASE1+1      ;RELOAD R1
1855 010034 042737 170340 007774      BIC     #170340,BASE1      ;MASK
1856 010042 052737 020040 007774      BIS     #20040,BASE1      ;CONVERT TO ASCII
1857 010050 032737 002000 007772      BIT     #BIT10,BASE
1858 010056 001403      BEQ     IS      ;BR IF NOT SET
1859 010060 052737 010000 007774      BIS     #BIT12,BASE1      ;SET BIT
1860 010066 013701 007774      MOV     BASE1,R1
1861 010072 000207      RTS     PC      ;EXIT
  
```

```

1862                                     ;DISPLAY SUBROUTINE
1863
1864 010074 010046 XPRNT: MOV      RD,-(SP)
1865 010076 010146      MOV      R1,-(SP)
1866 010100 005037 010546      CLR      ANESC      ;HOUSEKEEP
1867 010104 005037 010552      CLR      TERM
1868 010110 005037 010550      CLR      NOEXIT
1869 010114 012700 015562      MOV      @BUFFER,RD      ;SETUP BUFFER POINTER
1870 010120 105777 171204 1S:  TOS      @VTOS      ;TEST READY
1871 010124 100375      BPL      1S
1872 010126 104405      CKSWR      ;TEST IF "CTRL G" ON CONSOLE TTY
1873 010130 000240      NOP
1874 010132 000240      NOP
1875 010134 000240      NOP
1876 010136 000240      NOP
1877 010140 000240      NOP
1878 010142 112001 2S:  MOV      (RD)+,R1      ;GET A CHAR.
1879 010144 001562      BEQ      16S      ;BR IF TERM
1880 010146 122701 000033      CMPB     #33,R1      ;TEST FOR ESC
1881 010152 001031      BNE      4S      ;BR IF NOT
1882 010154 122710 000135      CMPB     @COPY,(RD)      ;TEST IF NEXT CHAR IS A "COPY"
1883 010160 001023      BNE      3S      ;BR IF NOT
1884 010162 032777 004000 170750      BIT      @BIT11,2SWR      ;TEST IF "INHIBIT" COPIER TESTING
1885 010170 001402      BEQ      13S      ;BR IF NOT
1886 010172 105720      STB      (RD)+      ;YES - ADJUST POINTER
1887 010174 000762      BR      2S
1888 010176 032777 002000 170734 13S:  BIT      @BIT10,2SWR      ;TEST IF "SAVE PAPER MODE" IS ENABLED
1889 010204 001407      BEQ      20S      ;BR IF NOT
1890 010206 013737 001202 010542      MOV      $PASS,30S      ;GET PASS COUNT
1891 010214 042737 177770 010542      BIC      @177770,30S      ;MASK OF BITS
1892 010222 001363      BNE      21S      ;BR IF NOT 1 OR 8 PASSES
1893 010224 105237 010550      INCB     NOEXIT      ;SET "DO NOT EXIT UNTILL X-ON"
1894 010230 005237 010546      INC      ANESC      ;SET SOFT FLAG
1895 010234 000402      BR      5S
1896 010236 005037 010546      CLR      ANESC      ;CLEAR SOFT FLAG
1897 010242 110177 171064      MOV      R1,@VT08      ;LOAD CHAR
1898 010246 105777 171052      TSTB     @VTIS      ;TEST INPUT FLAG
1899 010252 100322      BPL      1S      ;BR IF CLEARED
1900 010254 005737 010546      TST      ANESC      ;TEST IF "ESC" WAS JUST SENT
1901 010260 001317      BNE      1S
1902 010262 005037 001124      CLR      $GOODAT      ;CLEAR EXPECTED DATA
1903 010266 000420      BR      10S
1904

```

H04

 MAINDEC-11-DZVTD-B MACY11 27(732) 25-SEP-76 10:10 PAGE 47
 DZVTD8.SAC END OF PASS ROUTINE

```

1905 ;WAIT FOR A KEYBOARD FLAG - ERROR REPORT IF NONE
1906
1907 010270 013737 001316 011162 6S: MOV TIME0,TIME1
1908 010276 005037 011164 ;LOAD DELAY
1909 010302 105777 171016 7S: CLR TIME2 ;TEST IF INPUT FLAG
1910 010306 100410 BMI 10S ;BR IF SET
1911 010310 005337 011164 DEC TIME2 ;DELAY
1912 010314 001372 BNE 7S
1913 010316 104405 CKSWR ;TEST FOR "CTRG G" ON CONSOLE TTY
1914 010320 001337 011162 DEC TIME1 ;DELAY
1915 010324 001366 BNE 7S
1916 010326 000457 BR 14S ;REPORT ERROR
1917
1918 ;INFUT FLAG SET - FIND OUT WHAT CHARACTER IT WAS
1919
1920 010330 017737 170772 001126 10S: MOV 2VTIB,S80DAT ;READ CHAR
1921 010336 042737 177600 001126 BIC 8177600,S80DAT ;MASK
1922 010344 022737 000021 001126 CMP 8XON,S80DAT ;TEST FOR X ON
1923 010352 001006 BNE 11S
1924 010354 005037 010550 CLR NOEXIT
1925 010360 005737 010552 TST TERM ;TEST IF TERMINATOR WAS SET
1926 010364 001655 BEQ 1S ;BR IF NOT
1927 010366 000462 BR 17S ;BR IF ONLY ONE
1928 010370 022737 000023 001126 11S: CMP 8XOFF,S80DAT ;TEST FOR X OFF
1929 010376 001006 BNE 12S ;BR IF NOT
1930 010400 105237 010550 INCB NOEXIT ;SET "NO EXIT UNTIL X-ON" RCVD
1931 010404 012737 000621 001124 MOV 8XON,S80DAT ;LOAD EXPECTED VALUE
1932 010412 000726 BR 6S ;WAIT FOR X-ON
1933 010414 005037 001124 12S: CLR S80DAT ;LOAD EXPECTED
1934 010420 105777 170514 TSTB 2SWR ;TEST SWR
1935 010424 100020 BPL 14S ;BR IF CLEARED
1936 010426 022737 000057 001126 CMP 8'/',S80DAT ;COMPARE
1937 010434 001417 BEQ 15S ;BR IF EQUAL
1938 010436 022737 000134 001126 CMP 8'\',S80DAT ;COMPARE
1939 010444 001010 BNE 14S ;BR IF NOT
1940 010446 012737 000001 010544 MOV 81,LOOP ;SET SOFT FLAG
1941 010454 012737 011277 010572 MOV 8M00,FINDTA ;SETUP MESSAGE
1942 010462 000137 010562 JMP FINDOT
1943
1944 010466 104001 14S: ERROR 1 ;UNEXPECTED OR INCORRECT INPUT FLAG
1945 010470 000137 010130 JMP 1S
1946
1947 010474 005037 010544 15S: CLR LOOP
1948 010500 012737 011362 010572 MOV 8M01,FINDTA ;SETUP MESSAGE
1949 010506 000137 010562 JMP FINDOT
1950 010512 012737 000001 010552 16S: MOV 81,TERM ;SET "TERMINATOR HAS BEEN FOUND" FLAG
1951 010520 012737 000021 001124 MOV 8XON,S80DAT ;LOAD EXPECTED
1952 010536 105737 010550 TSTB NOEXIT ;TEST IF ALLOWED TO LEAVE ROUTINE ?
1953 010532 001256 BNE 6S ;BR IF NOT ALLOWED
1954 010534 012601 17S: MOV (SP)+,R1
1955 010536 012600 MOV (SP)+,R0
1956 010540 000207 RTS PC ;EXIT
1957
1958 010542 000000 30S: 0
1959 010544 000000 LOOP: 0
1960 010546 000000 MESC: 0

```

1961	010550	000000			NOEXIT: 0
1962	010552	000000			TERM: 0
1963					
1964	010554	004537	011070		FND0: JSR RS,AMSG
1965	010550	011446			MO2 ;ERROR ASK AGAIN
1966	010552	012706	001100		FIND0T: MOV #STACK,SP
1967	010556	004537	011070		JSR RS,AMSG
1968	010572	011362			FINDTA: MO1
1969	010574	004737	011110		JSR PC,GETCHR
1970	010600	000770			BR FIND0T
1971					
1972	010602	105777	170522		IS: TSTB 2VTD5
1973	010606	100375			BPL IS
1974	010610	110077	170516		MOVB RO,2VTD8
1975	010614	042700	177640		BIC #177640,RO ;MASK
1976	010620	122700	000101		CMPS #A,RO ;TEST FOR NUMBER
1977	010624	101353			BHI FND0
1978	010626	122700	000116		CMPS #N,RO ;TEST FOR OTHERS
1979	010632	103750			BLO FND0
1980	010634	042700	177740		BIC #177740,RO ;MAKE 0-37
1981	010640	005300			DEC RO
1982	010642	110037	001102		MOVB RO,\$YSTNM ;LOAD THAT TEST #
1983	010646	006300			ASL RO
1984	010650	005760	010676		TST DSPCH(RO) ,TEST IF VALID
1985	010654	001737			BEQ FND0 ;BR IF NOT
1986	010656	016037	010676	00110C	MOV DSPCH(RO),SLP4OR ;LOAD RETURN ADDRESS
1987	010658	016037	010676	001110	MOV DSPCH(RO),SLPERR ;LOAD ERROR LOOP ADDRESS
1988	010672	000170	010676		JMP 2DSPCH(RO) ;GO TO THAT TEST
1989					
1990	010676	002066			DSPCH: TST1+2
1991	010700	002522			TST2+2
1992	010702	003150			TST3+2
1993	010704	003406			TST4+2
1994	010706	003644			TST5+2
1995	010710	004064			TST6+2
1996	010712	004302			TST7+2
1997	010714	004520			TST10+2
1998	010716	004740			TST11+2
1999	010720	005174			TST12+2
2000	010722	005430			TST13+2
2001	010724	005726			TST14+2
2002	010726	006224			TST15+2
2003	010730	000000			0
2004	010732	000000			0
2005	010734	000000			0
2006	010736	000000			0
2007	010740	000000			0

J04

MAINDEC-11-DZVTD-B MACY11 27(732) 25-SEP-76 10:10 PAGE 49
 DZVTD8.SRC END OF PASS ROUTINE

```

2008                                ;PROGRAM DELAY ROUTINE
2009
2010 010742 013737 001320 011020 DELAY: MOV      SUBTST,10$      ;LOAD COUNT
2011 010750 005037 011022                CLR      11$
2012 010754 005737 001322                TST      WFTST          ;TEST IF W.F. MODE
2013 010760 001404                BEQ      2$              ;BR IF NOT
2014 010762 006237 011020                ASR      10$              ;CHANGE DELAY TIMER
2015 010766 006237 011020                ASR      10$
2016 010772 032777 010000 170140 2$:    BIT      #BIT12,2$WR      ;TEST SR
2017 011000 001006                BNE      3$              ;BR IF SET
2018 011002 005337 011022                DEC      11$              ;DELAY
2019 011006 001371                BNE      2$
2020 011010 005337 011020                DEC      10$
2021 011014 100366                BPL      2$              ;DELAY
2022 011016 000207                3$:    RTS      PC              ;EXIT
2023
2024 011030 000002                10$:    2
2025 011022 000000                11$:    0
2026
2027 011024 013737 001320 011064 ADELAY: MOV      SUBTST,10$
2028 011032 005037 011066                CLR      11$
2029 011036 006237 011064                ASR      10$
2030 011042 006237 011064                ASR      10$
2031 011046 005337 011066                2$:    DEC      11$
2032 011052 001375                BNE      2$
2033 011054 005337 011064                DEC      10$
2034 011060 100372                BPL      2$
2035 011062 000207                RTS      PC
2036 011064 000000                10$:    0
2037 011066 000000                11$:    0

```

K04

MAINDEC-11-DZVTD-B MACY11 27(732) 25-SEP-76 10:10 PAGE 50
 DZVTD8.SRC END OF PASS ROUTINE

```

2038
2039 ;HEADER SUBROUTINE FOR VT-50
2040
2041 011070 012537 011100      MSG:  MOV      (RS)+,10$      ;GET POINTER
2042 011074 004537 011166      JSR      RS,MT08      ;MOVE TO EJFFER
2043 011100 000000      10$:  0
2044 011102 004737 010074      JSR      PC,XPRNT      ;DISPLAY IT
2045 011106 000205      RTS      RS      ;EXIT
2046
2047
2048 ;SUBROUTINE TO GET A CHARACTER FROM THE VT50
2049
2050 011110 013737 001316 011162 GETCHR: MOV      TIME0,TIME1      ;LOAD TIME COUNTER
2051 011116 005037 011164      CLR      TIME2
2052
2053 011122 105777 170176      1$:  TSTB      2VTIS      ;TEST INPUT STATUS
2054 011126 100005      BPL      2$      ;BR IF CLEARED
2055 011130 017700 170172      MOV      2VTIB,R0      ;READ A CHAR
2056 011134 062716 000002      ADD      2,(SP)      ;UPDATE RETURN
2057 011140 000207      RTS      PC      ;EXIT
2058
2059 011142 005337 011164      2$:  DEC      TIME2      ;DELAY
2060 011146 071365      BNE      1$
2061 011150 015337 011162      DEC      TIME1      ;FINISHED ?
2062 011154 100362      BPL      1$      ;LOOP TILL TIME EXPIRED
2063 011156 104002      ERROR      2      ;NO INPUT FLAG FROM DEVICE
2064 011160 000207      RTS      PC      ;EXIT
2065
2066 011162 000000      TIME1:  0
2067 011164 000000      TIME2:  0
2068
2069 ;MOVE TO THE OUTPUT BUFFER
2070
2071 011166 012500 015562      MT08:  MOV      (RS)+,R0      ;LOAD DEST.
2072 011170 012701      MOV      #BUFFER,R1      ;LOAD R1
2073 011174 112021      1$:  MOVB      (R0)+,(R1)+      ;LOAD BYTE
2074 011176 001376      BNE      1$      ;BR UNTIL DONE
2075 011200 000205      RTS      RS      ;EXIT
2076
2077 .SBTTL  ASCII MESSAGES

```

L04

MAINDEC-11-DZVTD-B MACY11 27(732) 25-SEP-76 10:10 PAGE 51
 DZVTD8.SRC ASCII MESSAGES

```

2078
2079 ;ASCII MESSAGES
2080
2081 011202 031033 005015 050042 PWRMSG: .ASCIZ <33><62><15><12>/'POWER'/<15><12>
2082 011210 053517 051105 006442
2083 011216 000012
2084 011220 031033 006415 046412 TITLE: .ASCIZ <33><62><15><15><12>/MAINDEC-11-DZVTD-B VT55 ACCEPTANCE TEST/<15><12>
2085 011226 044501 042116 041505
2086 011234 030455 026461 055104
2087 011242 052126 026504 020102
2088 011250 052126 032465 040440
2089 011256 041503 050105 040524
2090 011264 041516 020105 042524
2091 011272 052123 005015 000
2092 011277 033 015462 015510 M00: .ASCIZ <33><62><33><110><33><112>/LOOP ON TEST PATTERN LETTER (A THRU M) ? = /
2093 011304 046112 047517 020120
2094 011312 047117 052040 051505
2095 011320 020124 040520 052124
2096 011326 051105 020116 042514
2097 011334 052124 051105 024040
2098 011342 020101 044124 052522
2099 011350 046440 020051 020077
2100 011356 020075 000040
2101 011362 031033 044033 045033 M01: .ASCIZ <33><62><33><110><33><112>/START AT TEST PATTERN LETTER (A THRU M) ? =
2102 011370 052123 051101 020124
2103 011376 052101 052040 051505
2104 011404 020124 040520 052124
2105 011412 051105 020116 042514
2106 011420 052124 051105 024040
2107 011426 020101 044124 052522
2108 011434 046440 020051 020077
2109 011442 020075 000040
2110 011446 005015 052012 054522 M02: .ASCIZ <15><12><12>/TRY AGAIN /<15><12>
2111 011454 040440 040507 047111
2112 011462 006440 000012
2113 011466 005015 042012 053105 WHAT0: .ASCIZ <15><12><12>/DEVICE ADDRESS ? = /.200)
2114 011474 041511 020105 042101
2115 011502 051104 051505 020123
2116 011510 020077 020075 000200
2117 011516 051105 047522 020122 EM1: .ASCIZ /ERROR FLAG SET ON TRANSMITTER STATUS/
2118 011524 046106 043501 051440
2119 011532 052105 047440 020116
2120 011540 051124 047101 046523
2121 011546 052111 042524 020122
2122 011554 052123 052101 051525
2123 011562 000
2124 011563 116 020117 047111 EM2: .ASCIZ /NO INPUT FLAG DETECTED/
2125 011570 052520 020124 046106
2126 011576 043501 042040 052105
2127 011604 041505 042524 000104
2128 011612 047125 054105 042520 EM4: .ASCIZ /UNEXPECTED OR INCORRECT INPUT CHAR/
2129 011620 052103 042105 047440
2130 011626 020122 047111 047503
2131 011634 051122 041505 020124
2132 011642 047111 052520 020124
2133 011650 044103 051101 000

```

2134	011655	111	053116	046101	EM3:	.ASCIZ	/INVALID BUS ADDRESS, TRY AGAIN/
2135	011662	042111	041040	051525			
2136	011670	040440	042104	042522			
2137	011676	051523	020054	051124			
2138	011704	020131	043501	044501			
2139	011712	000116					
2140	011714	051105	050122	020103	DH1:	.ASCIZ	/ERRPC VTNOW TSTNUM/
2141	011722	020040	053040	047124			
2142	011730	053517	020040	020040			
2143	011736	051524	047124	046525			
2144	011744	000					
2145	011745	105	051122	041520	DH4:	.ASCIZ	/ERRPC VTNOW TSTNUM EXPCT RECV/
2146	011750	020040	053040	047124			
2147	011760	053517	020040	052040			
2148	011766	052123	052516	020115			
2149	011774	020040	054105	041520			
2150	012002	020124	020040	042522			
2151	012010	053103	000				
2152	012013	033	015462	015510	DAHL:	.ASCIZ	<33><62><33><110><33><112>/GROWING HORIZ. LINE/
2153	012020	043512	047522	044527			
2154	012026	043516	044040	051117			
2155	012034	055111	020056	044514			
2156	012042	042516	000				
2157	012045	033	015462	015510	DAVL:	.ASCIZ	<33><62><33><110><33><112>/GROWING VERT. LINE/
2158	012052	043512	047522	044527			
2159	012060	043516	053040	051105			
2160	012066	027124	046040	047111			
2161	012074	000105					
2162	012076	033	062	033	SHLO:	.BYTE	33,62,33,110,33,112
2163	012101	110	033	112			
2164	012104	044504	050123	040514		.ASCIZ	/DISPLAY STEPPING HORIZ. LINE ON GRAPH 0/
2165	012112	020131	052123	050105			
2166	012120	044520	043516	044040			
2167	012126	051117	055111	020056			
2168	012134	044514	042516	047440			
2169	012142	020116	051107	050101			
2170	012150	020110	000060				
2171	012154	033	062	033	SHL1:	.BYTE	33,62,33,110,33,112
2172	012157	110	033	112			
2173	012162	044504	050123	040514		.ASCIZ	/DISPLAY STEPPING HORIZ. LINE ON GRAPH 1/
2174	012170	020131	052123	050105			
2175	012176	044520	043516	044040			
2176	012204	051117	055111	020056			
2177	012212	044514	042516	047440			
2178	012220	020116	051107	050101			
2179	012226	020110	000061				
2180	012232	033	062	033	GROA1:	.BYTE	33,62,33,110,33,112
2181	012235	110	033	112			
2182	012240	044504	050123	040514		.ASCIZ	/DISPLAY DATA ON GRAPH 0 AND 1/
2183	012246	020131	040504	040514			
2184	012254	047440	020116	051107			
2185	012262	050101	020110	020060			
2186	012270	047101	020104	000061			
2187	012276	033	062	033	SHGL0:	.BYTE	33,62,33,110,33,112
2188	012301	110	033	112			
2189	012304	052123	050105	044520		.ASCIZ	/STEPPING HISTOGRAM LINE ON GRAPH 0/

MAINDEC-11-DZVTD-B MACY11 27(732) 25-SEP-76 10:10 PAGE 53
 DZVTDB.SRC ASCII MESSAGES

2190	012312	043516	044040	051511	
2191	012320	047524	051107	046501	
2192	012326	046040	047111	020105	
2193	012334	047117	043440	040522	
2194	012342	044120	030040	000	
2195	012347	033	062	033	SHGL1: .BYTE 33,62,33,110,33,112
2196	012352	110	033	112	
2197	012355	123	042524	050120	.ASCIZ /SLEPPING HISTOGRAM LINE ON GRAPH 1/
2198	012362	047111	020107	044510	
2199	012370	052123	043517	040522	
2200	012376	020115	044514	042516	
2201	012404	047440	020116	051107	
2202	012412	050101	020110	000061	
2203	012420	033	062	033	HGDA1: .BYTE 33,62,33,110,33,112
2204	012423	110	033	112	
2205	012426	044510	052123	043517	.ASCIZ /HISTOGRAM ON GRAPH 0 AND 1/
2206	012434	040522	020115	047117	
2207	012442	043440	040522	044120	
2208	012450	030040	040440	042116	
2209	012456	030440	000		
2210	012461	033	062	033	CURGR0: .BYTE 33,62,33,110,33,112
2211	012464	110	033	112	
2212	012467	103	051125	047523	.ASCIZ /CURSORS ON GRAPH 0/
2213	012474	051522	047440	020116	
2214	012502	051107	050101	020110	
2215	012510	000060			
2216	012512	033	062	033	CURGR1: .BYTE 33,62,33,110,33,112
2217	012515	110	033	112	
2218	012520	052503	051522	051117	.ASCIZ /CURSORS ON GRAPH 1/
2219	012526	020123	047117	043440	
2220	012534	040522	044120	030440	
2221	012542	000			
2222	012543	033	062	033	SCORD0: .BYTE 33,62,33,110,33,112
2223	012546	110	033	112	
2224	012551	123	040524	052122	.ASCIZ /STARTING COORDINATE TEST ON GRAPH 0/
2225	012556	047111	020107	047503	
2226	012564	051117	044504	040516	
2227	012572	042524	052040	051505	
2228	012600	020124	047117	043440	
2229	012606	040522	044120	030040	
2230	012614	000			
2231	012615	033	062	033	SCORD1: .BYTE 33,62,33,110,33,112
2232	012620	110	033	112	
2233	012623	123	040524	052122	.ASCIZ /STARTING COORDINATE TEST ON GRAPH 1/
2234	012630	047111	020107	047503	
2235	012636	051117	044504	040516	
2236	012644	042524	052040	051505	
2237	012652	020124	047117	043440	
2238	012660	040522	044120	030440	
2239	012666	000			
2240		012670			.EVEN
2241	012670	001116	001312	001314	DT1: SERRPC,VTNOW,TSTNUM,0
2242	012676	000000			
2243	012700	001116	001312	001314	DT4: SERRPC,VTNOW,TSTNUM,SGDDAT,SBDDAT,0
2244	012706	001124	001126	000000	
2245					

```

2246 .SBTTL TTY INPUT ROUTINE
2247
2248 ::*****
2249 .ENABL LSB
2250
2251 ::*****
2252 *SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
2253 *ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
2254 *SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
2255 *WHEN OPERATING IN TTY FLAG MODE.
2256
2257 012714 022737 000176 001140 $CKSMR: CMP      #SWREG,SWR      :: IS THE SOFT-SWR SELECTED?
2258 012722 001074          BNE      15$      :: BRANCH IF NO
2259 012724 105777 166214      TSTB     2$TKS      :: CHAR THERE?
2260 012730 100071          BPL      15$      :: IF NO, DON'T WAIT AROUND
2261 012732 117746 166210      MOVB     2$TKB,-(SP)  :: SAVE THE CHAR
2262 012736 042716 177600      BIC      8(C177,(SP)  :: STRIP-OFF THE ASCII
2263 012742 022726 0000J7      CMP      87(SP)+    :: IS IT A CONTROL G?
2264 012746 001062          BNE      15$      :: NO, RETURN TO USER
2265 012750 123727 001174 000001      CMPS     $AUTOB,#1  :: ARE WE RUNNING IN AUTO-MODE?
2266 012756 001456          BEQ      15$      :: BRANCH IF YES
2267
2268 012760 104400 013441      TYPE     ,SCNTLG      :: ECHO THE CONTROL-G (1G)
2269 012764 104400 013446      SGTSMR: TYPE     ,SMSWR      :: TYPE CURRENT CONTENTS
2270 012770 013746 000175      MOV      SWREG,-(SP)  :: SAVE SWREG FOR TYPEOUT
2271 012774 104401          TYPOC      :: GO TYPE--OCTAL ASCII(ALL DIGITS)
2272 012776 104400 013457      TYPE     ,SMNEW      :: PROMPT FOR NEW SWR
2273 013002 005046          CLR      -(SP)      :: CLEAR COUNTER
2274 013004 005046          CLR      -(SP)      :: THE NEW SWR
2275 013006 105777 166132      7$:      TSTB     2$TKS      :: CHAR THERE?
2276 013012 100375          BPL      7$          :: IF NOT TRY AGAIN
2277
2278 013014 117746 166126      MOVB     2$TKB,-(SP)  :: PICK UP CHAR
2279 013020 042716 177600      BIC      8(C177,(SP)  :: MAKE IT 7-BIT ASCII
2280
2281
2282 013024 021627 000025      9$:      CMP      (SP),#25      :: IS IT A CONTROL-U?
2283 013030 001005          BNE      10$      :: BRANCH IF NOT
2284 013032 104400 013434      TYPE     ,SCNTLU      :: YES, ECHO CONTROL-U (1U)
2285 013036 062706 000006      20$:     ADD      16,SP      :: IGNORE PREVIOUS INPUT
2286 013042 000757          BR        19$      :: LET'S TRY IT AGAIN
2287
2288
2289 013044 021627 000015      10$:     CMP      (SP),#15      :: IS IT A (CR)?
2290 013050 001022          BNE      16$      :: BRANCH IF NO
2291 013052 005766 000004      TST      4(SP)      :: YES, IS IT THE FIRST CHAR?
2292 013056 001403          BEQ      11$      :: BRANCH IF YES
2293 013060 016677 000002 166052      MOV      2(SP),2$SWR  :: SAVE NEW SWR
2294 013066 062706 000006      11$:     ADD      86,SP      :: CLEAR UP STACK
2295 013072 104400 001171      14$:     TYPE     ,SCRLF      :: ECHO (CR) AND (LF)
2296 013076 123727 001135 000001      CMPS     $INTAG,#1  :: RE-ENABLE TTY KBD INTERRUPTS?
2297 013104 001003          BNE      15$      :: BRANCH IF NOT
2298 013106 012777 000100 166030      MOV      8100,2$TKS  :: RE-ENABLE TTY KBD INTERRUPTS
2299 013114 000002          RTI          :: RETURN
2300 013116 004737 015046      16$:     JSR      PC,$TYPEC  :: ECHO CHAR
2301 013122 021627 000060      CMP      (SP),#60      :: CHAR ( 0?

```

2302	013126	002420		BLT	185	...	BRANCH IF YES
2303	013130	021627	000067	CMP	(SP), #67	...	CHAR > 7?
2304	013134	003015		BGT	185	...	BRANCH IF YES
2305	013136	042726	000060	BIC	#60, (SP)+	...	STRIP-OFF ASCII
2306	013142	005766	000002	TST	2(SP)	...	IS THIS THE FIRST CHAR
2307	013146	001403		BEO	175	...	BRANCH IF YES
2308	013150	006316		ASL	(SP)	...	NO, SHIFT PRESENT
2309	013152	006316		ASL	(SP)	...	CHAR OVER TO MAKE
2310	013154	006316		ASL	(SP)	...	ROOM FOR NEW ONE.
2311	013156	005266	000002	175: INC	2(SP)	...	KEEP COUNT OF CHAR
2312	013162	056616	177776	BIS	-2(SP), (SP)	...	SET IN NEW CHAR
2313	013166	000707		BR	75	...	GET THE NEXT ONE
2314	013170	104400	001170	185: TYPE	\$QUES	...	TYPE ?(CR)(LF)
2315	013174	000720		BR	205	...	SIMULATE CONTROL-U
2316				.DSABL	LSB		
2317							
2318							
2319							
2320							
2321							
2322							
2323							
2324							
2325							
2326							
2327	013176	011646		SROCHR: MOV	(SP), -(SP)	...	PUSH DOWN THE PC
2328	013200	016666	000004 000002	MOV	4(SP), 2(SP)	...	SAVE THE PS
2329	013206	105777	165732	15: TSTB	25TKS	...	WAIT FOR
2330	013212	100375		BPL	15	...	A CHARACTER
2331	013214	117766	165726 000004	MOVB	25TKB, 4(SP)	...	READ THE TTY
2332	013222	042766	177600 000004	BIC	#1C(177), 4(SP)	...	GET RID OF JUNK IF ANY
2333	013230	026627	000004 000023	CMP	4(SP), #23	...	IS IT A CONTROL-S?
2334	013236	001013		BNE	35	...	BRANCH IF NO
2335	013240	105777	165700	25: TSTB	25TKS	...	WAIT FOR A CHARACTER
2336	013244	100375		BPL	25	...	LOOP UNTIL ITS THERE
2337	013246	117746	165674	MOVB	25TKB, -(SP)	...	GET CHARACTER
2338	013250	042716	177600	BIC	#1C177, (SP)	...	MAKE IT 7-BIT ASCII
2339	013256	022527	000021	CMP	(SP)+, #21	...	IS IT A CONTROL-A?
2340	013262	001366		BNE	25	...	IF NOT DISCARD IT
2341	013264	000750		BR	15	...	YES, RESUME
2342	013266	026627	000004 000140	35: CMP	4(SP), #140	...	IS IT UPPER CASE?
2343	013274	002407		BLT	45	...	BRANCH IF YES
2344	013276	026627	000004 000175	CMP	4(SP), #175	...	IS IT A SPECIAL CHAR?
2345	013304	003003		BGT	45	...	BRANCH IF YES
2346	013306	042766	000040 000004	BIC	#40, 4(SP)	...	MAKE IT UPPER CASE
2347	013314	000002		45: RTI		...	GO BACK TO USER
2348							
2349							
2350							
2351							
2352							
2353							
2354							
2355	013316	010346		SROLIN: MOV	R3, -(SP)	...	SAVE R3
2356	013320	012703	013424	15: MOV	25TTYIN, R3	...	GET ADDRESS
2357	013324	022703	013434	25: CMP	25TTYIN, R3	...	BUFFER FULL?

 THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
 *CALL:
 * R0CHR
 * RETURN HERE
 *

 INPUT A SINGLE CHARACTER FROM THE TTY
 CHARACTER IS ON THE STACK
 WITH PARITY BIT STRIPPED OFF

 THIS ROUTINE WILL INPUT A STRING FROM THE TTY
 *CALL:
 * R0LIN
 * RETURN HERE
 *

 INPUT A STRING FROM THE TTY
 ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
 TERMINATOR WILL BE A BYTE OF ALL 0'S

2358	013330	101405			BLOS	48		:: BR IF YES
2359	013332	104406			ROCHR			:: GO READ ONE CHARACTER FROM THE TTY
2360	013334	112613			MOVB	(SP)+, (R3)		:: GET CHARACTER
2361	013336	122713	000177	108:	CMPB	#177, (R3)		:: IS IT A RUBOUT
2362	013345	001003			BNE	38		:: SKIP IF NOT
2363	013344	104400	001170	48:	TYPE	\$QUES		:: TYPE A '?'
2364	013350	000763			BR	18		:: CLEAR THE BUFFER AND LOOP
2365	013352	111337	013422	38:	MOVB	(R3), 98		:: ECHO THE CHARACTER
2366	013356	104400	013422		TYPE	98		
2367	013362	122723	000015		CMPB	#15, (R3)+		:: CHECK FOR RETURN
2368	013366	001356			BNE	28		:: LOOP IF NOT RETURN
2369	013370	105063	177777		CLRB	-1(R3)		:: CLEAR RETURN (THE 15)
2370	013374	104400	001172		TYPE	\$LF		:: TYPE A LINE FEED
2371	013400	012603			MOV	(SP)+, R3		:: RESTORE R3
2372	013402	011646			MOV	(SP) - (SP)		:: ADJUST THE STACK AND PUT ADDRESS OF THE
2373	013404	016666	000004	000002	MOV	4(SP), 2(SP)		:: FIRST ASCII CHARACTER ON IT
2374	013412	012766	013424	000004	MOV	#TTYIN, 4(SP)		
2375	013420	000002			RTI			:: RETURN
2376	013422	000		98:	.BYTE	0		:: STORAGE FOR ASCII CHAR. TO TYPE
2377	013423	000			.BYTE	0		:: TERMINATOR
2378	013424	000010			.BLKB	8		:: RESERVE 8 BYTES FOR TTY INPUT
2379	013434	052536	005015	000	\$CNTLU:	.ASCIZ /U/<15><12>		:: CONTROL "U"
2380	013441	136	006507	000012	\$CNTLG:	.ASCIZ /G/<15><12>		:: CONTROL "G"
2381	013446	005015	053523	020122	\$MSWR:	.ASCIZ <15><12>/SWR = /		
2382	013454	020075	000					
2383	013457	040	047040	053505	\$MNEW:	.ASCIZ / NEW = /		
2384	013464	036440	000040					

.SBTTL READ AN OCTAL NUMBER FROM THE TTY

```

*****
*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
*CHANGE IT TO BINARY.
*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
*FOLLOWED BY A CARRIAGE RETURN-LINE FEED, THE COMPLETE NUMBER MUST
*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.

```

*CALL:

```

*      RDOCT
*      RETURN HERE

```

```

::READ AN OCTAL NUMBER
::LOW ORDER BITS ARE ON TOP OF THE STACK
::HIGH ORDER BITS ARE IN $HIOCT

```

```

::PROVIDE SPACE FOR THE
::INPUT NUMBER
::PUSH R0 ON STACK
::PUSH R1 ON STACK
::PUSH R2 ON STACK
::READ AN ASCII LINE
::GET ADDRESS OF 1ST CHARACTER
::AND SAVE IT
::CLEAR DATA WORD

```

```

::PICKUP THIS CHARACTER
::IF ZERO GET OUT
::MAKE SURE THIS CHARACTER
::IS AN OCTAL DIGIT

```

;;#2

;;#4

;;#8

```

::STRIP THE ASCII JUNK
::ADD IN THIS DIGIT
::LOOP
::CLEAN TERMINATOR FROM STACK
::SAVE THE RESULT

```

```

::POP STACK INTO R2
::POP STACK INTO R1
::POP STACK INTO R0
::RETURN

```

```

::CLEAN PARTIAL FROM STACK
::SET A TERMINATOR
::TYPE UP THRU THE BAD CHAR.

```

```

::"? "CR" & "LF"
::TRY AGAIN
::HIGH ORDER BITS GO HERE

```

```

SRDOCT: MOV      (SP), -(SP)
        MOV      4(SP), 2(SP)
        MOV      R0, -(SP)
        MOV      R1, -(SP)
        MOV      R2, -(SP)
1$:      ROL IN
        MOV      (SP)+, R0
        MOV      R0, $S
        CLR      R1
        CLR      R2
2$:      MOV B    (R0)+, -(SP)
        BEQ      $S
        CMP B    #'0, (SP)
        BGT      $S
        CMP B    #'7, (SP)
        BGT      $S
        BIL T    $S
        RSL      R1
        RSL      R2
        RSL      R1
        RSL      R2
        RSL      R1
        RSL      R2
        BIC      #'C7, (SP)
        ADD      (SP)+, R1
        BR       2$
3$:      TST      (SP)+
        MOV      R1, 12(SP)
        MOV      R2, $HIOCT
        MOV      (SP)+, R2
        MOV      (SP)+, R1
        MOV      (SP)+, R0
        RTI
4$:      TST      (SP)+
        CLRB     (R0)
        TYPE
5$:      .WORD    0
        TYPE     $QUES
        BR       1$
$HIOCT: .WORD    0

```

000004 000002

013616

000060

000067

177770

000012

013626

001170

```

2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
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
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
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
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
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
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
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
2660
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
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
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
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
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
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
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
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471

```

```

013630 013630 105777 165304 MSCOPE: TSTB 2SWR ;TEST BIT 7
013634 013634 100003 BPL $SCOPE ;NOT SET
013636 013636 005737 010544 TST LOOP ;TEST LOOP
013642 013642 001070 BNE $OVER ;SET LOOP ON TEST
;SBTTL SCOPE HANDLER ROUTINE

;*****
;THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
;AND LOAD THE TEST NUMBER($TSTN) INTO THE DISPLAY REG.(DISPLAY<7:0>)
;AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
;SW14=1 LOOP ON TEST
;SW09=1 LOOP ON ERROR
;SW08=1 LOOP ON TEST IN SWR<7:0>
;CALL
; SCOPE ;;SCOPE=IOT

$SCOPE: CKSWR ;;TEST FOR CHANGE IN SOFT-SWR
CKSWR
15: BIT 2BIT14,2SWR ;;LOOP ON PRESENT TEST?
BNE $OVER ;;YES IF SW14=1
;*****START OF CODE FOR THE XOR TESTER*****
$XTSTR: BR 6S
MOV 2ERRVEC, -(SP) ;;IF RUNNING ON THE "XOR" TESTER CHANGE
MOV 2ERRVEC, 2ERRVEC THIS INSTRUCTION TO A "NOP" (NOP=240)
TST 2177060 SET FOR TIMEOUT
MOV (SP)+, 2ERRVEC TIME OUT ON XOR?
BR $SVLAD RESTORE THE ERROR VECTOR
5S: CMP (SP)+, (SP)+ GO TO THE NEXT TEST
MOV (SP)+, 2ERRVEC CLEAR THE STACK AFTER A TIME OUT
BR 7S RESTORE THE ERROR VECTOR
6S: ;;*****END OF CODE FOR THE XOR TESTER*****
BIT 2BIT08,2SWR ;;LOOP ON SPEC. TEST?
BEQ 2S BR IF NO
CMPB 2SWR, 2TSTN ON THE RIGHT TEST? SWR<7:0>
BEQ $OVER BR IF YES
2S: TSTB 2ERFLG HAS AN ERROR OCCURRED?
BEQ $SVLAD BR IF NO
BIT 2BIT09,2SWR LOOP ON ERROR?
BEQ 4S BR IF NO
7S: MOV 2LPERR, 2LPADR SET LOOP ADDRESS TO LAST SCOPE
BR $OVER
4S: CLRB 2ERFLG ZERO THE ERROR FLAG
$SVLAD: INCB 2TSTN COUNT TEST NUMBERS
MOV 2TSTN, 2TESTN SET TEST NUMBER IN APT MAILBOX
MOV (SP), 2LPADR SAVE SCOPE LOOP ADDRESS
MOV (SP), 2LPERR SAVE ERROR LOOP ADDRESS
CLR 2ESCAPE CLEAR THE ESCAPE FROM ERROR ADDRESS
MOV 21, 2SERMAX ONLY ALLOW ONE(1) ERROR ON NEXT TEST
$OVER: MOV 2TSTN, 2DISPLAY DISPLAY TEST NUMBER
MOV 2LPADR, (SP) FUDGE RETURN ADDRESS
RTI FIXES PS

```

.SBTTL APT COMMUNICATIONS ROUTINE

```

*****
014040 112737 000001 014304 SATY1: MOVB #1,SFFLG ;;TO REPORT FATAL ERROR
014046 112737 000001 014302 SATY3: MOVB #1,SFFLG ;;TO TYPE A MESSAGE
014054 000403 BR SATYC
014056 112737 000001 014304 SATY4: MOVB #1,SFFLG ;;TO ONLY REPORT FATAL ERROR
014064 SATYC:
014064 010046 MOV RO,-(SP) ;;PUSH RO ON STACK
014066 010146 MOV RI,-(SP) ;;PUSH RI ON STACK
014070 105737 014302 TSTB SMFLG ;;SHOULD TYPE A MESSAGE?
014074 001450 BEQ 5$ ;;IF NOT: BR
014076 122737 000001 001214 CNPB #APTENV,SENV ;;OPERATING UNDER APT?
014104 001031 BNE 3$ ;;IF NOT: BR
014106 132737 000107 001215 BITB #APTSPool,SENV ;;SHOULD SPOOL MESSAGES?
014114 001425 BEQ 3$ ;;IF NOT: BR
014116 017600 000004 MOV #24(SP),RO ;;GET MESSAGE ADDR.
014122 062766 000002 000004 ADD #2,4(SP) ;;BUMP RETURN ADDR.
014130 005737 001174 1$: TST $MSGTYPE ;;SEE IF DONE W/ LAST XMISSION?
014134 001375 BNE 1$ ;;IF NOT: WAIT
014136 010037 001210 MOV RO,$MSGAD ;;PUT ADDR IN MAILBOX
014142 105720 2$: TSTB (RO)+ ;;FIND END OF MESSAGE
014144 001376 BNE 2$
014146 163700 001210 SUB $MSGAD,RO ;;SUB START OF MESSAGE
014150 006200 ASR RO ;;GET MESSAGE LGTH IN WORDS
014154 010037 001212 MOV RO,$MSGLGTH ;;PUT LENGTH IN MAILBOX
014160 012737 000004 001174 MOV #4,$MSGTYPE ;;TELL APT TO TAKE MSG.
014166 000413 BR 5$
014170 017637 000004 014214 3$: MOV #24(SP),4$ ;;PUT MSG ADDR IN JSR LINKAGE
014176 062766 000002 000004 ADD #2,4(SP) ;;BUMP RETURN ADDRESS
014204 013746 177776 MOV 177776,-(SP) ;;PUSH 177776 ON STACK
014210 034737 014634 JSR PC,$TYPE ;;CALL TYPE MACRO
014214 000000 4$: .WORD 0
014216 5$:
014216 105737 014304 10$: TSTB SFFLG ;;SHOULD REPORT FATAL ERROR?
014222 001416 BEQ 12$ ;;IF NOT: BR
014224 005737 001214 TST SENV ;;RUNNING UNDER APT?
014230 001413 BEQ 12$ ;;IF NOT: BR
014232 005737 001174 11$: TST $MSGTYPE ;;FINISHED LAST MESSAGE?
014236 001375 BNE 11$ ;;IF NOT: WAIT
014240 017637 000004 001176 MOV #24(SP),SFATAL ;;GET ERROR #
014246 062766 000002 000004 ADD #2,4(SP) ;;BUMP RETURN ADDR.
014254 005237 001174 INC $MSGTYPE ;;TELL APT TO TAKE ERROR
014260 10 337 014304 12$: CLF3 $FLG ;;CLEAR FATAL FLAG
014264 10 337 014303 CLF3 $FLG ;;CLEAR LOG FLAG
014270 105037 014302 CLRB SMFLG ;;CLEAR MESSAGE FLAG
014274 012601 MOV (SP)+,R1 ;;POP STACK INTO R1
014276 012600 MOV (SP)+,RO ;;POP STACK INTO RO
014300 000207 RTS PC ;;RETURN
014302 000 SMFLG: .BYTE 0 ;;MESSG. FLAG
014303 000 $LFLG: .BYTE 0 ;;LOG FLAG
014304 000 SFFLG: .BYTE 0 ;;FATAL FLAG
014306 .EVEN
APTSIZE=200
APTENV=001
APTSPool=100

```

H05

MAINDEC-11-DZVTD-B MACY11 27(733) 25-SEP-76 10:10 PAGE 60
DZVTD8.54C APT COMMUNICATIONS ROUTINE

25-18

000040

APTC SUP=040

.SBTTL ERROR HANDLER ROUTINE

```

*****
*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
*AND GO TO SERRTYP ON ERROR
*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
*SW15=1      HALT ON ERROR
*SW13=1      INHIBIT ERROR TYPEOUTS
*SW09=1      LOOP ON ERROR
*CALL
*      ERROR      N      ;;ERROR=ENT AND N=ERROR ITEM NUMBER

```

```

SERROR:      CKSWR      ;;TEST FOR CHANGE IN SOFT-SWR
              MOV      STSTNM,TSTNM
              INCB      SERRFLG      ;;SET THE ERROR FLAG
              BEQ      7S      ;;DON'T LET THE FLAG GO TO ZERO
              MOV      STSTNM,2DISP1AY      ;;DISPLAY TEST NUMBER AND ERROR FLAG
              INC      SERRTTL      ;;INC THE ERROR COUNT
              MOV      (SP),SERRPC      ;;GET ADDRESS OF ERROR INSTRUCTION
              SUB      #2,SERRPC
              MOV      2(SERRPC),SITEMB      ;;STRIP AND SAVE THE ERROR ITEM CODE
              BIT      #BIT13,SWR      ;;SKIP TYPEOUT IF SET
              BNE      20S      ;;SKIP TYPEOUTS
              JSR      PC,SERRTYP      ;;GO TO USER ERROR ROUTINE
              TYPE      ,SCLF

20S:          CMP      #APTENV,SENV      ;;RUNNING IN APT MODE
              BNE      2S      ;;NO SKIP APT ERROR REPORT
              MOV      SITEMB,2IS      ;;SET ITEM NUMBER AS ERROR NUMBER
              JSR      PC,SATY4      ;;REPORT FATAL ERROR TO APT

21S:          .BYTE      0
              .BYTE      0

22S:          BR      22S      ;;APT ERROR LOOP
2S:          TST      2SWR
              BPL      3S
              HALT
              CKSWR
              BIT      #BIT09,2SWR      ;;TEST FOR CHANGE IN SOFT-SWR
              BEQ      4S      ;;LOOP ON ERROR SWITCH SET?
              BR      IF NO
              FUDGE RETURN FOR LOOPING
              TST      $ESCAPE      ;;CHECK FOR AN ESCAPE ADDRESS
              BEQ      5S      ;;BR IF NONE
              MOV      $ESCAPE,(SP)      ;;FUDGE RETURN ADDRESS FOR ESCAPE

5S:          CMP      #SENDAD,2#42      ;;ACT-11 AUTO-ACCEPT?
              BNE      6S      ;;BRANCH IF NO
              HALT
              YES

6S:          RTI      ;;RETURN

```

```

2549 014306
2550 014306 104405
2551 014310 113737 001102 001314
2552 014316 105237 001103
2553 014322 001775
2554 014324 013777 001102 164610
2555 014332 005237 001112
2556 014336 011637 001116
2557 014342 162737 000002 001116
2558 014350 117737 164542 001114
2559 014356 032777 020000 164554
2560 014364 001004
2561 014366 004737 014500
2562 014372 104400 001171
2563 014376 22737 000001 001214
2564 014404 001007
2565 014406 113737 001114 014420
2566 014414 004737 014056
2567 014420 000
2568 014421 000
2569 014422 000777
2570 014424 005777 164510
2571 014430 100002
2572 014432 000000
2573 014434 104405
2574 014436 032777 001000 164474
2575 014444 001402
2576 014446 013716 001110
2577 014452 005737 001166
2578 014456 001402
2579 014460 013716 001166
2580 014464
2581 014464 022737 006776 000042
2582 014472 001001
2583 014474 000000
2584 014476
2600 014476 000002

```

```

2601
2602
2603
2604
2605
2606
2607
2608
2609 014500
2610 014500 104400 001171
2611 014504 010046
2612 014506 005000
2613 014510 153700 001114
2614 014514 001004
2615
2616 014516 013746 001116
2617
2618 014522 104401
2619 014524 000426
2620 014526 005300
2621 014530 006300
2622 014532 006300
2623 014534 006300
2624 014536 062700 001256
2625 014542 012037 014552
2626 014546 001404
2627 014550 104400
2628 014552 000000
2629 014554 104400 001171
2630 014560 012037 014570
2631 014564 001404
2632 014566 104400
2633 014570 000000
2634 014572 104400 001171
2635 014576 011000
2636 014600 001004
2637 014602 012600
2638 014604 104400 001171
2639 014610 000207
2640 014612
2641 014612 01304
2642 014614 104401
2643 014616 005710
2644 014620 001770
2645 014622 104400 014630
2646 014626 000771
2647 014630 020040 000
2648 014634

```

.SBTTL ERROR MESSAGE TYPEOUT ROUTINE

```

*****
THIS ROUTINE USES THE "ITEM CONTROL BYTE" (SITEMB) TO DETERMINE WHICH
ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" (SERRTB),
AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.

```

SERRTYP:

```

      TYPE      SCRLF      "CARRIAGE RETURN" & "LINE FEED"
      MOV      RO,-(SP)      SAVE RO
      CLR      RO      PICKUP THE ITEM INDEX
      BISB     2(SITEMB,RO)
      BNE      IS
      MOV      SERRPC,-(SP)  IF ITEM NUMBER IS ZERO, JUST
                              TYPE THE PC OF THE ERROR
                              SAVE SERRPC FOR TYPEOUT
                              ERROR ADDRESS
                              GO TYPE--OCTAL ASCII(ALL DIGITS)
                              GET OUT
                              ADJUST THE INDEX SO THAT IT WILL
                              WORK FOR THE ERROR TABLE
      IS:      BR      6S
      DEC      RO
      ASL      RO
      ASL      RO
      ASL      RO
      ADD      8(SERRTB,RO)  FORM TABLE POINTER
      MOV      (RO)+,2S      PICKUP "ERROR MESSAGE" POINTER
      BEQ      3S      SKIP TYPEOUT IF NO POINTER
      TYPE     0      TYPE THE "ERROR MESSAGE"
      .WORD    0      "ERROR MESSAGE" POINTER GOES HERE
      TYPE     SCRLF      "CARRIAGE RETURN" & "LINE FEED"
      MOV      (RO)+,4S      PICKUP "DATA HEADER" POINTER
      BEQ      5S      SKIP TYPEOUT IF 0
      TYPE     0      TYPE THE "DATA HEADER"
      .WORD    0      "DATA HEADER" POINTER GOES HERE
      TYPE     SCRLF      "CARRIAGE RETURN" & "LINE FEED"
      MOV      (RO),RO      PICKUP "DATA TABLE" POINTER
      BNE      7S      GO TYPE THE DATA
      MOV      (SP)+,RO      RESTORE RO
      TYPE     SCRLF      "CARRIAGE RETURN" & "LINE FEED"
      RTS      PC      RETURN
      2S:      MOV      2(RO)+,-(SP)  SAVE 2(RO)+ FOR TYPEOUT
      TYPEOC   0      GO TYPE--OCTAL ASCII(ALL DIGITS)
      TST      (RO)      IS THERE ANOTHER NUMBER?
      BEQ      6S      BR IF NO
      TYPE     8S      TYPE TWO(2) SPACES
      BR      7S      LOOP
      4S:      .ASCIZ  / /      TWO(2) SPACES
      5S:      .EVEN
      6S:
      7S:

```

```

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

```

```

:CALL:
:*1) USING A TRAP INSTRUCTION
:*      TYPE      ,MESADR      ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
:*OR
:*      TYPE
:*      MESADR
:*

```

```

2649 .SBTTL TYPE ROUTINE
2650
2651 *****
2652 *ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
2653 *THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
2654 *NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CH
2655 *NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED
2656 *NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
2657
2658 *
2659 *CALL:
2660 *1) USING A TRAP INSTRUCTION
2661 *      TYPE      ,MESADR      ;;MESADR IS FIRST ADDRESS OF AN ASCIZ ST
2662 *OR
2663 *      TYPE
2664 *      MESADR
2665 *
2666
2667 014634 105737 001157 $TYPE: TSTB $TPFLG      ;; IS THERE A TERMINAL?
2668 014640 100002      BPL      IS      ;; BR IF YES
2669 014642 000000      HALT      ;; HALT HERE IF NO TERMINAL
2670 014644 000430      BR      3$      ;; LEAVE
2671 014646 010046      1$: MOV      RD,-(SP)      ;; SAVE RD
2672 014650 017600 000002      MOV      22(SP),RD      ;; GET ADDRESS OF ASCIZ STRING
2673 014654 122737 000001 001214      CMPB      #APTENV,$ENV      ;; RUNNING IN APT MODE
2674 014662 001011      BNE      62$      ;; NO,GO CHECK FOR APT CONSOLE
2675 014664 132737 000100 001215      BITB      #APTSPool,$ENVH      ;; SPOOL MESSAGE TO APT
2676 014672 001405      BEQ      62$      ;; NO,GO CHECK FOR CONSOLE
2677 014674 010037 014704      MOV      RD,61$      ;; SETUP MESSAGE ADDRESS FOR APT
2678 014700 004737 014046      JSR      PC,$ATY3      ;; SPOOL MESSAGE TO APT
2679 014704 000000      61$: .WORD      0      ;; MESSAGE ADDRESS
2680 014706 132737 000040 001215      62$: BITB      #APTCsup,$ENVH      ;; APT CONSOLE SUPPRESSED
2681 014714 001003      BNE      60$      ;; YES,SKIP TYPE OUT
2682 014716 112046      2$: MOVB      (RD)+,-(SP)      ;; PUSH CHARACTER TO BE TYPED ONTO STACK
2683 014720 001005      BNE      4$      ;; BR IF IT ISN'T THE TERMINATOR
2684 014722 005722      TST      (SP)+      ;; IF TERMINATOR POP IT OFF THE STACK
2685 014724 012700      60$: MOV      (SP)+,RD      ;; RESTORE RD
2686 014726 067716 000002      3$: ADD      #2,(SP)      ;; ADJUST RETURN PC
2687 014732 007002      RTI      ;; RETURN
2688 014734 122716 000011      4$: CMPB      #HT,(SP)      ;; BRANCH IF <HT>
2689 014740 001430      BEQ      8$      ;;
2690 014742 122716 000200      CMPB      #CRLF,(SP)      ;; BRANCH IF NOT <CRLF>
2691 014746 001006      BNE      5$      ;;
2692 014750 005726      TST      (SP)+      ;; POP <CR><LF> EQUIV
2693 014752 104400      TYPE      ;; TYPE A CR AND LF
2694 014754 001171      SCRLF
2695 014756 105037 015112      CLRB      $CHARCNT      ;; CLEAR CHARACTER COUNT
2696 014762 000755      BR      2$      ;; GET NEXT CHARACTER
2697 014764 004737 015046      5$: JSR      PC,$TYPEC      ;; GO TYPE THIS CHARACTER
2698 014770 123726 001156      6$: CMPB      $FILLC,(SP)+      ;; IS IT TIME FOR FILLER CHARS.?
2699 014774 001350      BNE      2$      ;; IF NO GO GET NEXT CHAR.
2700 014776 013746 001154      MOV      $NULL,-(SP)      ;; GET # OF FILLER CHARS. NEEDED
2701      AND THE NULL CHAR.
2702 015002 105366 000001      7$: DECB      1(SP)      ;; DOES A NULL NEED TO BE TYPED?
2703 015006 002770      BLT      6$      ;; BR IF NO--GO POP THE NULL OFF OF STACK
2704 015010 004737 015046      JSR      PC,$TYPEC      ;; GO TYPE A NULL

```

MAINDEC-11-DZVTD-B MACY11 27(732) 25-SEP-76 10:10 PAGE 64
 DZVTD8.SRC TYPE ROUTINE

```

2705 015014 105337 015112          DECB  $CHARCNT      ;; DO NOT COUNT AS A COUNT
2706 015020 000770          BR      7$                ;; LOOP
2707
2708          ;HORIZONTAL TAB PROCESSOR
2709
2710 015022 112716 000040      8$:  MOVB  #' (SP)          ;; REPLACE TAB WITH SPACE
2711 015026 004737 015046      9$:  JSR   PC,$TYPEPC      ;; TYPE A SPACE
2712 015032 132737 000007 015112  BITB  #',$CHARCNT      ;; BRANCH IF NOT AT
2713 015040 001372          BNE   9$                ;; TAB STOP
2714 015042 005726          TST   (SP)+              ;; POP SPACE OFF STACK
2715 015044 000724          BR      2$                ;; GET NEXT CHARACTER
2716 015046 105777 164076      $TYPEPC: TSTB  0$TPS        ;; WAIT UNTIL PRINTER IS READY
2717 015052 100375          BPL   $TYPEPC
2718 015054 116677 000002 164070  MOVB  2(SP),0$TPB      ;; LOAD CHAR TO BE TYPED INTO DATA REG.
2719 015062 122766 000015 000002  CMPB  #CR,2(SP)        ;; IS CHARACTER A CARRIAGE RETURN?
2720 015070 001003          BNE   1$                ;; BRANCH IF NO
2721 015072 105037 015112      CLRB  $CHARCNT          ;; YES--CLEAR CHARACTER COUNT
2722 015076 000406          BR      $TYPEPC          ;; EXIT
2723 015100 122766 000012 000002 1$:  CMPB  #LF,2(SP)      ;; IS CHARACTER A LINE FEED?
2724 015106 001402          BEQ   $TYPEPC          ;; BRANCH IF YES
2725 015110 105327          INCB  (PC)+              ;; COUNT THE CHARACTER
2726 015112 000000      $CHARCNT: .WORD 0                ;; CHARACTER COUNT STORAGE
2727 015114 000207      $TYPEPC: RTS   PC
2728

```

```

2729
2730 .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
2731
2732 *****
2733 *THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
2734 *OCTAL (ASCII) NUMBER AND TYPE IT.
2735 *STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
2736 *CALL:
2737 *      MOV      NUM, -(SP)      ;; NUMBER TO BE TYPED
2738 *      TYPOS    ;; CALL FOR TYPEOUT
2739 *      .BYTE    N              ;; N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
2740 *      .BYTE    M              ;; M=1 OR 0
2741 *                                  ;; 1=TYPE LEADING ZEROS
2742 *                                  ;; 0=SUPPRESS LEADING ZEROS
2743
2744 *STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
2745 *STYPOS OR STYPOC
2746 *CALL:
2747 *      MOV      NUM, -(SP)      ;; NUMBER TO BE TYPED
2748 *      TYPON    ;; CALL FOR TYPEOUT
2749
2750 *STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
2751 *CALL:
2752 *      MOV      NUM, -(SP)      ;; NUMBER TO BE TYPED
2753 *      TYPOC    ;; CALL FOR TYPEOUT
2754
2755 015116 017646 000000 015341 STYPOS: MOV      2(SP), -(SP)      ;; PICKUP THE MODE
2756 015122 116637 000001      MOV      1(SP), $OFILL      ;; LOAD ZERO FILL SWITCH
2757 015130 112637 015343      MOV      (SP)+, $OMODE+1      ;; NUMBER OF DIGITS TO TYPE
2758 015134 062716 000002      ADD      #2, (SP)          ;; ADJUST RETURN ADDRESS
2759 015140 000406      BR      STYPON
2760 015142 112737 000001 015341 STYPOC: MOV      #1, $OFILL      ;; SET THE ZERO FILL SWITCH
2761 015150 112737 000006 015343      MOV      #6, $OMODE+1      ;; SET FOR SIX(6) DIGITS
2762 015156 112737 000005 015340 STYPON: MOV      #5, $OCNT      ;; SET THE ITERATION COUNT
2763 015164 010346      MOV      R3, -(SP)          ;; SAVE R3
2764 015166 010446      MOV      R4, -(SP)          ;; SAVE R4
2765 015170 010546      MOV      R5, -(SP)          ;; SAVE R5
2766 015172 113704 015343      MOV      $OMODE+1, R4      ;; GET THE NUMBER OF DIGITS TO TYPE
2767 015176 005404      NEG      R4
2768 015200 062704 000006      ADD      #6, R4          ;; SUBTRACT IT FOR MAX. ALLOWED
2769 015204 110437 015342      MOV      R4, $OMODE      ;; SAVE IT FOR USE
2770 015210 113704 015341      MOV      $OFILL, R4      ;; GET THE ZERO FILL SWITCH
2771 015214 016605 000012      MOV      12(SP), R5      ;; PICKUP THE INPUT NUMBER
2772 015220 005003      CLR      R3          ;; CLEAR THE OUTPUT WORD
2773 015222 006105      1$: ROL      R5          ;; ROTATE MSB INTO "C"
2774 015224 000404      BR      3$
2775 015226 006105      2$: ROL      R5          ;; GO DO MSB
2776 015230 006105      ROL      R5          ;; FORM THIS DIGIT
2777 015232 006105      ROL      R5
2778 015234 010503      MOV      R5, R3
2779 015236 006103      3$: ROL      R3          ;; GET LSB OF THIS DIGIT
2780 015240 105337 015342      DECB      $OMODE      ;; TYPE THIS DIGIT?
2781 015244 100016      BPL      7$          ;; BR IF NO
2782 015246 042703 177770      BIC      #177770, R3      ;; GET RID OF JUNK
2783 015252 001002      BNE      4$          ;; TEST FOR 0
2784 015254 005704      TST      R4          ;; SUPPRESS THIS 0

```

MAIN EC-11-DZVTD-B MACY11 27(732) 25-SEP-76 10:10 PAGE 66
 DZVT08.SRC BINARY TO OCTAL (ASCII) AND TYPE

```

2785 015256 001403      BEQ      5$      ;;BR IF YES
2786 015260 005204      4$: INC      R4      ;;DON'T SUPPRESS ANYMORE 0'S
2787 015262 052703 000060  B1S      #'0,R3    ;;MAKE THIS DIGIT ASCII
2788 015266 052703 000040  5$: B1S      #'1,R3    ;;MAKE ASCII IF NOT ALREADY
2789 015272 110337 015336  MOV      R3,6$    ;;SAVE FOR TYPING
2790 015276 104400 015336  TYPE      8$      ;;GO TYPE THIS DIGIT
2791 015302 105337 015340  7$: DECB     $OCNT    ;;COUNT BY 1
2792 015306 003347      BGT      2$      ;;BR IF MORE TO DO
2793 015310 002402      BLT      6$      ;;BR IF DONE
2794 015312 005204      INC      R4      ;;INSURE LAST DIGIT ISN'T A BLANK
2795 015314 000744      BR       2$      ;;GO DO THE LAST DIGIT
2796 015316 012605  6$:  MOV      (SP)+,R5    ;;RESTORE R5
2797 015320 012604      MOV      (SP)+,R4    ;;RESTORE R4
2798 015322 012603      MOV      (SP)+,R3    ;;RESTORE R3
2799 015324 016666 000002 000004  MOV      2(SP),4(SP)  ;;SET THE STACK FOR RETURNING
2800 015332 012616      MOV      (SP)+,(SP)
2801 015334 000702      RTI      ;;RETURN
2802 015336      000      8$: .BYTE      0      ;;STORAGE FOR ASCII DIGIT
2803 015337      000      .BYTE      0      ;;TERMINATOR FOR TYPE ROUTINE
2804 015340      000      $OCNT: .BYTE      0      ;;OCTAL DIGIT COUNTER
2805 015341      000      $OFILL: .BYTE      0      ;;ZERO FILL SWITCH
2806 015342 000000      $OMODE: .WORD      0      ;;NUMBER OF DIGITS TO TYPE
2807      .SBTTL TRAP DECODER
2808
2809      ;*****
2810      ;THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
2811      ;AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
2812      ;OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
2813      ;GO TO THAT ROUTINE.
2814
2815 015344 010046      STRAP: MOV      RO,-(SP)    ;;SAVE RO
2816 015346 016600 000002  MOV      2(SP),RO    ;;GET TRAP ADDRESS
2817 015352 005740      TST      -(RO)        ;;BACKUP BY 2
2818 015354 111000      MOV      (RO),RO        ;;GET RIGHT BYTE OF TRAP
2819 015356 006300      ASL      RO            ;;POSITION FOR INDEXING
2820 015360 016000 015366  MOV      $TRPAD(RO),RO  ;;INDEX TO TABLE
2821 015364 000200      RTS      RO            ;;GO TO ROUTINE
2822
2823      .SBTTL TRAP TABLE
2824
2825      ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
2826      ;*BY THE "TRAP" INSTRUCTION.
2827
2828      : ROUTINE
2829      : -----
2830      $TRPAD:
2831 015366 014634      $TYPE      ;;CALL=TYPE      TRAP+0(104400) TTY TYPEOUT ROUTINE
2832 015370 015142      $TYPOC     ;;CALL=TYPOC     TRAP+1(104401) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
2833 015372 015116      $TYPOS     ;;CALL=TYPOS     TRAP+2(104402) TYPE OCTAL NUMBER (NO LEADING ZEROS)
2834 015374 015156      $TYPON     ;;CALL=TYPON     TRAP+3(104403) TYPE OCTAL NUMBER (AS PER LAST CALL)
2835
2836 015376 012764      $GTSWR      ;;CALL=GTSWR     TRAP+4(104404) GET SOFT-SWR SETTING
2837
2838 015400 012714      $CKSWR      ;;CALL=CKSWR     TRAP+5(104405) TEST FOR CHANGE IN SOFT-SWR
2839 015402 013176      $PDCHR      ;;CALL=PDCHR     TRAP+6(104406) TTY TYPEIN CHARACTER ROUTINE
2840 015404 013316      $RDLIN      ;;CALL=RDLIN     TRAP+7(104407) TTY TYPEIN STRING ROUTINE

```

```

2841 015406 013470          SROOCT ;;CALL=ROOCT TRAP+10(104410) READ AN OCTAL NUMBER FROM TTY
2842          .SBTTL POWER DOWN AND UP ROUTINES
2843          ;;*****
2844          :POWER DOWN ROUTINE
2845          SPWRDN: MOV      $SILLUP,2#PWRVEC ;;SET FOR FAST UP
2846                  MOV      $340,2#PWRVEC+2 ;;PRIO:7
2847                  MOV      R0,-(SP)      ;;PUSH R0 ON STACK
2848                  MOV      R1,-(SP)      ;;PUSH R1 ON STACK
2849                  MOV      R2,-(SP)      ;;PUSH R2 ON STACK
2850                  MOV      R3,-(SP)      ;;PUSH R3 ON STACK
2851                  MOV      R4,-(SP)      ;;PUSH R4 ON STACK
2852                  MOV      R5,-(SP)      ;;PUSH R5 ON STACK
2853                  MOV      2$WR,-(SP)    ;;PUSH 2$WR ON STACK
2854                  MOV      SP,$SAVR6    ;;SAVE SP
2855                  MOV      $PWRUP,2#PWRVEC ;;SET UP VECTOR
2856                  HALT
2857                  BR      .-2           ;;HANG UP
2858
2859          ;;*****
2860          :POWER UP ROUTINE
2861          SPWRUP: MOV      $SILLUP,2#PWRVEC ;;SET FOR FAST DOWN
2862                  MOV      $SAVR6,SP      ;;GET SP
2863                  CLZ      $SAVR6        ;;WAIT LOOP FOR THE TTY
2864                  IS:      INC      $SAVR6 ;;WAIT FOR THE INC
2865                  BIC      1$
2866                  MOV      (SP)+,2$WR    ;;POP STACK INTO 2$WR
2867                  MOV      (SP)+,R5      ;;POP STACK INTO R5
2868                  MOV      (SP)+,R4      ;;POP STACK INTO R4
2869                  MOV      (SP)+,R3      ;;POP STACK INTO R3
2870                  MOV      (SP)+,R2      ;;POP STACK INTO R2
2871                  MOV      (SP)+,R1      ;;POP STACK INTO R1
2872                  MOV      (SP)+,R0      ;;POP STACK INTO R0
2873                  MOV      $SPWRDN,2#PWRVEC ;;SET UP THE POWER DOWN VECTOR
2874                  MOV      $340,2#PWRVEC+2 ;;PRIO:7
2875                  TYPE    PWRMSG        ;;REPORT THE POWER FAILURE
2876                  SPWRMG: .WORD PWRMSG  ;;POWER FAIL MESSAGE POINTER
2877                  MOV      (PC)+,(SP)    ;;RESTART AT P'EGIN
2878                  SPWRAD: .WORD RBEGIN   ;;RESTART ADDRESS
2879                  RTI
2880                  SILLUP: HALT
2881                  BR      .-2
2882                  $SAVR6: 0
2883                  BUFFER: 0
2884                  .END
2885
2841 015410 012737 015554 000024
2842 015416 012737 000340 000026
2843 015424 010046
2844 015426 010146
2845 015430 010246
2846 015432 010346
2847 015434 010446
2848 015436 010546
2849 015440 017746 163474
2850 015442 010637 015560
2851 015450 012737 015462 000024
2852 015456 000000
2853 015460 000776
2862 015462 012737 015554 000024
2863 015470 013706 015560
2864 015474 005037 015560
2865 015500 005237 015560
2866 015504 001375
2867 015506 012677 163426
2868 015512 012605
2869 015514 012604
2870 015516 012503
2871 015520 012602
2872 015522 012601
2873 015524 012600
2874 015526 012737 015410 000024
2875 015534 012737 000340 000026
2876 015542 104400
2877 015544 011202
2878 015546 012716
2879 015550 001402
2880 015552 000002
2881 015554 000000
2882 015556 000776
2883 015560 000000
2884 015562 000000
2885 000001

```

Variable	Value	Variable	Value	Variable	Value	Variable	Value	Variable	Value	Variable	Value	Variable	Value	Variable	Value	Variable	Value	Variable	Value
ABASE	176500	537	646	687															
ACOM1	000000	646	689																
ACOM2	000000	646																	
ACFLOP	000000	646	661																
ADDLIN	002000	733	883	827	965	969	1338	1391	1588	1592	1605	1609	1611	1834					
ADDND	000000	646																	
ADDN1	000000	646																	
ADDN10	000000	646																	
ADDN11	000000	646																	
ADDN12	000000	646																	
ADDN13	000000	646																	
ADDN14	000000	646																	
ADDN15	000000	646																	
ADDN2	000000	646																	
ADDN3	000000	646																	
ADDN4	000000	646																	
ADDN5	000000	646																	
ADDN6	000000	646																	
ADDN7	000000	646																	
ADDN8	000000	646																	
ADDN9	000000	646																	
ADLAY	011024	2027																	
ADVCT	000000	646	652																
ADVM	000000	646	688																
ADW	000000	646	657																
ADWM	000000	646	658																
AFATL	000000	646	649																
AMDR1	000000	646	674																
AMDR2	000000	646	678																
AMDR3	000000	646	681																
AMDR4	000000	646	684																
AMMS1	000000	646	668																

BUSSTR	001336
CH R	= 104405
COPY	= 070135
OP	= 000015
ORLF	= 000200
CUMORD	012461
CUMRI	012512
CUMTOR	007574
DAML	012013
DAML	012045
DOISP	= 177570
DELAY	010742

[illegible]

[illegible]

		1623	1624	1625	1626	1627	1628	1629	1630	1631	1745	1750	1784	1789
LOOP	010544	181	1822											
LSC	= 000110	751	1940*	1947*	1959*	2440								
MAXMOZ	= 000354	729	1052	1132	1196	1243	1746	1785	1818					
MAXHZC	= 000400	730	687	920	1149	1291	1386	1436	1499	1762				
MAXVRT	= 001000	731		976	1060	1110	1204	1251	1440	1503	1834			
MOO	011277	1941	2042											
MO1	011352	1948	1958	2101										
MO2	011446	1965	2110											
MSCOPE	013630	830	2438											
MT08	011166	2042	2071											
NOEXIT	010550	1878	1893*	1924*	1930*	1952	1971*							
PC	=x000007	456	884*	908*	909*	91*		943*	966*	990*	991*	997*	1021*	1022*
		1038*	1054*	1059*	1067*	1070*	1083*	1104*	1109*	1117*	1120*	1140*	1145*	1170*
		1171*	1186*	1198*	1203*	1211*	1214*	1233*	1245*	1250*	1258*	1261*	1282*	1287*
		1312*	1313*	1329*	1334*	1359*	1364*	1375*	1387*	1412*	1416*	1417*	1432*	1433*
		1437*	1453*	1478*	1479*	1495*	1500*	1516*	1541*	1542*	1562*	1573*	1563*	1595*
		1606*	1612*	1633*	16*	1659*	1662*	1669*	1674	1748*	1753*	1787*	1792*	1820*
		1825*	1861*	1956*	196*	2022*	2035*	2044*	2057*	2064*	2300*	2523*	2540*	2575*
		2581*	2639*	2678*	2697*	2704*	2711*	2725*	2727*	2878				
		442												
PIRO	= 177772	536												
PIROVE	= 000240	459												
PR0	= 000300	460												
PR1	= 000440	461												
PR2	= 000100	462												
PR3	= 000140	463												
PR4	= 000200	464												
PR5	= 000240	465												
PR6	= 000300	466												
PR7	= 000340	467												
PS	= 177776	439	440											
PSM	= 177776	440												
PWRVSC	011202	2081	2877											
PWRVEC	= 000024	531	800*	801*	2846*	2847*	2856*	2862*	2874*	2875*				
RBEGIN	001402	560	781	2879										
ROCHR	= 104406	2359	2839											
ROLIN	= 104407	2404	2840											
ROOCT	= 104410	843	2841											
RESVEC	= 000010	526												
RSTR	002016	853	1654											
RSTRTA	002010	840	851	1675										
RO	=x000000	447	853*	854*	855*	856*	857*	859*	859*	860*	870*	871*	872*	873*
		874*	875*	876*	877*	878*	879*	880*	881*	882*	883*	889*	890*	891*
		892*	893*	894*	895*	896*	897*	898*	899*	900*	901*	902*	903*	904*
		905*	906*	907*	912*	913*	914*	918*	923*	924*	925*	926*	927*	928*
		929*	930*	931*	932*	933*	934*	935*	936*	937*	938*	939*	940*	941*
		952*	953*	954*	955*	956*	957*	958*	959*	960*	961*	962*	963*	964*
		967*	971*	972*	973*	974*	975*	976*	977*	978*	979*	980*	981*	982*
		983*	984*	985*	986*	987*	988*	989*	992*	993*	994*	998*	1002*	1003*
		1004*	1005*	1006*	1007*	1008*	1009*	1010*	1011*	1012*	1013*	1014*	1015*	1016*
		1017*	1018*	1019*	1020*	1030*	1031*	1032*	1033*	1034*	1035*	1036*	1037*	1041*
		1042*	1043*	1044*	1045*	1046*	1047*	1049*	1050*	1051*	1052*	1055*	1056*	1057*
		1061*	1066*	1081*	1082*	1083*	1084*	1085*	1086*	1087*	1088*	1092*	1093*	1094*
		1095*	1096*	1097*	1098*	1099*	1100*	1101*	1102*	1105*	1106*	1107*	1111*	1116*
		1132*	1133*	1134*	1135*	1136*	1137*	1138*	1139*	1151*	1152*	1153*	1154*	1155*

		1156*	1157*	1158*	1159*	1160*	1161*	1162*	1163*	1164*	1165*	1166*	1167*	1168*
		1169*	1178*	1179*	1180*	1181*	1182*	1183*	1184*	1185*	1190*	1191*	1192*	1193*
		1194*	1195*	1196*	1199*	1200*	1201*	1205*	1210*	1225*	1226*	1227*	1228*	1229*
		1230*	1231*	1232*	1237*	1238*	1239*	1240*	1241*	1242*	1243*	1246*	1247*	1248*
		1252*	1257*	1274*	1275*	1276*	1277*	1278*	1279*	1280*	1281*	1293*	1294*	1295*
		1296*	1297*	1298*	1299*	1300*	1301*	1302*	1303*	1304*	1305*	1306*	1307*	1308*
		1309*	1310*	1311*	1321*	1322*	1323*	1324*	1325*	1326*	1327*	1328*	1340*	1341*
		1342*	1343*	1344*	1345*	1346*	1347*	1348*	1349*	1350*	1351*	1352*	1353*	1354*
		1355*	1356*	1357*	1358*	1374*	1375*	1376*	1377*	1378*	1379*	1380*	1381*	1393*
		1394*	1395*	1396*	1397*	1398*	1399*	1400*	1401*	1402*	1403*	1404*	1405*	1406*
		1407*	1408*	1409*	1410*	1411*	1424*	1425*	1426*	1427*	1428*	1429*	1430*	1431*
		1441*	1443*	1444*	1458*	1459*	1460*	1461*	1462*	1463*	1464*	1465*	1466*	1467*
		1468*	1469*	1470*	1471*	1472*	1473*	1474*	1475*	1476*	1477*	1487*	1488*	1489*
		1490*	1491*	1492*	1493*	1494*	1504*	1506*	1507*	1521*	1522*	1523*	1524*	1525*
		1526*	1527*	1529*	1529*	1530*	1531*	1532*	1533*	1534*	1535*	1536*	1537*	1538*
		1539*	1540*	1552*	1553*	1554*	1555*	1556*	1557*	1558*	1559*	1560*	1561*	1563*
		1564*	1565*	1566*	1567*	1569*	1572*	1579*	1580*	1581*	1582*	1583*	1584*	1585*
		1586*	1587*	1590*	1594*	1600*	1601*	1602*	1603*	1604*	1607*	1613*	1614*	1615*
		1616*	1617*	1618*	1619*	1620*	1621*	1622*	1623*	1624*	1625*	1626*	1627*	1629*
		1629*	1630*	1631*	1632*	1666*	1669*	1703*	1739*	1740*	1741*	1742*	1743*	1745*
		1746*	1749*	1750*	1751*	1754*	1766*	1776*	1778*	1779*	1780*	1781*	1782*	1783*
		1784*	1785*	1789*	1789*	1790*	1793*	1794*	1795*	1806*	1807*	1808*	1809*	1810*
		1811*	1812*	1815*	1816*	1817*	1818*	1821*	1822*	1823*	1826*	1836*	1854*	1869*
		1878*	1879*	1879*	1879*	1879*	1879*	1879*	1879*	1879*	1879*	1879*	1879*	1879*
		1986*	1987*	1988*	2055*	2071*	2073*	2401*	2405*	2406*	2409*	2429*	2432*	2500*
		2508*	2512*	2513*	2515*	2516*	2517*	2539*	2611*	2612*	2613*	2620*	2621*	2622*
		2623*	2624*	2625*	2630*	2635*	2637*	2641*	2643*	2671*	2672*	2677*	2682*	2685*
		2815*	2816*	2817*	2818*	2819*	2820*	2821*	2848*	2873*				
R1	=X000001	448*	835*	918*	967*	998*	1005*	1061*	1105*	1111*	1199*	1205*	1246*	1252*
		1440*	1446*	1454*	1503*	1509*	1517*	1568*	1570*	1590*	1607*	1613*	1749*	1754*
		1777*	1786*	1793*	1795*	1813*	1814*	1816*	1821*	1826*	1848*	1849*	1850*	1851*
		1852*	1853*	1854*	1855*	1855*	1879*	1880*	1897*	1954*	2072*	2073*	2402*	2407*
R2	=X000002	2415*	2417*	2419*	2422*	2425*	2429*	2531*	2538*	2849*	2872*			
		449*	1551*	1574*	1645*	1846*	1848*	2403*	2408*	2416*	2418*	2420*	2426*	2427*
		2850*	2871*											
R3	=X000003	450*	1439*	1455*	1502*	1519*	2355*	2356*	2357*	2360*	2361*	2365*	2367*	2369*
		2371*	2763*	2772*	2778*	2779*	2782*	2787*	2788*	2789*	2798*	2851*	2870*	
R4	=X000004	451*	2764*	2766*	2767*	2768*	2769*	2770*	2784*	2786*	2794*	2797*	2852*	2869*
R5	=X000005	452*	867*	949*	1028*	1079*	1129*	1142*	1147*	1176*	1223*	1272*	1284*	1289*
		1319*	1331*	1336*	1361*	1372*	1374*	1379*	1413*	1422*	1434*	1448*	1485*	1497*
		1511*	1736*	1737*	1767*	1773*	1774*	1775*	1797*	1804*	1805*	1837*	1964*	1967*
		2041*	2042*	2045*	2071*	2075*	2765*	2771*	2773*	2775*	2776*	2777*	2778*	2796*
		2853*	2858*											
R6	=X000006	453*	455*	788*	789*	790*								
R7	=X000007	454*	456*											
SAVE4	001334	768*	777*	784*	831*									
SCOF00	012543	1423*	2222*											
SCOF01	012615	1486*	2231*											
SHALO	012276	1177*	2187*											
SAGL1	012347	1224*	2195*											
SHLO	012076	1029*	2162*											
SHL1	012154	1030*	2171*											
SHUFF	007776	634*	917*	946*	997*	1054*	1059*	1104*	1109*	1198*	1203*	1245*	1250*	1589*
SINEND	007241	1606*	1612*	1748*	1753*	1787*	1792*	1820*	1825*	1845*				
SP	=X000008	1441*	1443*	1504*	1506*	1732*								
		455*	770*	792*	809*	817*	821*	844*	1776*	1777*	1795*	1796*	1864*	1865*

	1954	1955	1956*	2056*	2260*	2261*	2262	2269*	2272*	2273*	2277*	2278*	2282
	2285*	2289	2291	2293	2294*	2301	2303	2305*	2306	2308*	2309*	2310*	2311*
	2312*	2327*	2328*	2331*	2332*	2333	2337*	2338*	2339	2342	2344	2346*	2355*
	2360	2371	2372*	2373*	2374*	2399*	2400*	2401*	2402*	2403*	2405	2409*	2411
	2413	2421*	2422	2424	2425*	2427	2428	2429	2431	2463*	2466	2468	2469
	2485	2486	2490*	2500*	2501*	2508	2509*	2520	2521*	2522*	2532	2533*	2538
	2539	2570	2591*	2594*	2611*	2616*	2637	2641*	2671*	2672	2682*	2684	2685
	2686*	2698	2690	2692	2698	2700*	2702*	2710*	2714	2718	2719	2723	2755*
	2756	2757	2758*	2763*	2764*	2765*	2771	2796	2797	2798	2799*	2800*	2815*
	2816	2848*	2849*	2850*	2851*	2852*	2853*	2854*	2855	2863*	2867	2868	2869
	2870	2871	2872	2873	2878*								
STACK =	001100												
STCOND =	007436												
STROUT =	177774												
SUBST =	001320												
SUR =	001140												
	430*	792	1966										
	1448	1511	1773*										
	441*												
	762*	2010	2027										
	624*	750	811*	813	819*	826*	837	839	1894	1888	1934	2016	2256
	2293*	2438	2458	2472	2474	2478	2573	2585	2589	2854	2867*		
	557*	819	837	2256	2269								
	494*												
	484*	494											
	483*	493											
	482*	492											
	481*	491											
	480*	490											
	479*	489											
	478*	488											
	477*	487											
	476*	486											
	475*	485											
	493*												
	474*												
	473*												
	472*												
	471*												
	470*												
	469*												
	492*												
	491*												
	490*												
	489*												
	488*												
	487*												
	486*												
	485*												
	527*												
TRITVE =	000014												
TERM	010552	1925	1950*	1962*									
TIME3	001316	1907	2050										
TIME1	011162	1907*	1914*	2050*	2061*	2066*							
TIME2	011164	1908*	1911*	2051*	2059*	2067*							
TITLE	011220	834	2084*										
TITST	001714	831*											
TKVEC =	000760	534*											
TPVEC =	000034	535*											
TRPVE =	000034	533*	798*	799*									
TRTVEC =	000014	528*											
TSTNUM	071314	759*	2241	2243	2565*								

TST1	002064	775	781	862	866#	1990													
TST10	004516	1262	1271#	1997															
TST11	004736	1318#	1998																
TST12	005172	1371#	1999																
TST13	005426	1421#	2000																
TST14	005724	1484#	2001																
TST15	006222	1548#	2002																
TST2	002520	948#	1991																
TST3	003146	1027#	1992																
TST4	003404	1071	1078#	1993															
TST5	003642	1121	1138#	1994															
TST6	074062	1175#	1995																
TST7	004300	1215	1222#	1996															
TYPE =	104400	833	841	1665	2267	2268	2271	2284	2295	2314	2363	2366	2370	2433					
		2435	2576	2610	2627	2629	2632	2634	2638	2645	2693	2790	2831#	2876					
		2270	2618	2642	2832#														
TYP0C =	174401	2834#																	
TYP0N =	104403	2833#																	
TYP0S =	104402	1142	1147	1284	1289	1331	1384	1434	1497	1736#									
UPDOWN	007242	755#	1920	2055															
VTI9	001326	764#	853	1898	1909	2053													
VTIS	001324	758#	851*	854	855	857	859	1651	1653*	2241	2243								
VTNOM	001312	767#	1897*	1974*															
VT08	001332	766#	1870	1972															
VTOS	001330	763#	776*	783*	2012														
MFTEST	001322	842	2113#																
MHATO	011466	775*	781*	861	862#														
MHERE	002052	763#	1928																
XOFF =	000023	763#	1922	1931	1951														
XON =	000021	908	942	990	1021	1038	1067	1089	1117	1140	1145	1170	1186	1211					
XPRNT	010074	1233	1258	1282	1287	1312	1329	1334	1359	1364	1382	1387	1412	1416					
		1432	1437	1453	1478	1495	1500	1516	1541	1562	1573	1595	1633	1664#					
		2044																	
SAPTHD	001000	584	590#																
\$HSTAT=	***** U	2526	2541																
\$ATYC	014064	2497	2499#		</														

MAINDEC-11-DZVTD-B MACY11 27(732) 25-SEP-76 10:10 PAGE 76
 DZVTDB.SRC CROSS REFERENCE TABLE -- USER SYMBOLS

SDOAGN	007006	1661	1667	1673#					
SENDAD	006776	570	1669#	2596					
SENDCT	006756	802	1663#						
SENDMG	007015	1665	1677#						
SENULL	007012	1676#							
SENV	001214	657#	2504	2528	2578	2673			
SENVH	001215	658#	824	2506	2675	2680			
SEOP	006674	1647#							
SEOPCT	006750	802#	1660#	1664					
SEFLG	001103	607#	2447	2476	2482#	2492	2566#	2601	
SERMAX	001115	613#	804#	2488#	2492				
SERORR	014306	796	2563#						
SERAPC	001116	614#	2241	2243	2570#	2571#	2572	2601	2616
SERRTB	001256	706#	2624						
SERTY	014500	2575	2609#						
SERTTL	001112	611#	2569#	2601					
SESCAP	001166	638#	803#	2487#	2592	2594	2601		
SETABL	001214	656#							
SETEND	001256	596	690#						
SFATAL	001176	649#	2532#						
SFFLG	014304	2495#	2498#	2526	2535#	2543#			
SFILLC	001156	632#	698	2729					
SFILLS	001155	631#	2729						
SGOADR	001120	615#							
SGODAT	001124	617#	1902#	1931#	1933#	1951#	2243		
SGET42	006766	1666#							
SGTSMR	012764	2268#	2836						
SHD	000300	427							
SHIBTS	001030	591#							
SHIOCT	013626	2426#	2437#						
SICNT	001104	608#							
SILLUP	015554	2846	2862	2881#					
SINTAG	001135	622#	2296	2385					
SITEMB	001114	612#	2572#	2580	2601	2613			
SLF	001172	641#	2370	2379	2438	2601	2729		
SLFLG	014303	2536#	2512#						
SLPADR	001136	609#	805#	1966#	2480#	2485#	2490	2492	2492
SLPERR	001110	610#	806#	1987#	2480	2486#	2492	2591	
SMAOR1	001226	674#							
SMAOR2	001232	678#							
SMAOR3	001236	681#							
SMAOR4	001242	684#							
SMAIL	001174	592	596	647#	823	2484	2578	2673	
SMAIS1	001224	668#							
SMAIS2	001230	676#							
SMAIS3	001234	679#							
SMAIS4	001240	682#							
SMAORR	001002	592#							
SFLG	014302	2496#	2502	2537#	2541#				
SNEW	013457	2271	2383#						
SMSGAD	001210	654#	2512#	2515					
SMSGLC	001212	655#	2517#						
SMSGTY	001174	642#	2510	2518#	2530	2534#			
SMSMR	013446	2268	2381#						
SMTYP1	001225	669#							
SMTYP2	001231	677#							

SHTYP3	001235	680#												
SHTYP4	001241	683#												
\$NULL	001154	630#	2700	2729										
\$NWTST=	000001	863#	945#	1024#	1075#	1125#	1172#	1219#	1268#	1315#	1368#	1418#	1481#	1545#
\$OCNT	015340	2762*	2791*	2804#										
\$OMODE	015342	2757*	2761*	2766	2769*	2780*	2806#							
\$OVER	014024	2441	2459	2475	2481	2489#								
\$PASS	001202	651#	823*	1657*	1658*	1676	1890							
\$PASTH	001006	594#												
\$PIRAD	015550	2879#												
\$PIRON	015410	800	2846#	2874										
\$PIRMG	015544	2877#												
\$PIRUP	015462	2856	2862#											
\$QUES	001170	639#	2314	2363	2379	2435	2438	2601	2729					
\$ROCHR	013176	2327#	2833											
\$RODEC=	***** U	2842												
\$ROLIN	013316	2355#	2340											
\$ROOCT	013470	2399#	2841											
\$ROSZ =	000010	2348#												
\$REGAD	001160	634#												
\$REGO	001162	636#												
\$REGI	001164	637#												
\$ATNAD	007010	1675#												
\$R2A =	***** U	2642												
\$SAVRE=	***** U	2842												
\$SAVR6	015560	2855*	2863	2864*	2865*	2883#								
\$SCOPE	013644	794	2439	2455#										
\$SETUP=	000137	708#	793	794	796	798	800	802	803	805	1656	2251	2385	2456
		2564	2588	2596										
\$STUP =	177777	708#												
\$SVLAD	013770	2467	2477	2483#										
\$SVPC =	000214	568#	573											
\$SWR =	161400	417#	427	542	543	544	545	546	547	548	638	639	803	805
		806	867	949	1028	1079	1129	1176	1223	1272	1319	1372	1422	1485
		1549	1642	1657	1668	1674	1676	2448	2449	2450	2451	2458	2470	2472
		2473	2476	2477	2478	2483	2486	2489	2492	2556	2557	2558	2559	2569
		2573	2585	2589	2601	2880								
\$SCAPEG	001216	659#	826											
\$SWMK=	000000	548	549	2451	2452	2474								
\$TESTN	001200	650#	2484*											
\$TKB	001146	627#	2249	2260	2277	2331	2337							
\$TKS	001144	626#	2249	2258	2274	2298*	2329	2335						
\$TN =	000016	417#	427	863	867#	945	949#	1024	1028#	1071				

[illegible]

ANYMOR	1544#	1648													
CLRDAT	754#	873	955	1033	1084	1377	1427	1490	1555						
CLRUNT	754#	1030	1081	1374	1424	1487									
COMMEN	1#	537#													
COPYSC	754#	889	923	971	1002	1151	1293	1340	1393	1459	1522	1614			
ENDCON	1#	537#													
ERROR	431#	771	1944	2063											
ESCAPE	1#	537#													
GETPRI	1#	537#													
GETSWR	1#	537#													
MULT	1#	537#													
NEWTST	1#	537#	863	945	1024	1075	1125	1172	1219	1268	1315	1368	1418	1481	1545
POP	1#	537#	2427	2538	2539	2867	2868								
PUSH	1#	537#	2401	2499	2501	2522	2848	2854							
REPORT	1#	537#													
SCOPE	432#	866	948	1027	1078	1128	1175	1222	1271	1318	1371	1421	1484	1548	1648
SETPRI	1#	537#													
SETTRA	2823#	2832	2833	2834	2836	2838	2839	2840	2841						
SETUP	1#	537#	786												
SKIP	1#	537#	1071	1121	1215	1262	1937								
SLASH	1#	537#													
SPACE	537#														
STARS	1#	537#	566	577	579	586	599	642	645	863	865	945	947	1024	1026
	1075	1077	1125	1127	1172	1174	1219	1221	1268	1270	1315	1317	1368	1370	1418
	1420	1481	1483	1545	1547	1639	2248	2251	2319	2348	2387	2444	2494	2552	2604
	2652	2732	2809	2844	2860										
7WRSU	1#	537#	807#												
7RMTIP	2823#														
TYPBIN	1#	537#													
TYPDEC	1#	537#													
TYPNAM	1#	537#													
TYPNUM	1#	537#													
TYPOCS	1#	537#													
TYP OCT	1#	537#	2269	2616	2640										
TYPTXT	1#	537#													
\$\$SCHRC	597#	636	637												
\$\$SCHTH	597#														
\$\$ESCA	1#	537#													
\$\$NEWT	1#	537#	863	945	1024	1075	1125	1172	1219	1268	1315	1368	1418	1481	1545
\$\$SET	2823#	2832	2833	2834	2836	2838	2839	2840	2841						
\$\$SETH	823#														
\$\$SKIP	1#	537#	1071	1121	1215	1262									
.EQUAT	1#	417#	427												
.HEADE	1#	417#													
.KT11	1#														
.SETUP	1#	417#	708												
.SURHI	1#	417#	538												
.SURLO	549#														
.SACT1	1#	417#	564												
.SAPT8	1#	643#													
.SAPTH	1#	417#	575												
.SAPTY	1#	417#	2492												
.SASTA	1#														
.SCATC	1#	417#	549												
.SCHTA	1#	417#	597												
.SD82D	1#														

.SOB20	18		
.SOIV	18		
.SEOP	18	4178	1637
.SERRO	18	4178	25 0
.SERRT	18	4178	2602
.SMULT	18		
.SPARM	4178		
.SPOWE	18	4178	2842
.SRAND	18		
.SRODE	18		
.SROOC	18	4178	2385
.SREAD	18	4178	2246
.SR2AZ	18		
.SSAVE	18	4178	
.SSB20	18		
.SSB20	18		
.SSCOP	18	4178	2442
.SSIZE	18		
.SSPAC	4178		
.SSUPR	18		
.SSMCO	4178		
.STRAP	18	4178	2807
.STYPB	18		
.STYPD	18		
.STYPE	18	4178	2650
.STYPO	18	4178	2730
.S4OCA	18		
.1170	18		

B07

 MACDEC-11-DZVTD-B MACY11 27(732) 25-SEP-76 10:10 PAGE 83
 DZVTD8.SRC CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

ADD	856	858	860	1591	1608	1653	2056	2285	2294	2422	2509	2521	2533	2524	2686
ASL	2758	2768													
ASR	814	1983	2308	2309	2310	2415	2417	2419	2621	2622	2623	2819			
BCQ	2014	2015	2029	2030	2516										
	815	838	1445	1506	1650	1652	1667	1759	1831	1858	1879	1885	1889	1926	1937
	1925	2013	2265	2292	2307	2410	2473	2475	2477	2479	2503	2507	2527	2529	2567
BGT	2590	2593	2626	2631	2644	2676	2689	2724	2785						
BHI	1661	2304	2345	2412	2792										
BIC	846	1977													
	1658	1758	1830	1847	1855	1891	1921	1975	1980	2261	2278	2305	2332	2339	2346
BIS	2421	2782													
BISB	1856	1859	2312	2787	2788										
BIT	1743	1783	1811	1816	2613										
BITB	1857	1864	1888	2016	2458	2472	2478	2573	2587						
BLO	824	2506	2675	2680	2712										
BLOS	1979														
BLT	2358														
BMI	2302	2343	2414	2703	2793										
BNE	1910														
	791	814	832	888	921	970	1000	1063	1069	1113	1119	1207	1213	1254	1260
	1457	1520	1571	1575	1763	1835	1881	1883	1892	1901	1912	1915	1923	1929	1939
	1953	2017	2019	2032	2060	2074	2257	2263	2283	2290	2297	2334	2340	2362	2368
	2441	2459	2505	2511	2514	2531	2574	2579	2597	2614	2636	2674	2691	2683	2691
BPL	2699	2713	2720	2783	2866										
	840	1593	1610	1756	1765	1828	1871	1899	1935	1973	2021	2034	2054	2062	2259
BR	2275	2330	2336	2439	2586	2668	2717	2781							
	790	762	816	1071	1121	1215	1262	1442	1455	1505	1518	1760	1832	1887	1895
	1903	1916	1927	1932	1970	2286	2313	2315	2341	2364	2423	2436	2461	2467	2470
	2491	2497	2519	2584	2619	2646	2670	2696	2706	2715	2722	2759	2774	2795	2858
CLR	2632														
	776	777	789	803	823	829	848	850	1656	1747	1819	1866	1867	1868	1896
	1902	1908	1924	1933	1947	2011	2028	2051	2272	2273	2407	2408	2497	2612	2772
CLRE	2854														
	907	941	989	1020	1037	1066	1088	1116	1139	1169	1185	1210	1232	1257	1281
	1311	1328	1358	1381	1411	1431	1477	1434	1540	1561	1572	1594	1632	1766	1794
CMP	1836	2369	2432	2482	2535	2536	2537	2695	2721						
	770	790	813	837	845	887	920	969	1592	1609	1651	1762	1834	1922	1928
CMPB	1936	1938	2256	2262	2282	2289	2301	2303	2333	2339	2342	2344	2357	2468	2596
	1830	1832	1976	1978	2264	2296	2361	2367	2411	2413	2474	2504	2578	2673	2688
DEC	2690	2698	2719	2723											
	968	999	1062	1112	1206	1253	1454	1456	1517	1519	1570	1574	1659	1757	1764
DECB	1829	1911	1914	1981	2018	2020	2031	2033	2059	2061	2620				
EMT	2702	2705	2780	2791											
HALT	431														
INC	555	2587	2598	2669	2857	2881									
INCB	886	919	1657	1761	1833	1894	2311	2534	2569	2786	2794	2865			
IOT	1823	1930	2483	2566	2725										
JMP	432														
JSR	559	560	561	861	1654	1674	1942	1945	1949	1988					
	867	884	908	909	917	942	943	949	966	990	991	997	1021	1022	1028
	1038	1054	1059	1067	1070	1079	1089	1104	1109	1117	1120	1129	1140	1142	1145
	1147	1170	1171	1176	1186	1198	1203	1211	1214	1223	1233	1245	1250	1258	1261
	1272	1282	1284	1287	1289	1312	1313	1319	1329	1331	1334	1336	1359	1361	1364
	1365	1372	1382	1384	1387	1389	1412	1413	1416	1417	1422	1432	1434	1437	1448
	1453	1478	1479	1485	1495	1497	1500	1511	1516	1541	1542	1562	1573	1589	1595
	1606	1612	1633	1635	1669	1748	1753	1787	1792	1820	1825	1964	1967	1969	2042

MOV8

NEG
NOP
RESET
ROL
ROR
RTI
RTS
SUB
TRAP

2044	2300	2523	2575	2581	2678	2697	2704	2711	795	796	797	798	799	800
775	778	779	781	783	794	788	792	794	819	820	821	826	829	830
801	802	805	806	809	810	811	812	817	883	885	912	916	918	930
844	849	851	853	854	855	857	859	870	1055	1058	1060	1061	1081	1091
945	967	982	996	998	1000	1040	1041	1053	1190	1197	1199	1202	1204	1205
1092	1103	1105	1108	1110	1111	1132	1178	1188	1321	1374	1424	1439	1440	1441
1225	1235	1237	1244	1246	1249	1251	1252	1274	1521	1551	1552	1563	1568	1569
1443	1446	1458	1487	1502	1503	1504	1506	1509	1666	1666	1736	1738	1744	1749
1579	1588	1590	1600	1605	1607	1611	1613	1662	1798	1798	1793	1795	1796	1804
1752	1754	1773	1774	1775	1776	1777	1778	1786	1860	1864	1865	1869	1890	1907
1805	1806	1813	1821	1824	1826	1845	1846	1848	1966	1996	1987	2010	2027	2041
1920	1931	1940	1941	1946	1950	1951	1954	1955	2355	2356	2371	2372	2373	2374
2050	2055	2071	2072	2073	2075	2076	2077	2078	2427	2428	2429	2453	2464	2466
2236	2237	2240	2241	2242	2243	2244	2245	2246	2512	2517	2518	2520	2522	2532
2399	2400	2401	2402	2403	2404	2405	2406	2407	2630	2635	2637	2641	2671	2672
2567	2568	2569	2570	2571	2572	2573	2574	2575	2796	2797	2798	2799	2800	2815
2616	2617	2618	2619	2620	2621	2622	2623	2624	2853	2854	2855	2856	2862	2863
2864	2865	2866	2867	2868	2869	2870	2871	2872	879	880	881	882	899	890
2873	2874	2875	2876	2877	2878	2879	2880	2881	900	901	902	903	904	905
2882	2883	2884	2885	2886	2887	2888	2889	2890	929	930	931	932	933	934
2891	2892	2893	2894	2895	2896	2897	2898	2899	955	957	958	959	960	961
2900	2901	2902	2903	2904	2905	2906	2907	2908	976	978	979	980	981	982
2909	2910	2911	2912	2913	2914	2915	2916	2917	1002	1003	1005	1006	1007	1008
2918	2919	2920	2921	2922	2923	2924	2925	2926	1018	1019	1031	1032	1033	1034
2927	2928	2929	2930	2931	2932	2933	2934	2935	1049	1050	1052	1056	1057	1082
2936	2937	2938	2939	2940	2941	2942	2943	2944	1096	1097	1098	1100	1101	1102
2945	2946	2947	2948	2949	2950	2951	2952	2953	1151	1152	1153	1155	1156	1157
2954	2955	2956	2957	2958	2959	2960	2961	2962	1166	1167	1168	1179	1181	1182
2963	2964	2965	2966	2967	2968	2969	2970	2971	1200	1201	1226	1180	1229	1230
2972	2973	2974	2975	2976	2977	2978	2979	2980	1248	1275	1276	1227	1279	1280
2981	2982	2983	2984	2985	2986	2987	2988	2989	1301	1302	1303	1305	1306	1307
2990	2991	2992	2993	2994	2995	2996	2997	2998	1327	1340	1341	1343	1344	1345
2999	3000	3001	3002	3003	3004	3005	3006	3007	1354	1355	1357	1375	1376	1377
3008	3009	3010	3011	3012	3013	3014	3015	3016	1398	1399	1401	1402	1403	1404
3017	3018	3019	3020	3021	3022	3023	3024	3025	1427	1428	1430	1444	1459	1460
3026	3027	3028	3029	3030	3031	3032	3033	3034	1469	1470	1472	1473	1474	1475
3035	3036	3037	3038	3039	3040	3041	3042	3043	1523	1524	1525	1526	1527	1528
3044	3045	3046	3047	3048	3049	3050	3051	3052	1537	1538	1539	1554	1555	1556
3053	3054	3055	3056	3057	3058	3059	3060	3061	1580	1581	1582	1584	1585	1586
3062	3063	3064	3065	3066	3067	3068	3069	3070	1617	1618	1619	1621	1622	1623
3071	3072	3073	3074	3075	3076	3077	3078	3079	1631	1639	1640	1641	1642	1643
3080	3081	3082	3083	3084	3085	3086	3087	3088	1679	1680	1681	1682	1683	1684
3089	3090	3091	3092	3093	3094	3095	3096	3097	1739	1740	1741	1742	1743	1744
3098	3099	3100	3101	3102	3103	3104	3105	3106	1790	1807	1808	1809	1810	1811
3107	3108	3109	3110	3111	3112	3113	3114	3115	1982	2073	2260	2331	2337	2360
3116	3117	3118	3119	3120	3121	3122	3123	3124	2572	2580	2682	2710	2756	2757
3125	3126	3127	3128	3129	3130	3131	3132	3133						
3134	3135	3136	3137	3138	3139	3140	3141	3142	1875	1876	1877			
3143	3144	3145	3146	3147	3148	3149	3150	3151						
3152	3153	3154	3155	3156	3157	3158	3159	3160						
3161	3162	3163	3164	3165	3166	3167	3168	3169						
3170	3171	3172	3173	3174	3175	3176	3177	3178						
3179	3180	3181	3182	3183	3184	3185	3186	3187						
3188	3189	3190	3191	3192	3193	3194	3195	3196						
3197	3198	3199	3200	3201	3202	3203	3204	3205						
3206	3207	3208	3209	3210	3211	3212	3213	3214						
3215	3216	3217	3218	3219	3220	3221	3222	3223						
3224	3225	3226	3227	3228	3229	3230	3231	3232						
3233	3234	3235	3236	3237	3238	3239	3240	3241						
3242	3243	3244	3245	3246	3247	3248	3249	3250						
3251	3252	3253	3254	3255	3256	3257	3258	3259						
3260	3261	3262	3263	3264	3265	3266	3267	3268						
3269	3270	3271	3272	3273	3274	3275	3276	3277						
3278	3279	3280	3281	3282	3283	3284	3285	3286						
3287	3288	3289	3290	3291	3292	3293	3294	3295						
3296	3297	3298	3299	3300	3301	3302	3303	3304						
3305	3306	3307	3308	3309	3310	3311	3312	3313						
3314	3315	3316	3317	3318	3319	3320	3321	3322						
3323	3324	3325	3326	3327	3328	3329	3330	3331						
3332	3333	3334	3335	3336	3337	3338	3339	3340						
3341	3342	3343	3344	3345	3346	3347	3348	3349						
3350	3351	3352	3353	3354	3355	3356	3357	3358						
3359	3360	3361	3362	3363	3364	3365	3366	3367						
3368	3369	3370	3371	3372	3373	3374	3375	3376						
3377	3378	3379	3380	3381	3382	3383	3384	3385						
3386	3387	3388	3389	3390	3391	3392	3393	3394						
3395	3396	3397	3398	3399	3400	3401	3402	3403						
3404	3405	3406	3407	3408	3409	3410	3411	3412						
3413	3414	3415	3416	3417	3418	3419	3420	3421						
3422	3423	3424	3425	3426	3427	3428	3429	3430						
3431	3432	3433	3434	3435	3436	3437	3438	3439						
3440	3441	3442	3443	3444	3445	3446	3447	3448						
3449	3450	3451	3452	3453	3454	3455	3456	3457						
3458	3459	3460	3461	3462	3463	3464	3465	3466						
3467	3468	3469	3470	3471	3472	3473	3474	3475						
3476	3477	3478	3479	3480	3481	3482	3483	3484						
3485	3486	3487	3488	3489	3490	3491	3492	3493						
3494	3495	3496	3497	3498	3499	3500	3501	3502						
3503	3504	3505	3506	3507	3508	3509	3510	3511						
3512	3513	3514	3515	3516	3517	3518	3519	3520						
3521	3522	3523	3524	3525	3526	3527	3528	3529						
3530	3531	3532	3533	3534	3535	3536	3537	3538						
3539	3540	3541	3542	3543	3544	3545	3546	3547						
3548	3549	3550	3551	3552	3553	3554	3555	3556						
3557	3558	3559	3560	3561	3562	3563	3564	3565						
3566	3567	3568	3569	3570	3571	3572	3573	3574						
3575	3576	3577	3578	3579	3580	3581	3582	3583						
3584	3585	3586	3587	3588	3589	3590	3591	3592						
3593	3594	3595	3596	3597	3598	3599	3600	3601						
3602	3603	3604	3605	3606	3607	3608	3609	3610						
3611	3612	3613	3614	3615	3616	3617	3618	3619						
3620	3621	3622	3623	3624	3625									

TST	831	847	1649	1755	1827	1900	1925	1984	2012	2291	2305	2424	2431	2440	2465
TSTB	2510	2528	2530	2535	2532	2643	2684	2692	2714	2784	2817	2829	2835	2438	2476
.ASCII	839	1870	1886	1898	1909	1934	1952	1972	2053	2258	2274	2329	2335	2438	2476
.ASCII2	2503	2513	2526	2667	2716										
.ASCII2	641	640													
	2157	2164	2081	2084	2092	2101	2110	2113	2117	2134	2128	2134	2140	2145	2152
	2647		2173	2182	2189	2197	2205	2212	2218	2224	2233	2379	2380	2381	2383
.BLKB	2378														
.BYTE	606	607	612	613	621	622	630	631	632	633	657	658	659	669	676
	677	679	677	682	683	1143	1148	1295	1290	1332	1337	1342	1345	1390	1414
	1435	1449	1433	1512	1676	1690	1692	1687	1692	1697	1702	1707	1711	1716	1720
	1725	1730	1732	2162	2171	2180	2187	2195	2203	2210	2216	2222	2231	2376	2377
	2541	2542	2543	2582	2583	2802	2803	2804	2805						
.DSABL	2316														
.ENABL	1	417	2249												
.END	2835														
.ENDC	422	431	523	537	546	547	548	549	560	567	571	573	578	580	587
	600	604	606	634	638	639	643	646	668	676	679	682	685	686	687
	698	699	699	708	792	793	796	793	800	802	803	805	807	828	864
	835	836	867	945	947	948	949	1025	1026	1027	1028	1072	1076	1077	1078
	1079	1122	1126	1127	1128	1129	1173	1174	1175	1176	1216	1220	1221	1222	1223
	1263	1269	1270	1271	1272	1316	1317	1318	1319	1369	1370	1371	1372	1419	1420
	1421	1422	1482	1483	1484	1485	1546	1547	1548	1549	1640	1641	1642	1644	1646
	1656	1661	1664	1665	1666	1668	1674	1676	1679	1938	2249	2250	2252	2280	2316
	2320	2348	2349	2356	2358	2361	2363	2379	2385	2388	2394	2438	2445	2448	2452
	2458	2460	2471	2474	2475	2476	2478	2503	2505	2489	2492	2495	2496	2499	2526
	2541	2553	2556	2556	2570	2575	2576	2577	2577	2596	2600	2601	2605	2620	2649
	2640	2642	2642	2645	2646	2648	2651	2657	2662	2678	2680	2684	2687	2688	2839
.EQUIV	431	432	440	455	456	485	486	487	488	489	490	491	492	493	494
.EVEN	513	514	515	516	517	518	519	520	521	522					
.IF	646	1733	2240	2544	2648										
	418	428	485	523	545	546	547	548	549	558	566	569	570	577	579
	586	588	603	605	634	638	639	642	643	645	648	676	679	682	685
	636	637	633	693	690	692	708	787	792	794	793	798	800	832	803
	805	823	833	865	867	945	947	949	1024	1026	1029	1071	1075	1077	1079
	1121	1125	1127	1128	1172	1174	1176	1215	1219	1221	1223	1262	1268	1270	1272
	1315	1317	1319	1368	1370	1372	1418	1420	1422	1481	1483	1485	1545	1547	1549
	1639	1640	1641	1642	1643	1644	1648	1650	1652	1655	1656	1658	1674	1676	1677
	1679	1927	2248	2250	2251	2252	2290	2319	2357	2348	2348	2348	2361	2362	2378
	2379	2385	2387	2390	2391	2392	2417	2451	2457	2458	2458	2458	2473	2474	2476
	2477	2478	2479	2480	2481	2482	2491	2494	2496	2499	2501	2503	2552	2555	2565
	2569	2573	2573	2573	2573	2573	2573	2573	2573	2573	2573	2573	2573	2573	2673
	2732	2733	2733	2733	2733	2733	2733	2733	2733	2733	2733	2733	2733	2733	2842
	2844	2844	2844	2844	2844	2844	2844	2844	2844	2844	2844	2844	2844	2844	
.IFF	429	545	546	548	549	567	571	573	578	580	587	600	603	606	634
	643	646	646	664	665	666	667	667	647	648	649	1025	1026	1027	1028
	1072	1076	1077	1078	1079	1122	1126	1127	1128	1129	1173	1174	1175	1176	1216
	1220	1221	1222	1223	1263	1269	1270	1271	1272	1316	1318	1319	1369	1370	1370
	1371	1372	1419	1420	1421	1422	1482	1483	1484	1485	1546	1547	1549	1549	1640
	1643	1648	1661	1664	1676	1677	1927	2249	2252	2320	2322	2348	2349	2358	2362
	2379	2388	2395	2417	2474	2475	2477	2477	2477	2495	2553	2555	2570	2586	2605
	2840	2849	2853	2873	2810	2816	2845	2845	2876						
.IFT	2822	2827	2811	2831	2838	2883	2876	2861							
.IFTF	2267	2320	2323	2407	2415	2437	2482	2575							

MAINDEC-11-DZVTD-B MACY11 27(732) 25-SEP-76 10:10 PAGE 86
 DZVTDB.SRC CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

.IIF	417	422	427	542	543	544	546	548	549	555	642	646	793	796	802
	803	805	876	1641	1642	1656	1657	1676	1679	2249	2270	2371	2379	2385	2438
	2448	2449	2450	2451	2452	2456	2483	2484	2492	2556	2557	2558	2559	2564	2588
.IRP	2596	2601	2617	2642	2724	2831	2832	2833	2834	2836	2838	2839	2840	2841	
	708	863	945	1024	1075	1125	1172	1219	1268	1315	1368	1418	1481	1545	1648
	2401	2427	2457	2500	2501	2522	2538	2539	2565	2848	2854	2857	2868		
.LIST	1	417	537	548	555	634	636	637	638	643	646	708	763	807	863
	867	945	949	1024	1028	1075	1079	1125	1129	1172	1176	1219	1223	1268	1272
	1315	1319	1368	1372	1418	1422	1481	1485	1545	1549	1655	1668	2348	2451	2536
	2823	2831	2832	2833	2834	2835	2836	2837	2838	2839	2840	2841	2842		
.MACRO	1	549	597	754	823	1544	2823								
.MCALL	417	537	643	807											
.NEXT	691														
.NLIST	1	417	537	548	555	634	636	637	638	643	646	708	763	807	863
	867	945	949	1024	1028	1075	1079	1125	1129	1172	1176	1219	1223	1268	1272
	1315	1319	1368	1372	1418	1422	1481	1485	1545	1549	1656	1668	2348	2451	2596
	2623	2831	2832	2833	2834	2835	2836	2837	2838	2839	2840	2841	2842		
.PAGE	597	692													
.PCH	1														
.REPT	555	636	891	925	973	1004	1153	1295	1342	1395	1461	1524	1616		
.SBTTL	427	538	549	558	564	575	597	643	692	786	863	945	1024	1075	1125
	1172	1219	1268	1315	1368	1418	1481	1545	1637	2077	2246	2385	2442	2492	2550
	2602	2650	2730	2807	2823	2842									
.TITLE	417														
.WORD	555	556	557	572	591	592	593	594	595	596	605	608	609	610	611
	614	615	616	617	618	619	620	623	624	625	634	636	637	648	649
	650	651	652	653	654	655	659	660	661	674	678	681	684	685	686
	687	688	689	1144	1149	1286	1291	1333	1338	1363	1386	1391	1415	1436	1450
	1451	1499	1513	1514	1660	1663	1675	2434	2437	2524	2628	2633	2679	2726	2806
	2877	2879													

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

* DZVTDB.SRC/SOL/CRF/PAGNUM/NL: TOC=DZVTDB.SML, DZVTDB.SRC
 RUN-TIME: 36 46 6 SECONDS
 RUN-TIME RATIO: 184/89=2.0
 CORE USED: 34K (67 PAGES)

[illegible]