

Tektronix 7854 Source Code

Version 1.02

82/02/08

TMS 9900 ASSEMBLER V1.3F 82/02/08. 13.57.47. 7854 OSCILLOSCOPE FIRMWARE - VERSION 01.02

This document is an image that was created from a continuous fan-folded paper listing. The listing was used in a design review in 1982. It was produced by a high-speed line printer with a sprocket-driven paper feeder. Many characters in the original document are consistently weak and not fully resolved but are discernible to the naked eye. Such resolution and character recognition are not very machine memory efficient. A compromise was made to salvage the contents of the original document while keeping a manageable file size.

The first four pages are the original Table of Contents (TOCs). Links have been added to jump to the desired pages. Use the Bookmark at the left side panel to return to first page of the TOCs.

This document can be searched but with poor coverage.

TABLE OF CONTENTS

2	PROJECT PERSONNEL	00002
31	OVERVIEW & RULES	00003
140	REGISTER DEFINITIONS	00004
160	INTERRUPT VECTORS	00023
207	KEYTRE	00058
488	SETLAST	00277
505	BUTTONHIT	00288
535	BUFRIT	00315
594	POPBUFR	00359
668	NO-NUMBER-CHANGE TABLE	00428
681	COLDUP	00438
729	RAM TEST (8K BY 16)	00470
780	MAP RAM	00505
797	ROM TEST (16K BY 16)	00517
824	RAM TEST (1K BY 16)	00529
905	ROM TEST (8K BY 16)	00576
939	TEST REALTIME CLOCK	00592
990	TEST DISPLAY DONE INTERRUPT	00626
1015	WAIT ON POWER UP ERROR	00643
1030	INITIALIZE VARIABLES	00652
1251	CYCLE KEY	00812
1268	DISPLAY	00821
1259	'VXP' - VERTICAL EXPAND	00822
1404	VZR / HSCL / VSCL	00921
1426	'>HSCL', '>VSCL' - (X) TO HORIZONTAL, VERTICAL SCALE	00931
1461	SETZERO	00955
1543	CRT/WFM DISPLAY KEYS	01018
1743	VERTICAL/HORIZONTAL MODE KEYS	01158
1803	20 MILLI-SECOND CLOCK INTERRUPT HANDLER	01198
1962	DISPLAY INTERRUPT HANDLER	01314
2127	TEKCODE TO ASCII READOUT CONVERSION	01456
2279	UTILITY TEXT EDITING COMMANDS	01581
2382	ADD TEXT	01648
2479	SETUP 7854 4-LINE READOUT	01722
2909	VERTICAL POSITION KEYS' READOUT UPDATE	02084
2931	SORT WAVEFORMS	02099
2985	WAVEFORM ACQUISITION KEYS	02141
3276	'AQR' & 'GND' WAVEFORM ACQUISITION	02413
3426	'AVE' WAVEFORM ACQUISITION	02543
3522	'AQS' WAVEFORM ACQUISITION	02632
3561	'RDOJT' KEY HANDLER - READOUT ACQUISITION	02662
3615	COUNT REALTIME WAVEFORMS	02696
3674	REALTIME READOUT SEARCH PATTERN	02731
3757	CHECK VALID SCALE FACTORS	02767
3822	DECODE REALTIME READOUT	02803
3886	LOCATE SCALE FACTOR	02854
3929	WAVEFORM FILL CHECK	02893
3957	SWEEP INTERRUPT HANDLER	02921
3973	UNDEFINED INTERRUPT HANDLER	02937
3997	TABLE - REALTIME WAVEFORMS	02952
4097	TABLE - REALTIME READOUTS	03000
4201	TABLE - READOUT ADDRESSES AND POSITIONS	03059
4258	FRONT PANEL ACQUISITION MODES	03084
4275	GPIB	03101
4276	>>>> 7854 GPIB NOTES <<<<<	03102
4312	GPIB INTERRUPT HANDLER	03103
4376	(SELECTIVE) DEVICE CLEAR	03148
4420	GROUP EXECUTE TRIGGER	03180
4436	BYTE OUT	03186

TABLE OF CONTENTS

4459	BYTE IN	03193
4695	SERVICE REQUEST CONTROL KEYS	03367
4720	KEY 'READX'	03392
4888	DECODE WAVEFORM PREAMBLE	03527
5002	DECODE WAVEFORM	03606
5079	KEY '>TEXT'	03662
5161	KEY 'TEXT'	03721
5230	KEY 'SENDX'	03756
5280	CONSTRUCT WAVEFORM PREAMBLE	03780
5369	CONSTRUCT WAVEFORM	03842
5401	KEY 'SAVE'	03866
5478	KEY 'ID'	03922
5619	SETUP	04033
5642	QUERY 'ID?'	04043
5670	QUERY 'SRQ?'	04054
5693	QUERY 'ERR?'	04064
5719	KEYS 'SWL' & 'SWH'	04076
5727	KEY 'RQS'	04083
5731	I/O CONTROL TABLE FOR 'READX' & 'SENDX'	04087
5791	QUERY TABLE	04134
5810	STRLSTN - START LISTEN FUNCTION FOR GPIB INPUT KEYS	04145
5875	STRTALK - START TALK FUNCTION FOR GPIB OUTPUT KEYS	04189
5933	CHKDLM - CHECK FOR DELIMITER	04226
5958	GPIBIN - GPIB INPUT	04235
5999	GPIBOUT - GPIB OUTPUT	04249
6038	TERMINATE - GPIB OUTPUT TERMINATION	04271
6088	CHKEOI - CHECK IF <EOI> CAME WITH LAST DATA BYTE	04304
6099	INSTR - INPUT STRING FROM GPIB	04308
6119	INVAL - INPUT VALUE FROM GPIB	04321
6158	OUTSTR - OUTPUT STRING TO GPIB	04337
6184	OUTVAL - OUTPUT VALUE TO GPIB	04344
6224	RQS - REQUEST SERVICE	04360
6320	GPIB CONSTANTS	04418
6341	***** HARDWARE CONFIGURATION/INTERFACE *****	04435
6454	ROM	04519
6838	CONVRT TABLE	04822
6906	FORMAT TABLE FOR 4-LINE READOUT	04881
7033	TEKCODE TO ASCII CONVERSION TABLE	04987
7137	FRONT PANEL BUTTON KEY CODES	05075
7154	SPECIAL KEY TABLE FOR PROGRAM ENTRY AND EDITING	05092
7205	KEY HANDLERS	05135
7810	MNUMONIC TABLE	05650
8115	SYMBOL TABLE FOR ALPHA EXPONENTS	05946
8125	***** ROM HEADER FOR LOW ROMS *****	05956
8143	***** PROGRAMMING *****	05968
8145	KEYPROG	05970
8197	FWDNXT - RETURNS LOCATION OF FIRST COMMAND IN NEXT LINE	06012
8217	RVSXNT - RETURNS LOCATION OF FIRST COMMAND IN CURRENT LINE	06023
8239	CLRPTR	06036
8254	PROGIN	06044
8445	TXTSCRN	06227
8599	MOVSJB	06464
8739	GETLINE / GETPNT	06496
8797	KEYRTN	06535
8829	KEYGSB / KEYGOTO	06559
8893	KEYL3L	06612
8939	KEYNEXT / KEYPREV / KEYSTP	06650
9077	KEYLNN	06766
9093	KEYCLL	06774

TABLE OF CONTENTS

9126	KEYCLP	06802
9153	KEYPAUSE	06824
9179	KEYRUN	06843
9196	KEYSTART	06853
9219	KEYF / KEYCLF / KEYNOP	06869
9261	LASTNXT	06906
9281	MATHKY	06921
9282	KEYMJLT / KEYDIV / KEYPLUS / KEYMINJS / KEYIFXEQY / KEYIFYGTX	06922
9549	CHGSCL - SET SCALE FACTOR TO 1 AND BLANK UNITS	07146
9585	MATHARG --- GET ARGUMENTS FOR BINARY MATH FNCTIONS	07169
9676	SCALWFM --- DO SCALING RUN FOR FOUR-FUNCTION MATH KEYS	07236
9715	SCALWCRK --- DO MAIN WORK FOR SCALING RUN OF WFMS	07271
9794	KEYABS / KEYSGN	07328
9869	KEYEXP / KEYLN / KEYSQRT	07395
10022	KEYSMOOTH	07519
10192	'HXPJ' - HORIZONTAL EXPAND	07555
10416	KEYITRP	07863
10529	KEYINTG	07947
10706	HORIZONTAL POSITIONING	08098
10745	ROTATE WAVEFORM RIGHT	08136
10801	KEYDIFF	08182
10926	KEYMAX / KEYMIN / KEYMID / KEYP2P	08288
10995	KEYMEAN / KEYSMS	08351
11022	GETMID	08371
11067	KEYRISE / KEYFALL	08404
11154	KEYWIDTH	08475
11171	KEYDELAY	08487
11190	GETENDS --- GET END POINTS OF WINDOWS FOR FUNCTIONS WITH CURSOR	08501
11254	KEYAREA / KEYENERGY	08552
11282	KEYPER	08573
11291	KEYFREQ	08577
11313	GETPER --- DO THE DIRTY WORK FOR KEYS PER AND FREQ	08594
11350	KEYORD	08626
11419	KEY2ORD	08684
11573	KEYCRS1 / KEYCRS2.1 / KEYOFF	08827
11590	KEYHCRD / KEYVCRD	08839
11623	KEY2VCRD	08857
11721	KEY2HCRD	08925
11790	KEYCRS1LFT / KEYCRS1RGT / KEYCRS2LFT / KEYCRS2RGT --- CURSOR	08978
11943	KEYVPOD / KEYVPOP --- VERTICAL POSITION SLEW KEYS	09073
12048	KEYWFM	09150
12069	KEY2WFM	09166
12101	KEYCNS	09193
12114	KEY2CNS	09201
12130	CONCJM	09212
12150	KEYPNT	09226
12177	KEY2PNT	09242
12235	KEY2P.W	09289
12348	KEYP.W	09377
12363	KEYCLX / KEYCLR	09385
12388	KEYROLL	09398
12401	KEYX2Y	09406
12423	KEYENTER	09414
12438	DIGIT	09420
12600	SCAN NUMERIC INPUT AND CREATE FLOATING POINT NUMBER	09572
12784	GETEEX	09741
12806	GENSPT	09752
12807	CALCULATE SCALES	09753
12944	POPPING	09853

TABLE OF CONTENTS

12971	PSHING	09868
13010	ZEROW0	09894
13017	ADRWFH	09901
13049	OPWH2W0,XFRHEAD --- TRANSFER HEADER	09917
13090	HORDLTA --- FIND HORIZONTAL VALUE OF POINT SPREAD	09947
13111	CROSSPOINTS -- FORWARD AND BACKWARD SEARCHES ON WFM	09960
13279	FUZZCHK --- CHECKS TO SEE IF TWO VALUES ARE EQUAL TO WITHIN 1	10099
13304	FP2WFM --- FLOATING POINT TO WFM MANTISSA	10110
13336	PNT2FP & WFM2FP - WAVEFORM POINT TO FLOATING POINT VALUE	10132
13378	FILLPNT --- FILL UNDEFINED POINTS	10159
13492	OPWC4G	10238
13527	BUZZIT	10264
13557	FP2ELE --- CONVERT FROM FP HORIZONTAL VALUE TO ELEMENT NUMBER	10288
13590	SUM	10311
13676	SUMS3R	10371
13790	NEWHSC1	10459
13818	NEWVSCL	10478
13845	I2FP	10494
13860	NULLWFM	10504
13891	FPART4	10525
13892	ADDITION/SUBTRACTION	10526
14011	MULTIPLICATION/DIVISION	10625
14136	NORMALIZATION	10730
14235	ZERO EXPONENT	10807
14337	NATURAL LOGARITHM	10887
14495	EXPONENTIATION	11006
14629	SQUARE ROOT	11109
14702	ARITHMETIC COMPARISON	11156
14756	INTEGER TO FLOATING POINT	11194
14793	FLOATING POINT TO INTEGER	11210
14842	ASCII OUTPUT CONVERSION	11240
15283	AUTOSCALE TO TEK STANDARD (1-2-5 VERTICAL SCALE)	11549
15496	***** ROM HEADER FOR HIGH ROMS *****	11735
15514	RAM VARIABLES	11747
15733	PROM PATCHES FOR VERSION 01.02	

PROJECT PERSONNEL

00032

3	*	PROJECT MANAGER
4	*	
5	*	TOM ROUSSEAU
6	*	
7	*	PROJECT ENGINEER, SYSTEM
8	*	
9	*	VAL GARUTS
10	*	
11	*	SOFTWARE ENGINEERS
12	*	
13	*	BURT JOHNSON
14	*	JIM SCHLEGEL
15	*	ELLEN DELEGANES
16	*	
17	*	ELECTRICAL ENGINEERS
18	*	
19	*	JIM FALLMAN
20	*	GARY FLADSTOL
21	*	KIRK WIMMER
22	*	LES LARSON
23	*	JACK COLLINS
24	*	WES KOSTA
25	*	TIM HOLTE
26	*	
27	*	EVALUATION ENGINEERS
28	*	
29	*	GREG ROGERS
30	*	JIM PETERSON

```
32 *****
33 **
34 **          7854 SMART SCOPE SOFTWARE RULES          **
35 **09/20/79                                           **
36 *****
37 *(TD/11,21,31/)
38 *
39 *I.  UNIVERSAL REGISTERS
40 *      A.  SOFT = SOFTWARE USE STACK POINTER  (R9)
41 *      B.  USER = USERS STACK POINTER      (R10)
42 *
43 *II. NUMERIC REPRESENTATION IN USERSTACK (IN ORDER POPPED)
44 *      A.  FLOATING POINT ---
45 *              FLAG      0000
46 *              MANTISSA  XXXX
47 *              EXPONENT  XXXX
48 *      B.  WAVEFORM NUMBER (WFM#) ---
49 *              FLAG      0001
50 *              WFM#      XXXX
51 *              XTRA      ----
52 *
53 *III. SUBROUTINE LEVELS
54 *
55 *      A.  CALLING:
56 *          1) MAIN CONTROL IS LEVEL 0
57 *          2) BLWP TO HIGHER NUMBERED LEVELS (LEVELS 1-4)
58 *          3) BL TO LEVEL 5 ROUTINES FROM ANY LEVEL
59 *          4) CANNOT CALL LOWER NUMBERED LEVEL
60 *          5) R7,R8,R12 MAY BE DESTROYED BY A LEVEL 5 ROUTINE
61 *             (CHECK LEVEL 5 ROUTINES CAREFULLY AS SOME VIOLATE
62 *              THIS RULE, OTHERS HAVE FURTHER RESTRICTIONS)
63 *
64 *      B.  LEVEL 0 - INTERRUPT HANDLERS
65 *          -----
66 *          1) CALLED VIA INTERRUPTS
67 *             A) WP & PC DEFINED IN INTERRUPT VECTOR TABLE
68 *          2) MAY USE ANY REGISTER FOR INTERNAL USE
69 *          3) ONLY MAIN CONTROL CAN USE SOFTSTK & USERSTK
70 *          4) EACH INTERRUPT HAS ITS OWN WORKSPACE
71 *          5) ONLY MAIN CONTROL (ENTERED VIA PUP INTERRUPT)
72 *             CAN CALL LEVEL 1-4 ROUTINES
73 *
74 *      C.  LEVEL 1 - KEY HANDLERS
75 *          -----
76 *          1) CALLED VIA BLWP FROM LEVEL 0
77 *          2) MAY USE ANY REGISTER FOR INTERNAL USE
78 *          3) MAY USE SOFTSTK & USERSTK ACCORDINGLY:
79 *             A) SOFTSTK - SOFTWARE STACK FOR TEMPORARY STORAGE
80 *                AND ARGUMENT PASSING
81 *             B) USERSTK - USER STACK (X,Y,Z,T)
82 *          4) RESPONSIBLE FOR RETRIEVING & RETURNING ARGUMENTS
83 *             FROM AND TO CALLING ROUTINES
84 *          5) TERMINATES IN RTWP
85 *
86 *      D.  LEVEL 2 - TOP LEVEL KEY SUPPORT (AUTOSCALE, FP2ASC, ETC.)
87 *          -----
88 *          1) CALLED VIA BLWP FROM LEVELS 0-1
89 *          2) SEE LEVEL 1 - 2-5
90 *          3) MUST OBTAIN SOFT & USER FROM CALLING PROGRAM
```

```

91 *
92 *
93 * E. LEVEL 3 - ARITHMETIC FUNCTIONS (EXP(X), LN(X), SQRT(X))
94 * -----
95 * 1) CALLED VIA BLWP FROM LEVELS 0-2
96 * 2) SEE LEVEL 2 - 2-5
97 *
98 * F. LEVEL 4 - FLOATING POINT ARITHMETIC
99 * -----
100 * 1) CALLED VIA BLWP FROM LEVELS 0-3
101 * 2) SEE LEVEL 2 - 2-5
102 *
103 * G. LEVEL 5 - I/O, SIMPLE CONVERSION - BOTTOM LEVEL SUPPORT
104 * -----
105 * 1) CALLED VIA BL FROM LEVELS 0-4 OR INTERRUPT HANDLERS
106 * 2) MAY DESTROY R7,R8,R12
107 * 3) TERMINATES IN: 3L *R11
108 * 4) SEE LEVEL 1 - 4
109 *

```

110 *IV. SAMPLE PROGRAM FLOW (KEY 'INTG')

111	*	LEVEL	ROUTINE	COMMENT
112	*	=====	=====	=====
113	*			
114	*			
115	*	0	MAIN CONTROL	IDLE
116	*	0	KEYBOARD INTERRUPT	RECEIVE KEY 'INTG'
117	*	0	MAIN CONTROL	SETUP & CALL 'KEYINTG'
118	*	1	KEYINTG	'INTG' ROUTINE
119	*	5	POPREG	POP (X) OFF USERSTACK
120	*	1	KEYINTG	CHECK IF (X) IS A WFM NUMBER
121	*	4	OPWH2W0	TRANSFER OPW'S HEADER TO WJ
122	*	5	ADRWFM	GET ADDRESS OF OPW'S HEADER
123	*	4	OPWH2W0	
124	*	5	ADRWFM	GET ADDRESS OF W0'S HEADER
125	*	4	OPWH2W0	DO THE TRANSFER
126	*	1	KEYINTG	FIND MAX & MIN OF INTEGRAL WFM
127	*	4	FPMPY	VERT. SCALE * HORZ. INCREMENT
128	*	1	KEYINTG	PUSH MAX & MIN ONTO SOFTSTACK
129	*	5	NRMLIZ	CONVERT MINIMUM TO SINGLE PRECISION
130	*	1	KEYINTG	
131	*	4	FPMPY	CALCULATE ACTUAL VALUE OF MINIMUM
132	*	1	KEYINTG	SAVE MINIMUM FOR AUTOSCALE
133	*	5	NRMLIZ	CONVERT MAXIMUM TO SINGLE PRECISION
134	*	1	KEYINTG	
135	*	4	FPMPY	CALCULATE ACTUAL VALUE OF MAXIMUM
136	*	1	KEYINTG	SAVE MAXIMUM FOR AUTOSCALE
137	*	2	SCALE	AUTOSCALE VERTICAL TO TEK 1-2-5
138	*	1	KEYINTG	FINISH INTEGRATION
139	*	0	MAIN CONTROL	RETURN TO MAIN CONTROL

REGISTER DEFINITIONS

00014

141	0000	R0	EQU	0	REGISTER 0	00005
142	0001	R1	EQU	1	REGISTER 1	00006
143	0002	R2	EQU	2	REGISTER 2	00007
144	0003	R3	EQU	3	REGISTER 3	00008
145	0004	R4	EQU	4	REGISTER 4	00009
146	0005	R5	EQU	5	REGISTER 5	00010
147	0006	R6	EQU	6	REGISTER 6	00011
148	0007	R7	EQU	7	REGISTER 7	00012
149	0008	R8	EQU	8	REGISTER 8	00013
150	0009	R9	EQU	9	REGISTER 9	00014
151	000A	R10	EQU	10	REGISTER 10	00015
152	000B	R11	EQU	11	REGISTER 11	00016
153	000C	R12	EQU	12	REGISTER 12	00017
154	000D	R13	EQU	13	REGISTER 13	00018
155	000E	R14	EQU	14	REGISTER 14	00019
156	000F	R15	EQU	15	REGISTER 15	00020
157						
158	0009	SOFT	EQU	R9	SOFTSTACK STACK POINTER	00021
159	000A	USER	EQU	R10	USER STACK POINTER	00022

INTERRUPT VECTORS

00023

161			OPT GENERATE		00024
162		0000	ORG \$0000		00025
163	0000	0B00	WORD WPKYTR	POWER UP-DOWN	00026
164	0002	04E2	WORD COLDUP		00027
165			*		
166			WORD \$E200	HARDWARE STATUS DISPLAY INTERRUPT VECTOR	
167			WORD \$E218		
168			*		
169			WORD \$E604		
170			WORD \$F710		
171			WORD \$E624	DDT INTERRUPT VECTORS	
172			WORD \$F78E		
173			WORD \$E644		
174			WORD \$F734		
175			*		
176	0004	0C70	WORD WPINT		00028
177	0006	2210	WORD INTU		00029
178	0008	0C70	WORD WPINT		00030
179	000A	2210	WORD INTU		00031
180	000C	0C70	WORD WPINT		00032
181	000E	2210	WORD INTU		00033
182	0010	0C70	WORD WPINT		00034
183	0012	2210	WORD INTU	UNDEFINED INTERRUPTS	00035
184	0014	0C70	WORD WPINT		00036
185	0016	2210	WORD INTU		00037
186	0018	0C70	WORD WPINT		00038
187	001A	2210	WORD INTU		00039
188	001C	0C70	WORD WPINT		00040
189	001E	2210	WORD INTU		00041
190	0020	0C70	WORD WPINT		00042
191	0022	2210	WORD INTU		00043
192					
193	0024	0C00	WORD WPCLK	20 MILLISECOND REALTIME CLOCK	00044
194	0026	0E25	WORD CLOCK		00045
195	0028	0C20	WORD WPDSP	DISPLAY DONE	00046
196	002A	0FBC	WORD DISPLAY		00047
197	002C	0C60	WORD WPGPIB	GPIB	00048
198	002E	234E	WORD GPIB		00049
199	0030	0B20	WORD WPKEYS	KEYBOARD	00050
200	0032	0336	WORD BUTTON-HIT		00051
201	0034	0C40	WORD WPSWP	SWEEPS COUNTER	00052
202	0036	210A	WORD SWEEP		00053
203	0038	0C70	WORD WPINT		00054
204	003A	2210	WORD INTU		00055
205	003C	0C70	WORD WPINT		00056
206	003E	2210	WORD INTU		00057


```

208 *
209 * KEYTRE --- HANDLES GENERAL EXECUTION FLOW BETWEEN KEYS
210 *
211 * TOP LEVEL WITH IT'S OWN WORKSPACE
212 *
213 * SEE FLOWCHART FOR ADDITIONAL INFORMATION
214 *
215 * FPKY WORD WPBUFR 00059 DEL
216 * WORD BUFRT 00060 DEL
217 * GPIBKY WORD WPBUFR 00061 DEL
218 * WORD BUFRT 00062 DEL
219 *
220 * PROBLEM #8 - PATCH #9 (1 OF 1) 00009PATCH
221 *
222 * FPKY & GPIBKY B.WP VECTORS POINT TO ANOTHER B.WP VECTOR 00009PATCH
223 * INSTEAD OF POINTING TO CODE. 00009PATCH
224 *
225 0040 DB40 FPKY WORD WPBUFR 00009PATCH
226 0042 038E WORD BUFRT+4 POINT TO CODE 00009PATCH
227 0044 DB40 GPIBKY WORD WPBUFR 00009PATCH
228 0046 038E WORD BUFRT+4 POINT TO CODE 00009PATCH
229 *
230 * END OF PROBLEM #8 00009PATCH
231 *
232 0048 DB00 START WORD WPKYTRE 00063
233 004A 004C WORD KEYTRE 00064
234 KEYTRE EQU * 00065
235 004C 0300 000A LIMIT $A 00066
236 0050 02E0 0960 LWPI WPLVL1 00067
237 0054 0260 33DE MOV SOFTST,SOFT 00068
238 0058 02A0 33DE MOV XSTK,USER 00069
239 005C 02E0 0B00 LWPI WPKYTRE 00070
240 0060 0250 33DE MOV SOFTST,SOFT 00071
241 0064 02A0 33DE MOV XSTK,USER 00072
242 0068 0020 094A MOV FATAL,R0 00073
243 006C 111C JLT CHKSTP 00074
244 * MOV PROGRAM,R1 DO NOT WIPE BUFFER ON PROGRAM INPUT 00075 DEL
245 * JNE *+6 00076 DEL
246 *
247 * PROBLEM #15 - PATCH #18 (1 OF 2) 00018PATCH
248 *
249 * 'STOP' AFTER 'SAVE' IN PROGRAM ENTRY MODE DOESN'T CLEAR 00018PATCH
250 * COMMAND BUFFER 00018PATCH
251 *
252 006E 0460 971A B PATC#18 BRANCH TO PATCH #18 00018PATCH
253 *
254 * END OF PATCH #18 OF PROBLEM #15 00018PATCH
255 *
256 0072 1000 JMP *+002 00018PATCH
257 0074 0450 01CE BUFCHK1 B BUFCHK 00077
258 *
259 * HANDLE ERRORS:
260 * IF IN GPIB COMMAND STRING, IGNORE REST OF COMMANDS IN
261 * STRING AND RESTART GPIB
262 *
263 0078 04E0 09C0 CLR HALTIT IF HANDLING ERROR CASE, ALREADY STOPPED 00078
264 007C 0020 093A MOV ACPTCMD,R0 IS GPIB ACCEPTING COMMANDS? 00079
265 0080 1304 JEQ NOGCMD 00080
266 0082 0720 093E SETJ IGNORE YES, IGNORE REST OF STRING 00081
    
```

267	0086	0020	E05C		MOV B R7R,R0		00082
268	008A	C020	D992	NOGCMJ	MOV SUSGP1B,R0	IS GPIB SUSPENDED?	00083
269	008E	1308			JEQ ACTGP1B		00084
270	0090	0820	D994	E058	MOV B AMREG,R2W	YES, RESET RFD HOLD OFF	00085
271	0096	0820	3386	E06C	MOV B RFDR,R3W	COMPLETE HANDSHAKE	00086
272	009C	04E0	D992		CLR SUSGP1B		00087
273	00A0	04E0	D918	ACTGP1B	CLR CMDINDX		00088
274	00A4	101C			JMP WIPBUF2	WIPE BUFFER ON ERROR	00089
275	00A6	C020	D9C0	CHKSTP	MOV HALTIT,R0	WAS PANIC KEY HIT?	00090
276	00AA	13E4			JEQ BUFCHK1	IF NOT, CHECK BUFFER	00091
277				*			
278				*	HANDLE STOP FROM KEYBOARD OR FROM GPIB		
279				*			
280	00AC	C060	DAA0		MOV PROGRAM,R1	STOP ONLY IF IN EXECUTE MODE	00092
281	00B0	1533			JGT STOPR0G		00093
282	00B2	1326			JEQ RSTGP1B		00094
283	00B4	C320	33C4		MOV BUSYLED,R12	TURN ON BUSY LED	00095
284	00B8	1D00			SBO 0		00096
285	00BA	C060	D976		MOV INUSE,R1	WAS A KEY STOPPED IN PROGRESS?	00097
286	00BE	1505			JGT NOSTPWN	IF SO, NO WARNING	00098
287	00C0	04E0	D94C		CLR WARNING	WARN USER OF STOP IN IDLE MODE	00099
288	00C4	C820	3832	D958	MOV PANIC, LASTKEY	PUT OUT 'STOP' MNEMONIC	00100
289			00CA	NOSTPWN	EQU *		00101
290	00CA	04E0	DAE5		CLR RQSNJM		00102
291	00CE	06A0	32AE		BL RQS		00103
292	00D2	C820	333A	D976	MOV C1, INUSE	SET BUSY FLAG	00104
293	00D8	C020	D98A		MOV PROGRS,R0	IS NUMERIC ENTRY IN PROGRESS?	00105
294	00DC	1511			JGT RSTGP1B	DON'T WIPE BUFFER IF IN NUMERIC ENTRY	00106
295			00DE	WIP3JFR	EQU *		00107
296	00DE	0200	DA3A		LI R0, KEYBUF2		00108
297	00E2	C050	33E6		MOV BUFLEN,R1		00109
298	00E6	04F0			CLR *R0+	CLEAR BUFFER ON PANIC HALT	00110
299	00E8	0601			DEC R1		00111
300	00EA	16F0			JNE *-4		00112
301				*	SET0 PROGRAM	... AND STOP PROGRAM	00113 DEL
302				*			00019PATCH
303				*	PROBLEMS #14 & #15 - PATCH #19 (2 OF 2)		00019PATCH
304				*			00019PATCH
305				*	'RJN' AFTER ERROR DOESN'T INCREMENT TO NEXT COMMAND		00019PATCH
306				*			00019PATCH
307				*	'STOP' AFTER 'SAVE' IN PROGRAM ENTRY MODE DOESN'T CLEAR		00019PATCH
308				*	COMMAND BUFFER		00019PATCH
309				*			00019PATCH
310	00EC	0450	9732		B PATCH19	BRANCH TO PATCH #19	00019PATCH
311				*			00019PATCH
312				*	END OF PATCH #19 OF PROBLEMS #14 & #15		00019PATCH
313				*			00019PATCH
314	00F0	C820	33E6	D954	MOV BUFLEN, BUFAVL	RESET INDICATOR FOR CLEAR BUFFER	00114
315	00F6	0720	D972		SET0 RDTFLAG	CHANGE ALL READOUT LINES	00115
316	00FA	C020	D94A		MOV FATAL,R0	WAS IT AN ERROR?	00116
317	00FE	1367			JEQ BUFCHK	IF SO CHECK THE BUFFER	00117
318	0100	C020	D93C	RSTGP1B	MOV STPGP1B,R0	IS STOP FROM GPIB?	00118
319	0104	1320			JEQ STOPDON	NO, THEN GPIB OK	00119
320	0106	D820	D994	E068	MOV B AMREG,R2W	YES, RESET RFD HOLD OFF	00120
321	010C	D820	3386	E05C	MOV B RFDR,R3W	COMPLETE HANDSHAKE	00121
322	0112	04E0	D93C		CLR STPGP1B		00122
323	0116	1017			JMP STOPDON		00123
324			0118	STOPR0G	EQU *		00124
325	0118	C820	333A	D976	MOV C1, INUSE		00125

326	011E	0720	DAAG		SETO	PROGRAM	STOP THE PROGRAM	00126
327	0122	0820	3332	DAE6	MOV	C2,RQSNUM	CAUSE SRQ ON PROGRAM STOP	00127
328	0129	0820	DAAB	DABA	MOV	PROGLN,STPFLG	PUT OUT 'STOP IN' STATUS MESSAGE	00128
329	012E	0720	DAB8		SETO	STPROGF	INDICATE PROGRAM WAS STOPPED	00129
330	0132	0020	DAAA		MOV	EDITPNT,R0		00130
331	0136	9810	3833		CB	*R0,*ANICB	WAS STOP IN PROGRAM?	00131
332	013A	16E2			JNE	RSTGP1B		00132
333	013C	05A0	DAAA		INC	EDITPNT	MOVE EDIT POINTER PAST STOP	00133
334	0140	0820	333A	DAB0	MOV	C1,MOVFLAG		00134
335	0146	04E0	D9C0		STOPDGN CLR	HALTIT		00135
336	014A	0450	02D8		B	XCUTDGN		00136
337			014E		WAIT4KEY EQU *			00137
338	014E	0060	D93A		MOV	ACPTCMD,R1	FINISHED WITH COMMAND STRING?	00138
339	0152	163A			JNE	WAIT	NO, THEN NOT RETURN TO IDLE YET	00139
340	0154	0060	D976		MOV	INUSE,R1	IS THIS A RETURN TO IDLE?	00140
341	0158	131B			JEQ	CHKTWN	NO, THEN NO OPC SERVICE REQUEST	00141
342	015A	04E0	D976		CLR	INUSE	YES, FLAG 7854 IDLE	00142
343	015E	0050	D92E		MOV	STOPDCL,R1	IS THIS RETURN DUE TO A 'DCL' FROM GPIB?	00143
344	0162	1305			JEQ	RQSOK	NO, GENERATE A SERVICE REQUEST IF ON	00144
345	0164	04E0	D92E		CLR	STOPDCL	YES, DO NOT GENERATE SERVICE REQUEST	00145
346	0168	04E0	DAE6		CLR	RQSNJM	CLEAR ANY PENDING SERVICE REQUEST	00146
347	016C	1011			JMP	CHKTWN		00147
348	016E	0050	DAE6		RQSOK MOV	RQSNJM,R1	IS THERE AN RQS TO GENERATE?	00148
349	0172	1506			JGT	STSTRQS	YES, USE THAT ONE	00149
350	0174	0050	D928		MOV	REMKEY,R1	WAS A REMOTE COMMAND EXECUTED?	00150
351	0178	130B			JEQ	CHKTWN	NO, THEN NO RETURN TO IDLE SRQ	00151
352	017A	0820	333A	DAE6	MOV	C1,RQSNUM	RQS # FOR EXTERNAL COMMAND GROUP COMPLETED	00152
353	0180	0720	DA18		STSTRQS SETO	ROUPDT	FORCE CURRENT READOUT ON RETURN TO IDLE	00153
354	0184	0420	141A		BLWP	RDOUT	UPDATE CURRENT DISPLAY	00154
355	0188	06A0	32AE		BL	RQS	ISSUE SERVICE REQUEST	00155
356	018C	04E0	D928		CLR	REMKEY	CLEAR REMOTE RETURN TO IDLE FLAG	00156
357	0190	0060	D926		CHKTWN MOV	TOTL_0,R1	IS 7854 IN TALK/LISTEN MODE?	00157
358	0194	1619			JNE	WAIT	NO, CANNOT BE TALKED WITH NOTHING TO SAY	00158
359	0196	0060	E048		MOV3	R2R,R1	CHECK IF 7854 IS TALK ADDRESSED	00159
360	019A	2060	337E		COB	TACS,R1		00160
361	019E	1614			JNE	WAIT		00161
362	01A0	0060	E04C		MOV3	R3R,R1		00162
363	01A4	0241	7000		ANDI	R1,\$7000	KEEP HANDSHAKE LINES ONLY	00163
364	01A8	0281	3008		CI	R1,\$3008	IS THERE A VALID LISTENER?	00164
365	01AC	160D			JNE	WAIT	NO, EVERYTHING IS OK	00165
366	01AE	0300	000F		LIMI	\$F	ALLOW GPIB INTERRUPTS	00166
367	01B2	0701			SETO	R1	YES, SEND <FF>&<EOI> IF TEK TERMINATOR	00167
368	01B4	00A0	D924		MOV	TRMTPY,R2	OR <FF><CR><LF>&<EOI> IF HP TERMINATOR	00168
369	01B8	1305			JEQ	FFE0I		00169
370	01BA	06A0	3154		BL	GPIBOUT	OUTPUT <FF>	00170
371	01BE	06A0	3196		BL	TERMINATE	OUTPUT <CR><LF>&<EOI>	00171
372	01C2	1002			JMP	WAIT		00172
373	01C4	06A0	31F2		FFE0I BL	EOIOJT	OUTPUT <FF>&<EOI>	00173
374	01C8	0300	000F		WAIT LIMI	\$F	DON'T LET KEYBOARD STAY MASKED	00174
375	01CC	0340			IDLE			00175
376			01CE		RJNKEYR EQU *			00176
377			01CE		BJFRCHK EQU *			00177
378	01CE	0300	000A		LIMI	\$A		00178
379	0102	0420	141A		BLWP	RDOUT	KEEP REALTIME SCALE FACTORS UPDATED	00179
380	0106	0420	040A		BLWP	POPBJFR		00180
381	010A	0320	D992		MOV	SUSGPIB,R12	IS GPIB SUSPENDED?	00181
382	010E	1304			JEQ	NOTSUS		00182
383	01E0	04E0	D992		CLR	SUSGPIB	YES, START IT UP AGAIN	00183
384	01E4	0420	2648		BLWP	RESGPIB		00184

KEYTRE

00058

385	01E8	C020	D966		NOTSUS	MOV KEY,R0	KEYCODE OF NEXT KEY TO EXECUTE	00185
386	01EC	13B0				JEQ WAIT&KEY	IF NO KEY IN BUFFER, GO BACK AND WAIT	00186
387	01EE	0280	007F			CI R0,\$7F	IS THIS GPIB 'BAD KEY'?	00187
388	01F2	160C				JNE GOODKEY		00188
389	01F4	04E0	D94A			CLR FATAL	YES	00189
390	01F8	C820	D966	D968		MOV KEY,LASTKEY		00190
391	01FE	C820	333A	D976		MOV C1,INUSE	SET BUSY FLAG	00191
392	0204	C820	333E	JAE6		MOV C3,RQSNUM	RQS # FOR EXTERNAL COMMAND ERROR	00192
393	020A	1065				JMP XCUTDON		00193
394	020C	0280	0080		GOODKEY	CI R0,\$80	IS KEY 'PRE-SHIFTED'?	00194
395	0210	1503				JGT *+8		00195
396	0212	A820	D94E	D966		A GOLD,KEY	ADD SHIFT VALUE	00196
397	0218	04E0	D94E			CLR GOLD	CLEAR SHIFT VALUE NOW	00197
398	021C	04E0	DAC2			CLR EOPFLG	CLEAR END OF PROGRAM FLAG	00198
399	0220	0720	D96A			SETD TRUE	CLEAR TRUE/FALSE STATEMENT	00199
400	0224	0720	D94A			SETD FATAL	CLEAR ERROR FLAGS	00200
401	0228	0720	D94C			SETD WARNING		00201
402	022C	C320	33C2			MOV ERRLED,R12	TURN OFF ERROR LED	00202
403	0230	1E00				SBZ 0		00203
404	0232	C320	33C4			MOV BUSYLEJ,R12	TURN ON BUSY LED	00204
405	0236	1000				SBZ 0		00205
406	0238	C0A0	DAA0			MOV PROGRAM,R2	AM I INPUTTING A PROGRAM?	00206
407	023C	1602				JNE NOPRGNTR		00207
408	023E	0450	40E6			B PROGIN		00208
409			0242		NOPRGNTR	EQU *		00209
410	0242	1507				JGT STILNXT	'NEXT' IN DIRECT MODE IS 'STEP'	00210
411	0244	8820	D966	37EE		C KEY,NXTKEY		00211
412	024A	1603				JNE STILNXT		00212
413	024C	C820	3808	D966		MOV STEPKEYG,KEY		00213
414			0252		STI_NXT	EQU *		00214
415	0252	C020	D966			MOV KEY,R0		00215
416	0256	8800	3832			C R0,PANIC	IS IT STOP IN PROGRAM STORE?	00216
417	025A	1602				JNE *+6	NOT A STOP	00217
418	025C	0450	0118			B STOPROG	YES, STOP THE PROGRAM	00218
419	0260	0A20				SLA R0,2		00219
420	0262	0220	3836			AI R0,KEYTAB		00220
421	0266	06A0	0316			BL SETLAST		00221
422	026A	C010				MOV *R0,R0		00222
423	026C	C800	D948			MOV R0,KEYXFR2	POINT TO KEY HANDLER	00223
424	0270	0201	0402			LI R1,NUMCHG		00224
425	0274	8811	3C36			C *R1,ENDTAB		00225
426	0278	1303				JEQ NUMCHG		00226
427	027A	8031				C *R1+,R0		00227
428	027C	16FB				JNE *-8		00228
429	027E	101B				JMP XCUTE		00229
430			0280		NUMCHG	EQU *		00230
431	0280	C0A0	D9BA			MOV PROGRS,R2	NO, WAS LAST ONE?	00231
432	0284	1318				JEQ XCUTE		00232
433	0286	0200	DA62		NCHG2	LI R0,XTBJFR	YES, POINT TO STRING TO DECODE	00233
434	028A	C160	D9BA			MOV PROGRS,R5		00234
435	028E	0420	6618			BLWP SCANIN	CONVERT NUMBER	00235
436	0292	06A0	6918			BL POPSTK	POP GARBAGE	00236
437	0296	04C0				CLR R0		00237
438	0298	06A0	6962			BL PSHREG	PUSH NEW NUMBER	00238
439	029C	C145				MOV R5,R5		00239
440	029E	132A				JEQ NONTR		00240
441	02A0	E820	3348	D972		SOC CLINE16,ROTFLLAG		00241
442	02A6	0720	DAC8			SETD OBUSY		00242
443	02AA	0720	DABA			SETD STPFLG	TURN OFF 'STOP IN' MESSAGE	00243

KEYTRE

00058

444	02AE	0820	D966	3834	C KEY,NTRKEY	WAS THIS 'ENTER' KEY?	00244
445	02B4	1311			JEQ XCUTD0N	YES, NUMBER IS ALREADY ENTERED THOUGH	00245
446			02B6		XCUTE EQU *		00246
447	02B6	0720	DABA		SETJ STPFLG	TURN OFF 'STOP IN' MESSAGE	00247
448	02BA	04E0	DAE6		CLR RQSNUM		00248
449	02BE	06A0	32AE		BL RQS	SEND SITUATION NORMAL STATUS BYTE	00249
450	02C2	C820	333A	D976	MOV C1,INUSE	SET BUSY FLAG	00250
451	02C8	0420	16B0		BLWP PRGSTAT	PUT 'BUSY' AND MNU ON CRT	00251
452	02CC	0300	000F		LIMI SF	TURN INTERRUPTS BACK ON	00252
453	02D0	0420	D946		BLWP KEYXFR		00253
454			02D4		XCUTSTOP EQU *		00254
455	02D4	0420	6C98		BLWP OPWCHG		00255
456			02D8		XCJTJON EQU *		00256
457	02D8	0420	141A		BLWP RDOUT		00257
458	02DC	C320	33C4		MOV BUSYLED,R12		00258
459	02E0	1E00			SBZ 0		00259
460	02E2	C320	D94A		MOV FATAL,R12	IS THERE AN ERROR?	00260
461	02E6	1110			JLT CHKWRN		00261
462	02E8	C020	D9BA		MOV PROGRS,R0	STOP NUMERIC ENTRY ON ERROR	00262
463	02EC	1303			JEQ NONTR		00263
464	02EE	04E0	D9BA		CLR PROGRS	AND ENTER THE NUMBER	00264
465	02F2	10C9			JMP NCHG2		00265
466					*NONTR MOV C1,MOVFLAG		00266 DEL
467					*		00017PATCH
468					* PROBLEM #14 - PATCH #17 (1 OF 2)		00017PATCH
469					*		00017PATCH
470					* 'RUN' AFTER ERROR DOESN'T INCREMENT TO NEXT COMMAND		00017PATCH
471					*		00017PATCH
472	02F4	1000			NONTR JMP *+2		00017PATCH
473					*		00017PATCH
474					* END OF PROBLEM #14		00017PATCH
475					*		00017PATCH
476	02F6	1001			JMP *+004		00017PATCH
477	02F8	1000			JMP *+002		00017PATCH
478	02FA	C320	DAE6		MOV RQSNUM,R12	CHECK FOR BAD GPIB COMMAND SRQ	00267
479	02FE	1607			JNE BEEP		00268
480	0300	C820	3342	DAE6	MOV C5,RQSNUM	IF NOT, THEN COMMAND ERROR SRQ	00269
481	0306	1003			JMP BEEP		00270
482	0308	C320	D94C		CHKWRN MOV WARNING,R12	IS THERE A WARNING?	00271
483	030C	1102			JLT NOBEEP		00272
484			030E		BEEP EQU *		00273
485	030E	0420	6CE8		BLWP LITERR		00274
486			0312		NOBEEP EQU *		00275
487	0312	0450	004C		B KEYTRE		00276

489					*			
490					*	SETLAST	---	SET LASTKEY EQUAL TO KEY IF KEY IS NOT
491					*			PART OF THE FORBIDDEN SET
492					*			
493					*	LEVEL	5	
494					*			
495			0316			SETLAST	EQU *	00278
496	0316	8820	0966	3832			C KEY,PANIC	00279
497	031C	130B					JEQ NOSET	00280
498	031E	8820	0966	37EE			C KEY,NXTKEY	00281
499	0324	1307					JEQ NOSET	00282
500	0326	8820	0966	3810			C KEY,PAUSKEY	00283
501	032C	1303					JEQ NOSET	00284
502	032E	0820	0966	0968			MOV KEY,LASTKEY	00285
503			0334			NOSET	EQU *	00286
504	0334	045B					B *R11	00287

```

506
507 *          BUTTONHIT --- HANDLE KEYBOARD INTERRUPT
508 *
509 0336 0300 0000          BUTTONHIT LIMIT 0          MASK OFF ALL INTERRUPTS          00289
510 033A C0A0 D906          MOV INT7854,R2          SHOULD THIS INTERRUPT BE HANDLED BY 00290
511 033E 1302          JEQ **6          A DIAGNOSTIC ROUTINE?          00291
512 0340 0420 F044          BLWP $F044          YES, TRANSFER CONTROL TO IT          00292
513 0344 0300 0007          LIMIT 7          00293
514 0348 C0A0 E00E          MOV KBCODE,R2          READ KEY CODE          00294
515 034C C320 3388          MOV KYBRST,R12          RESET KEYBOARD INTERRUPT          00295
516 0350 1E00          SBZ 0
517 0352 1000          SBJ 0          00296
518 0354 0242 00FF          ANJI R2,$FF          KEEP KEYCODE ONLY          00298
519 0358 1601          JNE **4          00299
520 035A 0502          INC R2          KEY00 = KEY01          00300
521 035C C802 D962          MOV R2,FKEY          00301
522 0360 C060 D942          MOV GPIBOPT,R1          IS GPIB OPTION INSTALLED?          00302
523 0364 110F          JLT LCLKEY          NO, EXECUTE LOCAL KEY          00303
524 0366 D820 337C E05C          MOV3 RTL,R3W          SEND LOCAL RTL MESSAGE TO INTERFACE 00304
525 036C C041          MOV R1,R1          DUMMY INSTRUCTION FOR MORE MC68488 CLOCKS 00305
526 036E D060 E044          MOV3 R1R,R1          READ ADDRESS COMMAND REGISTER          00306
527 0372 2060 3386          COC REM,R1          IS GPIB IN REMOTE          00307
528 0376 04E0 E06C          CLR R3W          REMOVE RTL MESSAGE          00308
529 037A 1504          JNE LCLKEY          IF LOCAL, EXECUTE KEY          00309
530 037C 8802 380C          C R2,RQSKEY          IF REMOTE, EXECUTE 'RQS' ANYWAY          00310
531 0380 1301          JEQ LCLKEY          00311
532 0382 0380          RTWP          00312
533 0384 0420 038A          LCLKEY BLWP BUFRIT          00313
534 0388 0380          RTWP          00314
    
```

536				*					
537				*	BUFRIT	---	PUT FKEY INTO KEY BUFFER		
538				*					
539	038A		D340		BUFRIT	WORD	WPBUFR		00316
540	038C		038E			WORD	*+2		00317
541	038E	0300	0007			LIMI	7		00318
542	0392	C020	DAA0			MOV	PROGRAM,R0		00319
543	0396	8820	D962	3832		C	FKEY,PANIC	IS THIS A PANIC KEY?	00320
544	039C	161F				JNE	BUFIN		00321
545	039E	C060	D94E			MOV	GOLD,R1	IS SHIFT KEY SET?	00322
546				*		JNE	BUFIN	IF SO, KEY IS 'PAUSE' NOT 'STOP'	00323 DEL
547				*		MOV	C1,HALTIT	SET HALT FLAG	00324 DEL
548				*					00006PATCH
549				*		PROBLEM #5 - PATCH #6 (1 OF 1)			00006PATCH
550				*					00006PATCH
551				*		'F STOP' FROM GPIB SUSPENDS COMMAND INPUT; IT SHOULDN'T.			00006PATCH
552				*					00006PATCH
553	03A2	0460	9674			B	PATCH6	BRANCH TO PATCH #6	00006PATCH
554			03A6		BACK6	EQU	*	DEFINE REENTRY POINT	00006PATCH
555				*					00006PATCH
556				*		END OF PROBLEM #5			00006PATCH
557				*					00006PATCH
558	03A6	1001				JMP	*+004		00006PATCH
559	03A8	1000				JMP	*+002		00006PATCH
560	03AA	0720	D94A			SETO	FATAL	IF STOP, NO ERROR	00325
561	03AE	C060	D976			MOV	INUSE,R1	IS A KEY CURRENTLY EXECUTING?	00326
562	03B2	1606				JNE	NOTIDL		00327
563	03B4	C000				MOV	R0,R0	CURRENTLY IN PROGRAM INPUT MODE?	00328
564	03B6	1312				JEQ	BUFIN	IF SO, BUFFER STOP IF KEY NOT EXECUTING	00329
565	03B8	02E0	DB00			LWPI	WPKYTRE	RESTORE KEYTRE'S WORKSPACE	00330
566	03BC	0460	004C			B	KEYTRE	GO BACK TO KEYTRE IF STOP IN IDLE MODE	00331
567	03C0	C0A0	D965		NOTIDL	MOV	KEY,R2		00332
568	03C4	20A0	3382			COC	STOPKEY,R2	CAN KEY BE STOPPED?	00333
569	03C8	1303				JEQ	HLTKEY		00334
570	03CA	C000				MOV	R0,R0	CURRENTLY IN PROGRAM INPUT MODE?	00335
571	03CC	1307				JEQ	BUFIN	IF NOT SAVE, THEN BUFFER STOP KEY	00336
572	03CE	0380				RTWP		NO	00337
573	03D0	0420	D946		HLTKEY	BLWP	KEYXFR	YES --- RETURN TO LT KEY CLEAN ITSELF UP	00338
574	03D4	02E0	DB00			LWPI	WPKYTRE		00339
575	03D8	0460	02D4			B	XOUTSTOP		00340
576			03D0		BJFIN	EQU	*		00341
577	03DC	C000				MOV	R0,R0	IS THE PROGRAM EXECUTING?	00342
578	03DE	1507				JGT	NOBUFR	IF NOT, DO NOT ACCEPT NON-PANIC KEYBOARD	00343
579	03E0	C020	D964			MOV	BUFAVL,R0	DOES BUFFER HAVE ROOM?	00344
580	03E4	1505				JGT	ROOMAVL		00345
581	03E6	04E0	D94C			CLR	WARNING		00346
582	03EA	0420	6CF4			BLWP	BUZZIT	NO	00347
583			03EE		NOBJFR	EQU	*		00348
584	03EE	0380				RTWP			00349
585			03F0		ROOMAVL	EQU	*		00350
586	03F0	C020	33E6			MOV	BUFRLEN,R0		00351
587	03F4	5020	D964			S	BUFAVL,R0	# WORDS USED	00352
588	03F8	0A10				SLA	R0,1		00353
589	03FA	0201	DA3A			LI	R1,KEYBUFR		00354
590	03FE	A04C				A	R0,R1	ADDRESS OF NEXT FREE BUFFER LOCATION	00355
591	0400	C460	D962			MOV	FKEY,*R1	STORE KEY	00356
592	0404	0620	D964			DEC	BUFAVL		00357
593	0408	0380				RTWP			00358

595				*				
596				*	POPBUFR	---	POP KEY BUFFER OR GET NEXT KEY FOR PROGRAM EXECUTION	
597				*				
598				*			RETURNS KEY=0 IF BUFFER IS EMPTY	
599				*				
600	040A		0B40		POPBUFR	WORD	WPBUFR	00360
601	040C		040E			WORD	*+2	00361
602	040E	0300	0007				LIMI 7	00362
603	0412	C820	DA3A	D966		MOV	KEYBUFR,KEY BUFFER EMPTY?	00363
604	0418	130E				JEQ	CHKPROG YES, CHECK FOR PROGRAM EXECUTING	00364
605	041A	C1E0	33E6			MOV	BUFLEN,R7	00365
606	041E	0208	DA3A			LI	R8,KEYBUFR	00366
607	0422	C308				MOV	R8,R12	00367
608	0424	05CC				INCR	R12	00368
609	0426	05A0	D964			INC	BUFAVL ONE MORE SPACE NOW AVAILABLE	00369
610			042A		POPLOP	EQU	*	00370
611	042A	0607				DEC	R7 POP BUFFER	00371
612	042C	1602				JNE	*+6	00372
613	042E	0408				CLR	*R8 CLEAR TOP ELEMENT	00373
614	0430	0380				RTWP	DONE	00374
615	0432	CE3C				MOV	*R12+,*R8+	00375
616	0434	10FA				JMP	POPLOP	00376
617			0436		CHKPROG	EQU	*	00377
618	0436	C020	DAA0			MOV	PROGRAM,R0	00378
619	043A	1501				JGT	*+4	00379
620	043C	0380				RTWP	NO PROGRAM TO EXECUTE	00380
621	043E	C020	DAAA			MOV	EDITPNT,R0	00381
622	0442	C060	DAB0			MOV	MOVFLAG,R1 SHOULD WE INCREMENT TO NEXT PROGRAMED COMMAND?	00382
623	0446	1501				JGT	*+4	00383
624	0448	0580				INC	R0	00384
625	044A	9810	3820			CB	*R0,ENDKEYB	00385
626	044E	1329				JEQ	STOPPRG	00386
627	0450	8800	DAE2			C	R0,MAXPRG	00387
628	0454	1526				JGT	STOPPRG	00388
629	0456	8820	DAA0	333C		C	PROGRAM,C2 IS IT IN SINGLE STEP MODE?	00389
630	045C	162E				JNE	PROGON	00390
631	045E	C060	D94C			MOV	WARNING,R1 HAS A WARNING OCCURRED?	00391
632	0462	1602				JNE	*+6	00392
633	0464	0720	DACE			SETJ	STEWARN ...YES, SAVE WARNING	00393
634	0468	C060	DACE			MOV	STEWARN,R1	00394
635	046C	1308				JEQ	NOWARN1	00395
636	046E	0202	3812			LI	R2,NJMTAB DON'T STOP DURING NUMERIC ENTRY	00396
637	0472	8812	3C36			C	*R2,ENJTAB	00397
638	0476	1310				JEQ	STESTOP IF NON-NUMERIC, STOP EXECUTION	00398
639	0478	0582				INC	R2	00399
640	047A	9C90				CB	*R0,*R2+	00400
641	047C	16FA				JNE	*-10	00401
642			047E		NOWARN1	EQU	*	00402
643	047E	8820	DAA8	DAB4		C	PROGLN,PROGSTEP YES, IS THIS THE SAME LINE WE START ON?	00403
644	0484	131A				JEQ	PROGON	00404
645	0486	C820	DAA8	DAB4		MOV	PROGLN,PROGSTEP MAKE IT SO, SO THE NEXT STEP WILL BE OK	00405
646	048C	C060	DAB8			MOV	STPROGF,R1 SHOULD NEXT LINE BE EXECUTED?	00406
647	0490	1114				JLT	PROGON ...YES, EXECUTE NEXT LINE ALSO	00407
648	0492	1302				JEQ	STESTOP	00408
649	0494	3720	DAB8			SETJ	STPROGF EXECUTE NEXT LINE ON RETURN	00409
650	0498	0720	DAA0		STESTOP	SETJ	PROGRAM	00410
651	049C	04E0	DACE			CLR	STEWARN	00411
652	04A0	0380				RTWP		00412
653			04A2		STOPPRG	EQU	*	00413

654	04A2	C820	333C	DAE6	MOV	C2,RQSNUM	SIGNAL GPIB THAT PROGRAM STOPPED	00414
655	04A8	C800	DAAA		MOV	R0,EDITPNT	UPDATE EDIT POINTER	00415
656	04AC	0720	DAA0		SET3	PROGRAM	RAN OFF END OF PROGRAM	00416
657	04B0	0720	DAC2		SET0	EOPFLG	CLEAR PL# IN DISPLAY AND PUT UP 'END' STATUS	00417
658	04B4	06A0	40D6		BL	CLRPTR	CLEAR RETURN POINTERS	00418
659	04B8	0380			RTH>			00419
660			04BA	PROG0N	EQU	*		00420
661	04BA	9810	37EF		CB	*R0,NXTKEYB	IF FOLLOWING KEY IS 'NEXT' DON'T CLEAR FLAG	00421
662	04BE	1302			JEQ	*+6		00422
663	04C0	04E0	DAB8		CLR	STPROGF	CLEAR EXECUTE NEXT LINE FLAG FOR STEP	00423
664	04C4	04E0	DAB0		CLR	MOVFLAG	CAUSE EDITPNT TO ADVANCE	00424
665	04C8	C800	DAAA		MOV	R0,EDITPNT	POINT TO PROPER PROGRAM KEY TO EXECUTE	00425
666	04CC	0810	D967		MOVB	*R0,KEYB		00426
667	04D0	0380			RTH>			00427

669			*		
670			*	TABLE OF KEYS NOT TO CHANGE NUMBER ON	
671			*		
672		04D2		NONUMCHG EQU *	00429
673	04D2	4970		WORD KEYRUN	00430
674	04D4	476A		WORD KEYSTEP	00431
675	04D6	4938		WORD KEYPULSE	00432
676	04D8	49D4		WORD KEYNOP	00433
677	04DA	649E		WORD KEY0	00434
678	04DC	49C2		WORD KEYF	00435
679	04DE	49D4		WORD KEYCLF	00436
680	04E0	FFFF		WORD \$FFFF	00437

```

682 *
683 *      INITILIZE ALL NECESSARY VARIABLES AND HARDWARE ON POWER-UP
684 *
685 *      LEVEL 0
686 *
687 *      COLDUP EQU *
688 04E2 0300 0007      LIM1 7      DISABLE SYSTEM INTERRUPTS DURING POWER-UP      00439
689 04E6 04E0 0A14      CLR  CYCFLAG      SET FLAG INDICATING NORMAL POWER-UP      00440
690 04EA 04E0 0905      CYCLEJP CLR  INT7854      7854 INTERRUPTS HANDLED BY THIS FIRMWARE      00441
691 *
692 *      INITALIZE THE DISPLAY AND ACQJIRE HARDWARE
693 *
694 04EE 04E0 E000      CLR  DMWRD      SET DISPLAY MODE WORD TO 'STOP'      00443
695 04F2 04E0 E00A      CLR  AMWRD      SET ACQUIRE MODE WORD TO 'STOP'      00444
696 04F6 02E0 0900      LWPI WPKYTR      00445
697 04FA 020B 6036      LI   R11,$6036      TURN LEDS ON, OTHERS TO 0      00446
698 04FE 04CC      CLR  R12      00447
699 0500 320B      LDCR R11,0      FIRST 0      00448
700 0502 06CB      SWPB R11      00449
701 0504 020C 0020      LI   R12,32      SECOND 0 ARE FOR DEBUG USE      00450
702 0508 320B      LDCR R11,0      THIRD 0      00451
703 *
704 *      CHECK IF BATTERY BACKUP POWER-UP
705 *      IF BATTERY BACKUP SKIP POWER UP TESTS
706 *
707 *      TWO WORDS ARE USE TO TEST FOR BATTERY BACKUP.
708 *      ONE WORD MUST BE THE INVERT OF THE OTHER AND
709 *      NO BYTE IN EITHER WORD CAN BE $00 OR $FF.
710 *
711 050A C820 3344 0AE6      MOV C6,RQSNUM      DEFAULT TO NORMAL POWER UP SRQ      00452
712 0510 C020 0903      MOV BACKUP1,R0      READ FIRST BATTERY BACKUP FLAG      00453
713 0514 C040      MOV R0,R1      00454
714 0516 0241 00FF      ANDI R1,$FF      LOW BYTE CANNOT BE $00 OR $FF      00455
715 051A 1312      JEQ NOBATTERY      00456
716 051C 0281 00FF      CI  R1,$FF      00457
717 0520 130F      JEQ NOBATTERY      00458
718 0522 C040      MOV R0,R1      00459
719 0524 0981      SRL R1,0      HIGH BYTE CANNOT BE $00 OR $FF      00460
720 0526 130C      JEQ NOBATTERY      00461
721 0528 0281 00FF      CI  R1,$FF      00462
722 052C 1309      JEQ NOBATTERY      00463
723 052E 0540      INV R0      00464
724 0530 8020 090A      C  BACKUP2,R0      SECOND FLAG MUST NOW BE INVERSE OF FIRST      00465
725 0534 1605      JNE NOBATTERY      00466
726 0536 C820 09C2 0E10      MOV FPTEMP,FP      RE-SETUP FRONT PANEL MODE      00467
727 053C 0460 08A0      B  HWDENBL      00468
728 0540 0720 E010      NOBATTERY SETO FP      00469
    
```

```

730 *****
731 **
732 ** RAM TEST FOR 2K, 4K, & 8K RAM SYSTEM **
733 ** **
734 ** POWER UP ROUTINE (LEVEL 0) **
735 ** **
736 ** NOTE --- **
737 ** THIS ROUTINE IS USED TO DETERMINE WHETHER THE 7854 IS **
738 ** CONFIGURED TO RUN AS A 2K, 4K, OR 8K RAM BASED SYSTEM. **
739 ** THE RAM IS ALSO VERIFIED USING BOTH AN ADDRESS TEST **
740 ** AND A WALKING ONES TEST. **
741 ** **
742 ** THE MEMORY IS THEN MAPPED ACCORDING TO THE AMOUNT OF **
743 ** RAM FOUND IN THE 7854. **
744 ** **
745 *****
746 0544 02E0 A000 TESTRAM LWPI RAMBASE USE BASE RAM TO CHECK TOP RAM 00471
747 0548 0201 D800 LI R1,RAMTOP 00472
748 054C 0281 D800 CI R1,RAMTOP QUICK CHECK OF WORKSPACE 00473
749 0550 162C JNE RAMFAIL 00474
750 0552 C820 DA14 A000 MOV CYCFLAG,RAMBASE SAVE CYCLE FLAG 00475
751 0558 06A0 05FC 3L RAMTEST TEST 1K OF RAM AT RAMTOP 00476
752 055C C820 A000 DA14 MOV RAMBASE,CYCFLAG RESTORE CYCLE FLAG 00477
753 0562 C1C7 MOV R7,R7 WAS RAM GOOD? 00478
754 0564 1122 JLT RAMFAIL NO, THEN RAM FAILURE 00479
755 0566 02E0 D800 LWPI WPKYTR YES, USE TOP RAM TO CHECK THE REST 00480
756 056A 0201 A000 LI R1,RAMBASE 00481
757 056E 0202 0008 LI R2,MAXKRAM LOAD MAXIMUM K OF RAM 00482
758 0572 0602 DEC R2 1K ALREADY TESTED 00483
759 0574 04C3 CLR R3 00484
760 0576 06A0 05FC TEST1K BL RAMTEST TEST NEXT 1K OF RAM 00485
761 057A 0913 SRL R3,1 SHIFT RAM STATUS WORD 00486
762 057C C1C7 MOV R7,R7 WAS THIS BLOCK OF RAM GOOD? 00487
763 057E 1102 JLT *+6 NO, STATUS BIT = 0 00488
764 0580 0223 8000 AI R3,$8000 YES, STATUS BIT = 1 00489
765 0584 0221 0800 AI R1,2048 ADVANCE ADDRESS TO NEXT 1K BLOCK OF RAM 00490
766 0588 0602 DEC R2 CHECK REMAINING BLOCKS 00491
767 058A 15F5 JGT TEST1K 00492
768 058C 0913 SRL R3,1 00493
769 058E 0223 8000 AI R3,$8000 FLAG INDICATING TOP RAM GOOD 00494
770 0592 C820 3344 JAE6 MOV C6,RQSNUM POWER ON SERVICE REQUEST 00495
771 0598 8803 34CC C R3,SYS2K 2K SYSTEM? 00496
772 059C 130C JEQ IS2K 00497
773 059E 8803 34CE C R3,SYS4K 4K SYSTEM? 00498
774 05A2 130C JEQ IS4K 00499
775 05A4 8803 34D0 C R3,SYS8K 8K SYSTEM? 00500
776 05A8 130C JEQ IS8K 00501
777 05AA C820 34D2 E010 RAMFAIL MOV PWRERR1,FP SET FRONT PANEL BUTTONS TO RAM ERROR 00502
778 05B0 06A0 0734 BL PWRSRQ SETUP POWER-UP ERROR SRQ 00503
779 05B4 1003 JMP IS4K DEFAULT TO 4K SYSTEM 00504
    
```

```

781 *****
782 **
783 ** COPY APPROPRIATE RAM MAP DEPENDING ON RAM SIZE **
784 **
785 *****
786 05B6 0201 348A IS2K LI R1,MAP2 TABLE FOR 2K RAM MAP 00506
787 05BA 1005 JMP MAPMEM 00507
788 05BC 0201 34A0 IS4K LI R1,MAP4 TABLE FOR 4K RAM MAP 00508
789 05C0 1002 JMP MAPMEM 00509
790 05C2 0201 3486 IS8K LI R1,MAP8 TABLE FOR 8K RAM MAP 00510
791 05C6 0202 0A00 MAPMEM LI R2,RAMMAP ADDRESS OF RAM MEMORY MAP 00511
792 05CA 0203 0016 LI R3,MAPSIZE NUMBER OF BYTES IN MAP 00512
793 05CE 0913 SRL R3,1 DIVIDE BY 2 TO GET NUMBER OF WORDS IN MAP 00513
794 05D0 CC31 MOV *R1+,*R2+ SETUP SYSTEM MEMORY MAP 00514
795 05D2 0603 DEC R3 00515
796 05D4 15FD JGT *-4 00516
    
```

```

798 *****
799 **
800 ** ROM TEST FOR 16K BY 16 ROM SYSTEM **
801 ** ** **
802 ** POWER UP ROUTINE (LEVEL 0) **
803 ** ** **
804 ** NOTE --- **
805 ** THIS ROUTINE VERIFIES THE 4 ROMS IN THE 7854 SYSTEM. **
806 ** THE ROMS ARE 64K BIT (8K BY 8) ROMS WHICH ARE MAPPEJ **
807 ** INTO A 16K BY 16 SYSTEM. THIS RESULTS IN THE ROMS **
808 ** CONTRIBUTING EVERY OTHER BYTE OF INFORMATION SO THE **
809 ** TEST WAS DESIGNED TO CHECK 1 ROM AT A TIME I.E. **
810 ** USING ALTERNATE BYTES. **
811 ** **

```

```

812 *****
813 05D6 0201 0000 TESTROM LI R1,ROMBASE CHECK BASE ROMS 00518
814 05DA 06A0 0668 BL ROMTEST 00519
815 05DE C1C7 MOV R7,R7 DID ROM VERIFY? 00520
816 05E0 1606 JNE ROMFAIL 00521
817 05E2 0201 4000 LI R1,ROMTOP CHECK TOP ROMS 00522
818 05E6 06A0 0668 BL ROMTEST 00523
819 05EA C1C7 MOV R7,R7 DID ROM VERIFY? 00524
820 05EC 134D JEQ TESTCLK 00525
821 05EE C820 3404 E010 ROMFAIL MOV PWRERR2,FP SET FRONT PANEL BUTTONS TO ROM ERROR 00526
822 05F4 06A0 0734 BL PWRSRQ SETUP POWER-UP ERROR SRQ 00527
823 05F8 0450 0688 B TESTCLK 00528

```

```

825 *****
826 **
827 ** RAM TEST - ADDRESS & WALKING ONES (1K BY 16) **
828 **
829 ** POWER UP ROUTINE (LEVEL 5) **
830 **
831 ** INPUT: R1 - STARTING ADDRESS OF 1K BLOCK OF RAM **
832 ** OUTPUT: R7 - ZERO IF TEST GOOD **
833 ** NEGATIVE IF TEST BAD **
834 ** DESTROYS: R7-R10 **
835 **
836 ** NOTE --- **
837 ** THIS ROUTINE TESTS 1K OF RAM USING BOTH AN ADDRESS **
838 ** TEST AND A WALKING ONES TEST. FOR THE ADDRESS TEST **
839 ** THE ADDRESS OF EACH WORD OF RAM IS WRITTEN INTO ITSELF **
840 ** IN ONE PASS THROUGH THE RAM. A SECOND PASS THROUGH **
841 ** THE RAM IS USED TO VERIFY THAT ADDRESSES CAN BE READ **
842 ** CORRECTLY. THE WALKING ONES TEST WRITES A PATTERN **
843 ** WHICH RESEMBLES A SINGLE 1 BIT WALKING THROUGH THE 16 **
844 ** BITS OF EACH WORD. THE PATTERN IS WRITTEN INTO THE **
845 ** ENTIRE 1K BEFORE THE VERIFICATION LOOP. THE WALKING **
846 ** ONES TEST TAKES 16 PASSES THROUGH RAM TO PERFORM A **
847 ** COMPLETE CHECK. **
848 **
849 **
850 *****
    
```

```

851
852 ***** PERFORM A QUICK CHECK TO WEED OUT NON-EXISTANT RAM *****
853
854 05FC 04F1 RAMTEST CLR *R1+ SET FIRST WORD OF 1K RAM BLOCK TO $3000 00530
855 05FE 0711 SETD *R1 SET SECOND WORD OF 1K RAM BLOCK TO $FFFF 00531
856 0600 0641 DECT R1 00532
857 0602 8831 3338 C *R1+,C0 CHECK IF FIRST WORD IS $0000 00533
858 0606 1628 JNE RAMBAD IF NOT, THEN 1K BLOCK OF RAM IS BAD 00534
859 0608 8811 3336 C *R1,CN1 CHECK IF SECOND WORD IS $FFFF 00535
860 060C 1625 JNE RAMBAD IF NOT, THEN 1K BLOCK OF RAM IS BAD 00536
861 060E 0641 DECT R1 00537
862
    
```

```

863 ***** PERFORM ADDRESS TEST ON RAM *****
864
865 0610 C1C1 MOV R1,R7 STARTING ADDRESS OF 1K BLOCK OF RAM 00538
866 0612 0208 0400 LI R8,1024 CHECK NEXT 1K WORDS 00539
867 0616 CDC7 MOV R7,*R7+ WRITE ADDRESS OF WORD INTO WORD 00540
868 0618 0608 DEC R8 DO THIS FOR ENTIRE 1K 00541
869 061A 15FD JGT *-4 00542
870 061C C1C1 MOV R1,R7 RESET ADDRESS TO START OF 1K 00543
871 061E 0208 0400 LI R8,1024 00544
872 0622 80C7 C R7,*R7+ CHECK IF ADDRESSES ARE CORRECT 00545
873 0624 1619 JNE RAMBAD IF NOT, THEN BAD RAM 00546
874 0626 0608 DEC R8 CHECK ENTIRE 1K 00547
875 0628 15FD JGT *-6 00548
876
    
```

```

877 ***** PERFORM A WALKING ONES TEST ON RAM *****
878
879 062A 0209 8000 LI R9,$8000 LOAD INITIAL PATTERN 00549
880 062E C1C1 WALKONE MOV R1,R7 STARTING ADDRESS OF 1K BLOCK OF RAM 00550
881 0630 0208 0400 LI R8,1024 CHECK 1K OF RAM 00551
882 0634 C289 MOV R9,R10 00552
883 0636 C0CA MOV R10,*R7+ MOVE PATTERN TO NEXT RAM WORD 00553
    
```


RAM TEST (1K BY 16)

00529

884	0638	0B1A		SRC	R10,1	SHIFT PATTERN BY WALK TO GET WALKING EFFECT	00554
885	063A	060B		DEC	R8	FILL ENTIRE 1K BLOCK	00555
886	063C	15FC		JGT	*-6		00556
887	063E	C1C1		MOV	R1,R7	RESET ADDRESS TO START OF RAM BLOCK	00557
888	0640	020B	0400	LI	R8,1024		00558
889	0644	C2B9		MOV	R9,R10	RESET PATTERN TO INITIAL VALUE	00559
890	0646	80CA		C	R10,*R7+	CHECK IF PATTERN IS CORRECT IN RAM	00560
891	0648	1607		JNE	RAMBAD	IF NOT, RAM IS BAD	00561
892	064A	0B1A		SRC	R10,1	SHIFT PATTERN	00562
893	064C	060B		DEC	R8		00563
894	064E	15FB		JGT	*-8		00564
895	0650	0919		SRL	R9,1	SHIFT INITIAL PATTERN TO CHECK ALL 16 BITS	00565
896	0652	16ED		JNE	WALKJNE		00566
897	0654	04C7		CLR	R7	RAM TESTS SUCCESSFUL	00567
898	0656	1001		JMP	*+4		00568
899	0658	0707	RAMBAD	SETJ	R7	RAM TESTS BAD	00569
900	065A	C201		MOV	R1,R8	STARTING ADDRESS OF 1K RAM BLOCK	00570
901	065C	0209	0400	LI	R9,1024		00571
902	0660	04F8		CLR	*R8+	SET 1K RAM BLOCK TO ALL \$0000	00572
903	0662	0609		DEC	R9		00573
904	0664	15FD		JGT	*-4		00574
905	0666	045B		B	*R11		00575

ROM TEST (8K BY 16)

00576

```

907 *****
908 **
909 ** ROM TEST - SHIFT & ADD WITH CARRY CHECKSUM (8K BY 16) **
910 **
911 ** POWER UP ROUTINE (LEVEL 5) **
912 **
913 ** INPUT: R1 - STARTING ADDRESS OF 64K ROM **
914 ** OUTPUT: R7 - ZERO IF ROM TESTED GOOD **
915 **          NEGATIVE IF ROM TESTED BAD **
916 ** DESTROYS: R7-R10 **
917 **
918 ** NOTE --- **
919 ** THIS ROUTINE TESTS 8K OF ROM (8K BY 16) BY USING A **
920 ** SHIFT AND ADD WITH CARRY CHECKSUM. THE SHIFT IS **
921 ** A RIGHT CIRCULAR SHIFT OF 3 BITS. **
922 **
923 *****
924 0668 C1C1 ROMTEST MOV R1,R7 STARTING ADDRESS OF 64K ROM 00577
925 066A 0208 1FFF LI R8,8191 CHECKSUM ALL BUT LAST WORD 00578
926 066E 04CA CLR R10 INITIALIZE CHECKSUM TO 0 00579
927 0670 0B3A ROMSUM SRC R10,3 SHIFT CHECKSUM RIGHT CIRCULAR 3 BITS 00580
928 0672 A237 A *R7+,R10 ADD NEXT WORD TO CHECKSUM WITH CARRY 00581
929 0674 1701 JNC *+4 00582
930 0676 058A INC R10 ADD CARRY 00583
931 0678 0608 DEC R8 00584
932 067A 15FA JGT ROMSUM 00585
933 067C 85CA C R10,*R7 COMPARE CHECKSUM WITH STORED VALUE 00586
934 067E 1602 JNE ROMBAD BAD ROM IF NOT EQUAL 00587
935 0680 04C7 CLR R7 ROM TESTED GOOD 00588
936 0682 045B B *R11 00589
937 0684 0707 ROMBAD SETJ R7 ROM TESTED BAD 00590
938 0686 045B B *R11 00591

```

TEST REALTIME CLOCK

00592

```

940 *****
941 **
942 ** TEST THE REALTIME CLOCK **
943 ** **
944 ** THE REALTIME CLOCK IS TESTED TO OPERATE BETWEEN 15 & 20 **
945 ** MILLISECONDS. **
946 ** **
947 *****
948 0688 02E0 0C00 TESTCLK LWPI WPCLK INITIALIZE CONTROL VARIABLES SO NO FRONT 00593
949 068C 0204 000A LI R4,10 PANEL KEY IS EXECUTED AND NO DISPLAY 00594
950 0690 0205 FFF5 LI R5,-10 IS GENERATED DURING TEST 00595
951 0694 04C6 CLR R6 00596
952 0696 02E0 0B00 LWPI WPKYTR 00597
953 069A 0720 0942 SETD GPIBOPT SET FLAG INDICATING NO GPIB OPTION 00598
954 069E 0720 09C4 SETD REFRESH SET FLAG INDICATING NO REFRESH OF DISPLAY 00599
955 06A2 04E0 0938 CLR INTFLAG CLEAR INTERRUPT FLAGS 00600
956 06A6 C320 33BC MOV CLKRST,R12 00601
957 06AA 1D00 SBZ 0 ENABLE 20 MILLISECOND CLOCK INTERRUPT 00602
958 06AC 0300 0009 LIMIT 9 00603
959 06B0 0202 07D0 LI R2,2000 WAIT FOR 40 MILLISECONDS FOR FIRST INTERRUPT 00604
960 06B4 C060 0938 WAIT1 MOV INTFLAG,R1 GET INTERRUPT FLAGS 00605
961 06B8 2060 337A COC CLKINT,R1 HAS THE 20 MILLISECOND INTERRUPT OCCURED? 00606
962 06BC 1303 JEQ FRSTCLK YES, CHECK TIME UNTIL SECOND CLOCK 00607
963 06BE 0602 DEC R2 DECREMENT TIMING REGISTER 00608
964 06C0 15F9 JGT WAIT1 00609
965 06C2 100F JMP CLKFAIL IF NO CLOCK IN 40 MILLISECONDS THEN ERROR 00610
966 06C4 0202 0341 FRSTCLK LI R2,833 WAIT MAX OF 20 MILLISECONDS FOR SECOND CLOCK 00611
967 06C8 04E0 0938 CLR INTFLAG CLEAR INTERRUPT FLAGS 00612
968 06CC C060 0938 WAIT2 MOV INTFLAG,R1 GET INTERRUPT FLAGS 00613
969 06D0 2060 337A COC CLKINT,R1 HAS THE 20 MILLISECOND INTERRUPT OCCURED? 00614
970 06D4 1303 JEQ SCNDCLK YES, CHECK MINIMUM TIME 00615
971 06D6 0602 DEC R2 DECREMENT TIMING REGISTER 00616
972 06D8 15F9 JGT WAIT2 00617
973 06DA 1003 JMP CLKFAIL IF NO CLOCK IN 24 MILLISECONDS THEN ERROR 00618
974 *SCNDCLK CI R2,208 MINIMUM TIME IS 15 MILLISECONDS 00619 DEL
975 * 00012PATCH
976 * PROBLEM #11 - PATCH #12 (1 OF 1) 00012PATCH
977 * 00012PATCH
978 * MINIMUM TEST TIME FOR THE REALTIME CLOCK SHOULD BE 10 MILLISECONDS 00012PATCH
979 * 00012PATCH
980 06DC 0282 01A0 SCNDCLK CI R2,416 MINIMUM TIME IS 10 MILLISECONDS 00012PATCH
981 * 00012PATCH
982 * END OF PROBLEM #11 00012PATCH
983 * 00012PATCH
984 06E0 1105 JLT CLKGOOD 00620
985 06E2 C820 34D6 ED10 CLKFAIL MOV PWRERR3,FP SET FRONT PANEL BUTTONS TO CLOCK ERROR 00621
986 06E8 06A0 0734 BL PWRSRQ SETUP POWER-UP ERROR SRQ 00622
987 06EC 0300 0007 CLKGOOD LIMIT 7 MASK OFF SYSTEM INTERRUPTS 00623
988 06F0 C320 33BC MOV CLKRST,R12 00624
989 06F4 1E00 SBZ 0 DISABLE 20 MILLISECOND INTERRUPT 00625

```

TEST DISPLAY DONE INTERRUPT

00626

```

991 *****
992 **
993 ** TEST DISPLAY DONE INTERRUPT **
994 ** **
995 ** SETTING THE CRU LINE 'STPDSY', STOP DISPLAY, SHOULD **
996 ** ACTIVATE THE DISPLAY DONE INTERRUPT. **
997 ** **
998 *****
999 06F6 04E0 0938 TESTDSP CLR INTFLAG CLEAR INTERRUPT FLAGS 00627
1000 06FA 04E0 09C8 CLR DSPRLT NO DISPLAY DURING TEST 00628
1001 06FE C820 3382 D900 MOV MWACQR,ACQWFM MAKE SYSTEM THINK 2 WAVEFORM ACQUIRE 00629
1002 0704 E820 337E D900 SOC MW2WFM,ACQWFM IS OCCURING TO FORCE NO DISPLAY 00630
1003 070A C320 3332 MOV DSPRST,R12 00631
1004 070E 1000 SBZ 0 ENABLE DISPLAY DONE INTERRUPT, SINCE DSPSTP 00632
1005 0710 0300 000F LIM1 $F IS LOW AN INTERRUPT SHOULD OCCUR IMMEDIATLY 00633
1006 0714 0300 0007 LIM1 7 00634
1007 0718 1E00 SBZ 0 DISABLE DISPLAY DONE INTERRUPT 00635
1008 071A 04E0 0900 CLR ACQWFM 00636
1009 071E C050 0938 MOV INTFLAG,R1 GET INTERRUPTS FLAGS 00637
1010 0722 2060 3382 COC DSPINT,R1 HAS DISPLAY DONE INTERRUPT OCCURED? 00638
1011 0726 1314 JEQ POWERUP YES, POWER UP TEST SAYS ALL'S WELL 00639
1012 0728 C820 3438 E010 DSPFAIL MOV PWRERR4,FP SET FRONT PANEL BUTTONS TO DISPLAY ERRJR 00640
1013 072E 06A0 0734 BL PWRSRQ SETUP POWER-UP ERROR SRQ 00641
1014 0732 100E JMP POWERUP 00642

```

WAIT ON POWER UP ERROR

00643

```

1016 *****
1017 **
1018 **      WAIT FOR FRONT PANEL MODE BUTTON TO BE HIT TO CONTINUE      **
1019 **      POWER UP TEST.                                               **
1020 **
1021 *****
1022 0734 C820 3346 DAES PWRSRQ MOV C7,RQSNUM      POWER ON CHECK FAILURE SRQ      00644
1023 073A C060 E010      MOV FP,R1      WAIT FOR FRONT PANEL BUTTON TO BE PRESSED 00645
1024 073E 2060 34DA      CJC NOFPKEY,R1      00646
1025 0742 13FB          JEQ *-8      00647
1026 0744 C060 E010      MOV FP,R1      WAIT FOR FRONT PANEL BUTTON TO BE LET JP 00648
1027 0748 2060 34DA      CJC NOFPKEY,R1      00649
1028 074C 16FB          JNE *-8      00650
1029 074E 045B          B *R11      00651
    
```

INITIALIZE VARIABLES

00652

```

1031 *****
1032 **
1033 ** IF POWER UP IS FIRST TIME POWER UP (NON-BATTERY BACKUP) **
1034 ** INITIALIZE ALL VARIABLES. **
1035 **
1036 *****
1037 0750 C820 33F0 E010 POWERUP MOV LEFTB,FP SET FRONT PANEL BUTTONS TO LEFT-B 00653
1038 0756 C820 3350 DA25 MOV C12,VERTM 00654
1039 075C C820 3352 DA28 MOV C13,HORZM 00655
1040 0762 0201 AAAA LI R1,$AAAA CHECK IF DIAGNOSTICS RAM IS AVAILABLE 00656
1041 0766 C801 E500 MOV R1,DIAG 00657
1042 076A 8801 E500 C R1,DIAG 00658
1043 076E 1603 JNE **8 00659
1044 0770 C820 333A D904 MOV C1,DIAGINT YES, SEND UNDEFINED INTERRUPTS TO DIAGNOSTICS 00660
1045 0776 C260 33DE MOV SOFTST,SOFT 00661
1046 077A C050 DAE4 MOV RAMOPT,R1 LOAD RAM OPTION 00662
1047 077E C831 D970 MOV *R1+,RESOLV GET SMALLEST RESOLUTION 00663
1048 0782 C051 MOV *R1,R1 MAXIMUM WAVEFORMS AT SMALLEST RESOLUTION 00664
1049 0784 0601 DEC R1 MAXIMUM WAVEFORM # 00665
1050 0786 C801 D974 MOV R1,WFMAXN 00666
1051 078A 0649 CLERWFM DECT SOFT 00667
1052 078C C641 MOV R1,*SOFT 00668
1053 078E 06A0 6F25 BL NULLWFM NULL ALL WAVEFORMS 00669
1054 0792 0601 DEC R1 00670
1055 0794 15FA JGT CLERWFM 00671
1056 0796 13F9 JEQ CLERWFM 00672
1057 0798 C820 337A D970 MOV FRSTRES,RESOLV 00673
1058 079E 06A0 12BE BL CLRTXT INITIALIZE TEXT DISPLAY 00674
1059 07A2 0200 D360 LI R0,WPLVL1 SET UP VARIABLE DEFAULT VALUES 00675
1060 07A6 C800 D946 MOV R0,KEYXFR 00676
1061 07AA 0720 D94C SETJ WARNING 00677
1062 07AE C820 334C D954 MOV C10,BUFAVL 00678
1063 07B4 C820 D970 D958 MOV RESOLV,CURS2 00679
1064 07BA 0620 D958 DEC CURS2 00680
1065 07BE C820 DA06 D95C MOV WFMBAS,JPWFMD 00681
1066 07C4 C820 DA04 D95E MOV WOHEAD,JPWFMD 00682
1067 07CA 0720 D958 SETJ LASTKEY 00683
1068 07CE 0720 DAA8 SETJ PROGLN SET UP NO PROGRAM 00684
1069 07D2 C820 333A DAB0 MOV C1,MOVFLAG 00685
1070 07D8 0720 DAA2 SETJ PROMODE 00686
1071 07DC 0720 DAAE SETJ LINE99 00687
1072 07E0 C020 DAED MOV PROGHEM,R0 00688
1073 07E4 C800 DAAA MOV R0,EDITPNT 00689
1074 07E8 C050 DAE2 MOV MAXPROG,R1 00690
1075 07EC 0221 0003 AI R1,3 00691
1076 07F0 DC20 3820 MOVB ENDKEYB,*R0+ 00692
1077 07F4 8040 C R0,R1 00693
1078 07F6 1AFC JL *-5 00694
1079 07F8 0720 DAA8 SETJ PROGRAM POWER UP TO REAL-TIME DISPLAY 00695
1080 07FC C020 DAE4 MOV RAMOPT,RJ 00696
1081 0800 8830 D970 C *R0+,RESOLV 00697
1082 0804 1302 JEQ **6 00698
1083 0806 05C0 INCT R0 00699
1084 0808 10FB JMP *-8 00700
1085 080A C810 D974 MOV *R0,WFMAXN 00701
1086 080E 0620 D974 DEC WFMAXN 00702
1087 0812 04E0 D978 CLR XYWFM 00703
1088 0816 C020 DA04 MOV WOHEAD,R0 TURN ON WFM 0 00704
1089 081A A020 3354 A DISPLA,R0 00705

```

```

1090 081E C420 333C          MOV C2,*R0          00706
1091 0822 C820 33F0 0922    MOV LEFT3,FPTEMP   COPY OF FRONT PANEL BUTTON SETTINGS 00707
1092 0828 C820 3394 097A    MOV CHE002,YTWM X PRELOAD FOR Y-T WAVEFORMS 00708
1093 082E C820 339C 097C    MOV CHE022,YTWM+2 INTERLACE FOR Y-T WAVEFORMS 00709
1094 0834 C820 33A4 09C8    MOV DSPRLTI,DSPRLT 00710
1095 083A C820 33AA 09CA    MOV DSPRJI,DSPRO  INITIALIZE ACQUIRE/DISPLAY CONTROL WORDS 00711
1096 0840 C820 33A8 09CC    MOV DSPCRSI,DSPCRS 00712
1097 0846 C820 33A6 09CE    MOV DSPWFI,DSPWFM  00713
1098
1099 **
1100 ** ON FIRST POWER UP TWO WORDS IN RAM ARE INITIALIZED TO BE **
1101 ** USED AS FLAGS FOR BATTERY BACKUP POWER UPS. **
1102 **
1103 ** THESE TWO WORDS ARE RANDOMLY GENERATED USING BYTES READ **
1104 ** FROM THE 7854'S KEYBOARDS. THESE BYTES ARE RANDOM SINCE **
1105 ** THE KEYBOARDS ARE BEING CONTINUOUSLY SCANNED FOR A KEY **
1106 ** CLOSURE. READING THE KEYCODE FROM THE KEYBOARDS READS THE **
1107 ** KEY CURRENTLY BEING SCANNED. THE KEYCODES $00 AND $FF ARE **
1108 ** NOT USED. TWO BYTES ARE READ INTO THE FIRST WORD AND THE **
1109 ** ONE'S COMPLIMENT OF THIS WORD (INVERT) IS PLACED IN THE **
1110 ** SECOND WORD. **
1111 **
1112 *****
1113 084C 0420 6CF4          BLWP BUZZIT        BEEP SPEAKER          00714
1114          BATSET1 EQU *          0050          00715
1115 0850 C060 E00E          MOV KBCODE,R1     READ FIRST BYTE (NOT $00 OR $FF) 00716
1116 0854 0241 00FF          ANDI R1,$FF
1117 0858 13FB          JEQ BATSET1          00718
1118 085A 0281 00FF          CI R1,$FF          00719
1119 085E 13F8          JEQ BATSET1          00720
1120 0860 0420 6CF4          BLWP BUZZIT        BEEP SPEAKER          00721
1121          BATSET2 EQU *          0864          00722
1122 0864 C0A0 E00E          MOV KBCODE,R2     READ SECOND BYTE (NOT $00 OR $FF) 00723
1123 0868 0242 00FF          ANDI R2,$FF          00724
1124 086C 13FB          JEQ BATSET2          00725
1125 086E 0282 00FF          CI R2,$FF          00726
1126 0872 13F8          JEQ BATSET2          00727
1127 0874 0A81          SLA R1,8           00728
1128 0876 E042          SDC R2,R1         PUT BOTH BYTES IN FIRST FLAG 00729
1129 0878 C801 D908          MOV R1,BACKUP1
1130 087C 0541          INV R1             PUT INVERSE OF FIRST FLAG IN SECOND FLAG 00731
1131 087E C801 D90A          MOV R1,BACKUP2          00732
1132 *****
1133 **
1134 ** DISPLAY 'SELF TEST COMPLETE' MESSAGE ON CRT **
1135 **
1136 *****
1137 0882 0649          DECT SOFT          00733
1138 0884 C650 3354          MOV C14,*SOFT     DISPLAY MESSAGE ON LINE 14 00734
1139 0888 0649          DECT SOFT          00735
1140 088A C660 333A          MOV C1,*SOFT     START MESSAGE IN COLUMN 1 00736
1141 088E 0649          DECT SOFT          00737
1142 0890 0204 098C          LI R4,VERSION     00738
1143 0894 C644          MOV R4,*SOFT     PUT ADDRESS OF MESSAGE ON SOFTSTACK 00739
1144 0896 0649          DECT SOFT          00740
1145 0898 C660 335C          MOV C18,*SOFT    LENGTH OF MESSAGE IS 18 CHARACTERS 00741
1146 089C 06A0 136C          BL TEXT           00742
1147 *****
1148 **

```

```

1149          **      ANY POWER UP MUST INITIALIZE THE HARDWARE      **
1150          **
1151          ****
1152 08A0 02E0 DC00 HADENBL LWPI WPCLK          INITIALIZE 20 MILLISECOND CLOCK VARIABLES 00743
1153 08A4 04C4          CLR R4                      00744
1154 08A6 04C5          CLR R5                      00745
1155 08A8 04C6          CLR R6                      00746
1156 08AA 02E0 DB00 LWPI WPKYTR                     00747
1157 08AE 0209 C4C0 LI R11,$04C0          CRU MASK          00748
1158 08B2 C320 D9C6 MOV REALTIME,R12      CHECK IF IN 'STORED' MODE 00749
1159 08B6 1102          JLT *+6                      00750
1160 08B8 020B 04C0 LI R11,$04C0          IF NOT, TURN OFF STORED WFM LED 00751
1161          *
1162          *      STORE THE ROM AND PROM PATCH VERSION
1163          *      FOR THE GPIB ID KEY
1164          *
1165 08BC 0201 DA04 LI R1,VRSNUM          GET STORE ADDRESS      00752
1166 08C0 CC60 3FE8 MOV ROMV,*R1+          STORE FIRST DIGIT      00753
1167 08C4 DC60 372F MOV3 DECPNT,*R1+        STORE DECIMAL POINT    00754
1168 08C8 0202 3FF8 LI R2,PROMV          GET ADDRESS OF PROM PATCH VERSION 00755
1169          *      MOV3 *R2+,*R1+          STORE FIRST DIGIT      00756 DEL
1170          *      MOV3 *R2,*R1+          STORE SECOND DIGIT    00757 DEL
1171          *
1172          *      PROBLEM #19 - PATCH #24
1173          *
1174          *      CORRECT PROM VERSION DISPLAY
1175          *
1176 08CC 0460 97FD          B PATCH24          BRANCH TO PATCH #24    00024PATCH
1177          BACK24 EQU *          DEFINE REENTRY POINT      00024PATCH
1178          *
1179          *      END OF PROBLEM #19
1180          *
1181 08D0 DC60 3321 MOV3 COMMB,*R1+
1182 08D4 C0A0 DA00 MOV KRAM,R2          GET AMJNT OF RAM IN SYSTEM - 2, + OR BK 00759
1183 08D8 0222 0030 AI R2,$0030          CONVERT NUMBER TO ASCII 00760
1184 08DC 0A82          SLA R2,8                      00761
1185 08DE DC42          MOV3 R2,*R1+          00762
1186 08E0 0202 4B00 LI R2,$4B00          MOVE AN ASCII 'K' INTO HIGH ORDER BYTE OF R2 00763
1187 08E4 D442          MOV3 R2,*R1          00764
1188          *
1189          *      INITIALIZE THE GPIB
1190          *
1191 08E6 0720 D942 SET3 GPIBOPT          SET FLAG INDICATING NO GPIB OPTION 00765
1192 08EA 8820 DA00 333C C KRAM,C2          IS ONLY 2K OF RAM AVAILABLE? 00766
1193 08F0 1322          JEQ ENABLE          YES, THEN NO GPIB      00767
1194 08F2 D820 3390 E05C MOV3 RESET,R3W        SET SOFTWARE MC68400 GPIA RESET 00768
1195 08F8 0720 E070 SET3 R4W          SET GPB ADDRESS TO 31 00769
1196 08FC 9820 3338 E054 CB C0,R5R          WHEN GPIB IS RESET SERIAL POLL REGISTER = 0 00770
1197 0902 1619          JNE ENABLE          00771
1198 0904 04E0 E06C CLR R3W          ENABLE GPIA          00772
1199 0908 D820 338C E074 MOV3 CH3FFF,R5W        SET SERIAL POLL REGISTER TO $3F 00773
1200 090E 9820 338C E054 CB CH3FFF,R5R        DID THIS VALUE GET LOADED? 00774
1201 0914 1619          JNE ENABLE          NO, THEN NO GPIB BOARD 00775
1202 0916 D820 3390 E06C MOV3 RESET,R3W        SET SOFTWARE GPIB CHIP RESET 00776
1203 091C 04E0 D942 CLR GPIBOPT          SET FLAG INDICATING GPIB PC BOARD IN 00777
1204 0920 C820 3334 D91E MOV CN2,RSVENBL      ENABLE ALL SERVICE REQUESTS 00778
1205 0926 D060 E050 MOV3 R4R,R1          READ 7854'S GPIB ADDRESS 00779
1206 092A 0541 INV R1          FORCE INITIALIZATION OF GPIB BY 20 00780
1207 092C C801 D91A MOV R1,GPIBADR          MILLISECOND CLOCK 00781
    
```


1208	0930	C820	333A	D976	MOV	C1,INUSE	SET BUSY FLAG FOR POWER ON SRQ	00782
1209					*			
1210					*	INITIALIZE HARDWARE INTERRUPTS		
1211					*			
1212	0936	04CC			ENABLE	CLR R12		00783
1213	0938	320B			LDCR	R11,8	FIRST 8	00784
1214	093A	06CB			SWPB	R11		00785
1215	093C	020C	0020		LI	R12,32	SECOND 8 ARE FOR DEBUG USE	00786
1216	0940	320B			LDCR	R11,8	THIRD 8	00787
1217	0942	04E0	D938		CLR	INTFLAG		00788
1218	0946	0300	000F		LIMI	\$F		00789
1219	094A	0340			CLKWAIT	IDLE	WAIT FOR 20 MILLISECOND CLOCK TO	00790
1220	094C	C320	D938		MOV	INTFLAG,R12	COMPLETE INITIALIZATION	00791
1221	0950	2320	337A		COC	CLKINT,R12		00792
1222	0954	16FA			JNE	CLKWAIT		00793
1223	0956	04E0	D9C4		CLR	REFRESH	FLAG INDICATING DISPLAY NEEDS REFRESHING	00794
1224	095A	C320	0A14		MOV	CYCF_AG,R12	SHOULD POWER-UP TEST BE CYCLED?	00795
1225	095E	1604			JNE	CYCTEST	YES, INTERLEAVE SCOPE DISPLAY	00796
1226	0960	1F00			TB	0	SHOULD POWER-UP TEST BE CYCLED?	00797
1227					*	JNE	CYCTEST	00798 DEL
1228					*			00005PATCH
1229					*	PROBLEM #4 - PATCH #5 (1 OF 1)		00005PATCH
1230					*			00005PATCH
1231					*	CRJ INPUT TEST FOR POWER-UP TEST CYCLING IS WRONG. THE		00005PATCH
1232					*	POWER-UP TEST SHOULD CYCLE WHEN THE CRU INPUT IS HIGH.		00005PATCH
1233					*			00005PATCH
1234	0962	1302			JEQ	CYCTEST	YES, IF CRU LINE IS HIGH	00005PATCH
1235					*			00005PATCH
1236					*	END OF PROBLEM #4		00005PATCH
1237					*			00005PATCH
1238	0964	0420	0048		BLWP	START		00799
1239	0968	020C	0054		CYCTEST	LI R12,100	SCOPE DISPLAY FOR 100 CLOCKS	00800
1240	096C	04E0	D938		CLR	INTFLAG	CLAR INTERRUPT FLAG	00801
1241	0970	0340			CYCHWAIT	IDLE	WAIT FOR AN INTERRUPT	00802
1242	0972	C2E0	D938		MOV	INTFLAG,R11	GET INTERRUPT FLAG	00803
1243	0976	22E0	337A		COC	CLKINT,R11	WAS IT THE REALTIME CLOCK	00804
1244	097A	16FA			JNE	CYCHWAIT	NO, WAIT FOR IT	00805
1245	097C	0420	141A		BLWP	ROOUT	UPDATE REALTIME SCALE FACTORS	00806
1246	0980	060C			DEC	R12	YES, DECREMENT CLOCK COUNT	00807
1247	0982	15F4			JGT	CYCHWAIT-4		00808
1248	0984	04E0	D908		CLR	BACKUP1	OVERWRITE BATTERY BACKUP FLAG	00809
1249	0988	0460	04EA		B	CYCLEUP	CYCLE POWER-UP TEST	00810
1250	098C		53		VERSION	FCC	'SELF TEST COMPLETE'	00811
	098D		45					
	098E		4C					
	098F		46					
	0990		20					
	0991		54					
	0992		45					
	0993		53					
	0994		54					
	0995		20					
	0996		43					
	0997		4F					
	0998		4D					
	0999		50					
	099A		4C					
	099B		45					
	099C		54					

099D

45

CYCLE KEY

00812

```

1252 *****
1253 **
1254 ** CYCLE POWER UP TEST **
1255 ** **
1256 ** 'F' 'SCOPE' WITH A FRONT PANEL MODE BUTTON DEPRESSED CYCLES **
1257 ** THE POWER TEST. **
1258 ** **
1259 *****
    
```

1260	099E	C060	E010	KEYCYCLE MOV	FP,R1	READ FRONT PANEL MODE BUTTONS	00813
1261	09A2	0241	7800	ANJI	R1,\$7800	KEEP MODE BUTTON INFO ONLY	00814
1262	09A6	0281	7800	CI	R1,\$7800	IS A MODE BUTTON DEPRESSED?	00815
1263	09AA	1306		JEQ	NOCYC	NO, DON'T CYCLE POWER UP TEST	00816
1264	09AC	04E0	D908	CLR	BACKUP1	OVERWRITE BATTERY BACKUP FLAG	00817
1265	09B0	0720	0A14	SET0	CYCFLAG	SET FLAG INDICATING CYCLE	00818
1266	09B4	0450	04EA	B	CYCLEUP		00819
1267	09B8	0380		NOCYC	RTWP		00820

1270			*					
1271			*	HANDLE VEXP KEY (VERTICAL EXPAND)				
1272			*					
1273			*	LEVEL 1 ROUTINE				
1274			*					
1275			*	NOTE:				
1276			*	TO SATISFY THE 20 DIVISION VERTICAL RANGE RESTRICTION,				
1277			*	THE RESTRICTION PLACED ON VXP IS THE EXPANDED VERTICAL				
1278			*	VALUES MUST BE WITHIN + OR - 10 DIVISIONS OF SCREEN				
1279			*	CENTER. FOR THE SAME REASON, THE VZR MUST NOT EXCEED				
1280			*	THE + OR - 10 DIVISION RESTRICTION.				
1281			*					
1282			*	IF THE OPERATIONAL WAVEFORM IS WAVEFORM 0, THE RESULTING				
1283			*	WAVEFORM VALUES EXCEEDING THE 10 DIVISION RESTRICTION WILL				
1284			*	BE CLIPPED AT THE 10 DIVISION BOUNDARY				
1285			*					
1286			*					
1287			*	REGISTER USAGE --				
1288			*					
1289			*	R0 - WAVEFORM INFORMATION				
1290			*	R1,R2 - VERTICAL EXPANSION RATIO				
1291			*	R3 - COUNTER				
1292			*	R4,R5 - WORKING REGISTERS				
1293			*	R6 - ADDRESS OF VZR FOR WAVEFORM				
1294			*					
1295		09BA	*	KEYVXP EQU *				00823
1296	09BA	06A0		BL POPREG				00824
1297	09BE	166E		JNE NOG00D2				00825
1298	09C0	C020		MOV OPMFMD,RJ				00826
1299	09C4	0204		LI R4,\$7FFF	R4=MIN			00827
1300	09C8	0205		LI R5,\$8000	R5=MAX			00828
1301	09CC	C0E0		MOV RESOLV,R3	R3=COUNTER			00829
1302		09D0	*	VXPDRNG EQU *				00830
1303	09D0	8110		C *R0,R4				00831
1304	09D2	1501		JGT *+4				00832
1305	09D4	C110		MOV *R0,R4	NEW MIN			00833
1306	09D6	8150		C *R0,R5				00834
1307	09D8	1101		JLT *+4				00835
1308	09DA	C150		MOV *R0,R5	NEW MAX			00836
1309	09DC	05C0		INCT R0				00837
1310	09DE	0603		DEC R3				00838
1311	09E0	16F7		JNE VXPDRNG				00839
1312	09E2	04C3		CLR R3				00840
1313		09E4	*	VXPTEST EQU *				00841
1314	09E4	0649		DECT SOFT	PUSH MIN OR MAX			00842
1315	09E6	04D9		CLR *SOFT	ZERO EXPONENT			00843
1316	09E8	0649		DECT SOFT				00844
1317	09EA	C644		MOV R4,*SOFT				00845
1318	09EC	0420	70A0	BLWP FPMYZ	CALCULATE EXPANDED MAX OR MIN			00846
1319	09F0	8819	338C	C *SOFT,VXPMAX	CHECK FOR > 10 DIVS			00847
1320	09F4	158B		JGT VXPBAD1				00848
1321	09F6	8819	33A2	C *SOFT,VXPMIN	CHECK FOR < -10 DIVS			00849
1322	09FA	110B		JLT VXPBAD2				00850
1323	09FC	C1B9		MOV *SOFT+,R6	SAVE EXPANDED MAX OR MIN			00851
1324	09FE	05C9		INCT SOFT				00852
1325	0A00	C0C3		VXPST1 MOV R3,R3	IS THIS FIRST OR SECOND PASS?			00853
1326	0A02	160F		JNE TSTVZR				00854
1327	0A04	C105		MOV R5,R4	GO BACK AND DO IT FOR MAX			00855
1328	0A06	C146		MOV R6,R5	SAVE EXPANDED MIN			00856

1329	0A09	0583		INC	R3		00857
1330	0A0A	10EC		JMP	VXPTST		00858
1331				*VXPBAD1	EQU	*	00859 DEL
1332				*	LI	R4,87FFF	00860 DEL
1333				*	JMP	**6	00861 DEL
1334				*VXPBAD2	LI	R5,88000	00862 DEL
1335				*			00888PATCH
1336				*	PROBLEM #7 - PATCH #8 (1 OF 1)		00008PATCH
1337				*			00008PATCH
1338				*	'VXPD' VERTICAL RANGE RESTRICTIONS ARE NOT DETECTED CORRECTLY		00008PATCH
1339				*			00008PATCH
1340	0A0C	C1A0	338C	VXPBAD1	MOV	VXPMAX,R6	00008PATCH
1341	0A10	10D2			JMP	**6	00008PATCH
1342	0A12	C1A0	33A2	VXPBAD2	MOV	VXPMIN,R6	00008PATCH
1343				*			00008PATCH
1344				*	END OF PROBLEM #7		00008PATCH
1345				*			00008PATCH
1346	0A16	C020	D95A		MOV	OPWFH,R0	00863
1347	0A1A	1640			JNE	NOG0DD2	00864
1348	0A1C	04E0	D94C		CLR	WARNING	00865
1349	0A20	10EF			JMP	VXPTST1	00866
1350			0A22	TSTVZR	EQJ	*	00867
1351	0A22	C106			MOV	R6,R4	00868
1352	0A24	C1A0	D95E		MOV	OPWFH,R6	00869
1353	0A28	A1A0	3348		A	VOFFAB,R6	00870
1354	0A2C	0649			DECT	SOFT	00871
1355	0A2E	04D9			CLR	*SOFT	00872
1356	0A30	0649			DECT	SOFT	00873
1357	0A32	C656			MOV	*R6,*SOFT	00874
1358	0A34	0420	70A0		BLWP	FPMPYZ	00875
1359	0A38	1901			JNO	**4	00876
1360	0A3A	1030			JMP	NOG0DD2	00877
1361	0A3C	A159			A	*SOFT,R5	00878
1362	0A3E	1901			JNO	**4	00879
1363	0A40	102D			JMP	NOG0DD2	00880
1364	0A42	A119			A	*SOFT,R4	00881
1365	0A44	1901			JNO	VZROK1	00882
1366	0A46	102A			JMP	NOG0DD2	00883
1367			0A48	VZROK1	EQU	*	00884
1368	0A48	C5B9			MOV	*SOFT+,*R6	00885
1369	0A4A	05C9			INCT	SOFT	00886
1370	0A4C	C020	D95E		MOV	OPWFH,R0	00887
1371	0A50	A020	3338		A	VEXP,R0	00888
1372	0A54	C0F0			MOV	*R0+,R3	00889
1373	0A56	0649			DECT	SOFT	00890
1374	0A58	C650			MOV	*R0,*SOFT	00891
1375	0A5A	0649			DECT	SOFT	00892
1376	0A5C	C643			MOV	R3,*SOFT	00893
1377	0A5E	0420	7050		BLWP	FPDIV	00894
1378	0A62	0640			DECT	R0	00895
1379	0A64	C439			MOV	*SOFT+,*R0	00896
1380	0A66	0770			ABS	*R0+	00897
1381	0A68	C419			MOV	*SOFT,*R0	00898
1382	0A6A	C0E0	D970		MOV	RESOLV,R3	00899
1383	0A6E	C120	D95C		MOV	OPWFMD,R4	00900
1384	0A72	04D9			CLR	*SOFT	00901
1385	0A74	0649			DECT	SOFT	00902
1386			0A76	VXP4PY	EQU	*	00903
1387	0A76	C654			MOV	*R4,*SOFT	00904

'VXPD' - VERTICAL EXPAND

00822

1388	0A78	0420	70A0	BLWP	FPMPYZ		00905
1389	0A7C	8819	338C	C	*SOFT,VXPMAX	IS VALUE > 10 DIVS?	00906
1390	0A80	1102		JLT	*+6		00907
1391	0A82	3660	338C	MOV	VXPMAX,*SOFT	IF SO SET VALUE TO MAX (10 DIVS)	00908
1392	0A85	8819	33A2	C	*SOFT,VXPMIN	IS VALUE < -10 DIVS?	00909
1393	0A8A	1502		JGT	*+6		00910
1394	0A8C	3650	33A2	MOV	VXPMIN,*SOFT	IF SO, SET VALUE TO MIN (-10 DIVS)	00911
1395	0A90	CD19		MOV	*SOFT,*R4+	SAVE NEW VALUE	00912
1396	0A92	0603		DEC	R3		00913
1397	0A94	16F0		JNE	VXPHY		00914
1398	0A96	0720	0972	SETJ	RDTFLAG	UPDATE CALCULATOR READOUT	00915
1399	0A9A	0380		RTWP			00916
1400	0A9C	04E0	094A	CLR	FATAL	CAUSE ERROR	00917
1401	0AA0	C020	33D4	MOV	WSTK,R0	RESTORE STACK VALUES	00918
1402	0AA4	06A0	6936	BL	PSHSTK		00919
1403	0AA8	0380		RTWP			00920

```

1405 *****
1406 **
1407 ** 'VZR', 'HSCL' & 'VSCL' KEY HANDLER **
1408 **
1409 ** LEVEL 1 ROUTINE **
1410 **
1411 ** NOTE --- **
1412 ** 'VZR' PUSHES VERTICAL OFFSET TO (X) **
1413 ** 'HSCL' PUSHES HORIZONTAL SCALE FACTOR TO (X) **
1414 ** 'VSCL' PUSHES VERTICAL SCALE FACTOR TO (X) **
1415 **
1416 *****
1417 0AAA 0420 67E6 KEYVZR BLWP ZROREF GET VERTICAL OFFSET 00922
1418 0AAE 1002 JMP *+6 00923
1419 0AB0 0420 6844 KEYHSCL BLWP HRZSCL GET HORIZONTAL SCALE FACTOR 00924
1420 0AB4 1002 JMP *+6 00925
1421 0AB6 0420 6814 KEYVSCL BLWP VRTSCL GET VERTICAL SCALE FACTOR 00926
1422 0ABA 04C0 CLR R0 FLAG INDICATING FLOATING POINT NUMBER TO (X) 00927
1423 0ABC 06A0 6962 BL PSHREG PUSH SCALE FACTOR TO (X) 00928
1424 0AC0 E820 3348 0972 SDC CLINE15,ROTFLAG UPDATE LINE #16 OF STORED READOUT 00929
1425 0AC6 0380 RTWP 00930

```

'>HSCL', '>VSCL' - (X) TO HORIZONTAL, VERTICAL SCALE

00931

```

1427 *****
1428 **
1429 ** '>HSCL' & '>VSCL' KEY HANDLER **
1430 **
1431 ** LEVEL 1 ROUTINE **
1432 **
1433 ** NOTE --- **
1434 ** '>HSCL' MOVES (X) TO HORIZONTAL SCALE FACTOR **
1435 ** '>VSCL' MOVES (X) TO VERTICAL SCALE FACTOR **
1436 **
1437 *****
1438 DAC8 C020 D978 KEY2HSCL MOV XYWFM,R0 '>HSCL' IS ERROR IS XY DISPLAY MODE 00932
1439 DACC 1619 JNE ERR2SCL 00933
1440 DACE 0206 6EAB LI R6,NEVHSCL SUBROUTINE FOR NEW HORIZONTAL SCALE 00934
1441 DAD2 1002 JMP **6 00935
1442 DAD4 0206 6EE0 KEY2VSCL LI R6,NEVVSCL SUBROUTINE FOR NEW VERTICAL SCALE 00936
1443 DAD8 06A0 690E BL POPREG POP (X) 00937
1444 DADC 06A0 6962 BL PSHREG LEAVE IT ON THE STACK 00938
1445 DAE0 C000 MOV R0,R0 00939
1446 DAE2 160E JNE ERR2SCL WFM # IN X IS ERROR 00940
1447 DAE4 C041 MOV R1,R1 00941
1448 DAE6 110C JLT ERR2SCL (X) <= 0 IS ERROR 00942
1449 DAE8 130B JEQ ERR2SCL 00943
1450 DAEA 0649 DECT SOFT PUSH NEW SCALE FACTOR ONTO SOFTSTACK 00944
1451 DAE C642 MOV R2,*SOFT 00945
1452 DAE 0649 DECT SOFT 00946
1453 DAF0 C641 MOV R1,*SOFT 00947
1454 DAF2 0649 DECT SOFT 00948
1455 DAF4 C660 D95A MOV OPWFM,*SOFT PUSH OPWFM # ONTO SOFTSTACK 00949
1456 DAF8 0416 BLWP *R6 TRANSFER NEW SCALE FACTOR 00950
1457 DAFA 0720 D972 SETO RDTFLAG UPDATE ENTIRE STORED READOUT 00951
1458 DAFE 0380 RTWP 00952
1459 0B00 04E0 D94A ERR2SCL CLR FATAL SET ERROR FLAG 00953
1460 0B04 0380 RTWP 00954

```



```

1462 *
1463 * HANDLE >VZR KEY
1464 *
1465 * LEVEL 1 ROUTINE
1466 *
1467 * NOTE:
1468 * IF THE OPERATIONAL WAVEFORM IS NOT ZERO, THE TRANSFORMED
1469 * WAVEFORM CANNOT HAVE POINTS OUTSIDE THE 20 DIVISION RANGE.
1470 * WAVEFORM ZERO POINTS EXCEEDING THE 20 DIVISION RANGE WILL BE
1471 * CLIPPED AT THE 20 DIVISION BOUNDARY.
1472 *
1473 * WOFFAB = - ZERO REF / 20
1474 *
1475 * REGISTER USAGE -
1476 * R0 - WAVEFORM DATA ADDRESSES
1477 * R1 - NEW VERTICAL OFFSET (NEGATED)
1478 * R3 - OLD VERTICAL OFFSET (NEGATED)
1479 * R11 - WAVEFORM DATA VALUES
1480 *
1481 0B06 06A0 690E KEY2VZR BL POPREG 00956
1482 0B0A 06A0 6962 BL PSHREG 00957
1483 0B0E C000 MOV R0,R0 X MUST BE A CONSTANT 00958
1484 0B10 1645 JNE BAD6 00959
1485 0B12 0501 NEG R1 OFFSET STORED AS NEGATIVE 00960
1486 0B14 C0C1 MOV R1,R3 00961
1487 0B16 0741 ABS R1 00962
1488 0B18 0649 DECT SOFT 00963
1489 0B1A C642 MOV R2,*SOFT 00964
1490 0B1C 0649 DECT SOFT 00965
1491 0B1E C641 MOV R1,*SOFT 00966
1492 0B20 C060 33F8 MOV FP20M,R1 00967
1493 0B24 C0A0 33FA MOV FP20E,R2 00968
1494 0B28 0420 7524 BLWP FPCMPR MUST BE <=20 00969
1495 0B2C 1137 JLT BAD6 00970
1496 0B2E 1603 JNE *+8 00971
1497 0B30 C0C3 MOV R3,R3 IS IT +20 OR -20? 00972
1498 0B32 1101 JLT *+4 IF +20 INCREMENT SO NO OVERFLOW WILL OCCUR 00973
1499 0B34 0583 INC R3 00974
1500 0B36 C643 MOV R3,*SOFT 00975
1501 0B38 0420 7046 BLWP FPDIVZ CALCULATE NEW WOFFAB 00976
1502 0B3C C079 MOV *SOFT+,R1 00977
1503 0B3E 05C9 INCT SOFT 00978
1504 0B40 C0A0 D95E NDSRA MOV OPWFMD,R2 00979
1505 0B44 A0A0 3348 A WOFFAB,R2 00980
1506 0B48 C0D2 MOV *R2,R3 GET OLD WOFFAB 00981
1507 0B4A C020 D95C MOV OPWFMD,R0 00982
1508 0B4E C0A0 D970 MOV RESOLV,R2 00983
1509 0B52 C1E0 D95A MOV OPWFMD,R7 00984
1510 0B56 130B JEQ ADVZR 00985
1511 0B58 TESTVZR EQU * 00986
1512 0B58 C2F0 MOV *R0+,R11 NOT WFM=0, SO ASSURE NO OVERFLOW FIRST 00987
1513 0B5A A2C3 A R3,R11 SUBTRACT OLD VERTICAL OFFSET 00988
1514 0B5C 52C1 S R1,R11 ADD NEW VERTICAL OFFSET 00989
1515 0B5E 1901 JND *+4 00990
1516 0B60 101D JMP BAD6 00991
1517 0B62 0602 DEC R2 00992
1518 0B64 16F9 JNE TESTVZR 00993
1519 0B66 C0A0 D970 MOV RESOLV,R2 00994
1520 0B6A C020 D95C MOV OPWFMD,R0 00995
    
```

1521			0B6E	ADDVZR	EQU *		00996
1522	0B6E	C2D0			MOV *R0,R11		00997
1523	0B70	A2C3			A R3,R11	SUBTRACT OLD VERTICAL OFFSET	00998
1524	0B72	62C1			S R1,R11	ADD NEW VERTICAL OFFSET	00999
1525	0B74	1908			JNO VZROK		01000
1526	0B76	04E0	D94C		CLR WARNING		01001
1527	0B7A	0208	7FFF		LI R11,\$7FFF		01002
1528	0B7E	C041			MOV R1,R1		01003
1529	0B80	1102			JLT VZROK		01004
1530	0B82	1301			JEQ VZROK		01005
1531	0B84	0588			NEG R11		01006
1532			0B86	VZROK	EQU *		01007
1533	0B86	CC0B			MOV R11,*R0*		01008
1534	0B88	0602			DEC R2		01009
1535	0B8A	16F1			JNE ADDVZR		01010
1536	0B8C	C8A0	D95E		MOV OPHFMH,R2		01011
1537	0B90	A0A0	3348		A VOFFAB,R2		01012
1538	0B94	C481			MOV R1,*R2	STORE NEW OFFSET ONLY AFTER CHANGE CONFIRMED	01013
1539	0B96	0720	D972		SETO RDTFLAG		01014
1540	0B9A	0380			RTMP		01015
1541	0B9C	04E0	D94A	BA05	CLR FATAL		01016
1542	0BA0	0380			RTMP		01017

```

1544 *****
1545 **
1546 ** CRT/WFM DISPLAY KEYS- 'SCOPE', 'CALC', 'BOTH' **
1547 ** 'DSW', 'CLW', 'CLD' **
1548 ** 'VS', 'TIME' **
1549 ** 'DOTS', 'VECT' **
1550 ** LEVEL 1 ROUTINE **
1551 ** **
1552 ** INPUT- SOFT,USER **
1553 ** OUTPUT- NONE **
1554 ** **
1555 ** STACK OPERATIONS- **
1556 ** USERSTK- 'DSW', 'CLW', 'VS' POP 1 **
1557 ** **
1558 ** NOTE --- **
1559 ** MNEMONIC FUNCTION **
1560 ** ----- **
1561 ** 'SCOPE' REALTIME DISPLAY ONLY **
1562 ** 'CALC' CALCULATOR DISPLAY ONLY **
1563 ** 'BOTH' REALTIME & CALCULATOR DISPLAY **
1564 ** 'DSW' DISPLAY WFM (X) **
1565 ** 'CLW' CLEAR WFM (X) FROM DISPLAY **
1566 ** 'CLD' CLEAR DISPLAY EXCEPT CWF **
1567 ** 'VS' DISPLAY X-Y WITH (X) AS X **
1568 ** 'TIME' DISPLAY Y-T **
1569 ** 'DOTS' DOT WAVEFORM/CURSOR DISPLAY **
1570 ** 'VECT' VECTOR WAVEFORM/CURSOR DISPLAY **
1571 ** **
1572 *****
    
```

```

1574 *
1575 * 'SCOPE' KEY HANDLER
1576 *
1577 0BA2 0300 0007 KEYSOPE LIM1 7 MASK OFF ALL SYSTEM INTERRUPTS 01019
1578 0BA6 06A0 12BE BL CLRTXT BLANK CRT DISPLAY 01020
1579 0BAA 04E0 09C6 CLR REALTIME SELECT REALTIME DISPLAY 01021
1580 0BAE E820 336E 09CA SOC MW8KHZ,DSPRO DISPLAY READOUT IN 8KHZ MODE 01022
1581 0BB4 E820 3380 09CA SOC MWDRLT,DSPRO SET REALTIME DISPLAY MODE BIT 01023
1582 0BBA E820 3380 09C8 SOC MWDRLT,DSPRLT 01024
1583 0BC0 C320 33C8 MOV WFMLED,R12 STORED WFM LIGHT ON FRONT PANEL 01025
1584 0BC4 1E00 SBZ 0 TURN IT OFF FOR 'SCOPE' MODE 01026
1585 0BC6 0380 RTWP 01027
    
```

```

1586 *
1587 * 'CALC' KEY HANDLER
1588 *
1589 0BC8 0300 0007 KEYSOPE LIM1 7 MASK OFF SYSTEM INTERRUPTS 01028
1590 0BCC 06A0 12BE BL CLRTXT CLEAR TEXT TO BLANK SCREEN 01029
1591 0BD0 0720 DACC SETO OPROGLN REDISPLAY PL# IF NECESSARY 01030
1592 0BD4 0720 09C6 SETO REALTIME SELECT CALCULATOR DISPLAY 01031
1593 0BD8 0720 0972 SETO RDTFLAG UPDATE ALL 4 LINES OF CALCULATOR READOUT 01032
1594 0BDC 4820 336E 09CA SZC MW8KHZ,DSPRO DISPLAY READOUT IN BURST MODE 01033
1595 0BE2 4820 3380 09CE SZC MWDRLT,DSPWFM 01034
1596 0BE8 4820 3380 09CC SZC MWDRLT,DSPCRS RESET REALTIME DISPLAY MODE BIT 01035
1597 0BEE 4820 3380 09CA SZC MWDRLT,DSPRO 01036
1598 0BF4 4820 3380 09C8 SZC MWDRLT,DSPRLT 01037
1599 0BFA C320 33C8 MOV WFMLED,R12 STORED WFM LIGHT ON FRONT PANEL 01038
1600 0BFE 1D00 SBZ 0 TURN IT ON FOR 'CALC' MODE 01039
1601 0C00 0380 RTWP 01040
1602 *
    
```

1603					* 'BOTH' KEY HANDLER		
1604					*		
1605	0C02	0300	0007	KEYBOTH	LIMI 7	MASK OFF SYSTEM INTERRUPTS	01041
1606	0C06	06A0	12BE		BL CLRTXT	CLEAR TEXT TO BLANK SCREEN	01042
1607	0C0A	0720	0ACC		SETO OPROGLN	REDISPLAY PL# IF NECESSARY	01043
1608	0C0E	0720	09C6		SETO REALTIME	SELECT CALCULATOR DISPLAY	01044
1609	0C12	0720	0972		SETO RDTFLAG	UPDATE ALL 4 LINES OF CALCULATOR READOUT	01045
1610	0C16	4820	336E	D9CA	SZC MW0KHZ, DSPRO	DISPLAY READOUT IN BURST MODE	01046
1611	0C1C	4820	3380	D9CA	SZC MWDRLT, DSPRO	NO REALTIME DURING READOUT OR	01047
1612	0C22	4820	3380	D9CC	SZC MWDRLT, DSPCRS	CURSOR DISPLAY	01048
1613	0C28	E820	3380	D9CE	SOC MWDRLT, DSPWFM	SET REALTIME DISPLAY MODE BIT	01049
1614	0C2E	E820	3380	D9C8	SOC MWDRLT, DSPRLT		01050
1615	0C34	C320	33C8		MOV WFMLED, R12	STORED WFM LIGHT ON FRONT PANEL	01051
1616	0C38	1000			SBJ 0	TURN IT ON FOR 'BOTH' MODE	01052
1617	0C3A	0380			RTWP		01053
1618					*		
1619					* 'JSW' KEY HANDLER		
1620					*		
1621	0C3C	06A0	690E	KEYJSW	BL POPREG	POP X OFF USER STACK	01054
1622	0C40	162A			JNE DSWERR	X MUST BE A FLOATING POINT NUMBER	01055
1623	0C42	06A0	75A0		BL FP2INT	CONVERT X TO INTEGER	01056
1624	0C46	1127			JLT DSWERR	X MUST BE IN RANGE 0 <= X <= WFMAXN	01057
1625	0C48	8801	0974		C R1, WFMAXN		01058
1626	0C4C	1524			JGT DSWERR		01059
1627	0C4E	C101			MOV R1, R4		01060
1628	0C50	C050	0974		MOV WFMAXN, R1	MAXIMUM NUMBER OF WAVEFORMS ALLOWED	01061
1629	0C54	04C2			CLR R2	SET DISPLAYED WAVEFORM COUNT TO 0	01062
1630	0C56	06A0	6984	KNTJSWS	BL ADRWFM	GET HEADER AND DATA ADDRESSES FOR NEXT WFM	01063
1631	0C5A	C0F9			MOV *SOFT+, R3	HEADER ADDRESS	01064
1632	0C5C	05C9			INCT SOFT	DATA ADDRESS	01065
1633	0C5E	A0E0	3354		A DISPLA, R3	POINT TO DISPLAY FLAG IN HEADER	01066
1634	0C62	C0D3			MOV *R3, R3	GET DISPLAY FLAG	01067
1635	0C64	20E0	333A		CJC C1, R3	IS WAVEFORM BEING DISPLAYED VIA 'JSW'?	01068
1636	0C68	1601			JNE *+4	NO, DON'T COUNT IT	01069
1637	0C6A	0582			INC R2	YES, COUNT IT	01070
1638	0C6C	0601			DEC R1	CHECK ALL WFMS	01071
1639	0C6E	15F3			JGT KNTDSWS	WFMS ARE NUMBERED FROM 0 - WFMAXN	01072
1640	0C70	13F2			JEQ KNTDSWS		01073
1641	0C72	8802	3348		C R2, DSPMAX	CAN MORE WFMS BE ADDED TO DISPLAY?	01074
1642	0C76	140A			JHE DSWHARN	NO, THIS IS AN WARNING	01075
1643	0C78	C044			MOV R4, R1		01076
1644	0C7A	06A0	6984		BL ADRWFM	GET HEADER AND DATA ADDRESSES FOR WFM X	01077
1645	0C7E	C0B9			MOV *SOFT+, R2	HEADER ADDRESS	01078
1646	0C80	05C9			INCT SOFT	DATA ADDRESS	01079
1647	0C82	A0A0	3354		A DISPLA, R2	POINT TO DISPLAY FLAG IN HEADER	01080
1648	0C86	E4A0	333A		SOC C1, *R2	SET 'DSW' DISPLAY FLAG FOR WFM X	01081
1649	0C8A	1008			JMP DSWR3		01082
1650	0C8C	06A0	6918	DSWHARN	BL POPSTK	POP X OFF USER STACK	01083
1651	0C90	04E0	094C		CLR WARNING	SET WARNING FLAG TO TELL USER	01084
1652	0C94	1005			JMP DSWR3		01085
1653	0C96	C020	33D4	DSWERR	MOV WSTK, R0	ON ERROR, PUSH X BACK ONTO USER STACK	01086
1654	0C9A	06A0	6935		BL PSHSTK		01087
1655	0C9E	04E0	094A		CLR FATAL	SET ERROR FLAG TO TELL USER	01088
1656	0CA2	0720	0972	DSWR0	SETO RDTFLAG	UPDATE CALCULATOR READOUT	01089
1657	0CA6	0380			RTWP		01090
1658					*		
1659					* 'CLW' KEY HANDLER		
1660					*		
1661	0CA8	06A0	690E	KEYCLW	BL POPREG	POP X OFF USER STACK	01091

1662	QCAC	161B		JNE	CLWERR	X MUST BE A FLOATING POINT NUMBER	01092	
1663	QCAE	06A0	75A0	BL	FP2INT	CONVERT X TO INTEGER	01093	
1664	QCB2	1118		JLT	CLWERR	X MUST BE IN RANGE 0<=X<=WFMAXN	01094	
1665	QCB4	8801	D974	C	R1,WFMAXN		01095	
1666	QCB8	1515		JGT	CLWERR		01096	
1667	QCB8	06A0	6984	BL	ADRFWM	GET HEADER AND DATA ADDRESSES FOR WFM X	01097	
1668	QCB8	C0B9		MOV	*SOFT+,R2	HEADER ADDRESS	01098	
1669	QCC0	05C9		INCT	SOFT	DATA ADDRESS	01099	
1670	QCC2	A0A0	3354	A	DISPLA,R2	POINT TO DISPLAY FLAG IN HEADER	01100	
1671	QCC6	C0D2		MOV	*R2,R3	GET DISPLAY FLAG	01101	
1672	QCC8	20E0	333A	CO	C1,R3	IS THIS WFM BEING DISPLAYED VIA 'DSW'?	01102	
1673	QCC8	1608		JNE	CLWRRN	NO, THIS IS A WARNING	01103	
1674	QCCE	04D2		CLR	*R2	YES, RESET 'DSW' DISPLAY FLAG FOR WFM X	01104	
1675	QCD0	C0A0	D95E	MOV	OPWFMM,R2		01105	
1676	QCD4	A0A0	3354	A	DISPLA,R2	SET 'CFW' DISPLAY FLAG	01106	
1677	QCD8	E4A0	333C	SOC	C2,*R2		01107	
1678	QCD8	1009		JMP	CLWRD		01108	
1679	QCDE	04E0	D94C	CLWRRN	CLR	WARNING SET WARNING FLAG TO TELL USER	01109	
1680	QCE2	1006		JMP	CLWRD		01110	
1681	QCE4	C020	33D4	CLWERR	MOV	HSTK,R0 ON ERROR, PUSH X BACK ONTO USER STACK	01111	
1682	QCE8	06A0	6935	BL	PSHSTK		01112	
1683	QCEC	04E0	D94A	CLR	FATAL	SET ERROR FLAG TO TELL USER	01113	
1684	QCF0	E820	333C	D972	CLWRD	SOC	CLINE2,ROTFLAG UPDATE READOUT LINE #2	01114
1685	QCF6	E820	3348	D972		SOC	CLINE16,ROTFLAG UPDATE READOUT LINE #16	01115
1686	QCF8	0380			RTWP		01116	
1687				*				
1588				*	'CLD' KEY HANDLER			
1689				*				
1690	QCFE	C060	D974	KEYCLD	MOV	WFMAXN,R1 GET MAXIMUM ALLOWED NUMBER OF WFMS	01117	
1691	QD02	06A0	6984		3L	ADRFWM GET HEADER AND DATA ADDRESSES FOR NEXT WFM	01118	
1692	QD06	C0B9			MOV	*SOFT+,R2 HEADER ADDRESS	01119	
1693	QD08	05C9			INCT	SOFT DATA ADDRESS	01120	
1694	QD0A	A0A0	3354		A	DISPLA,R2 POINT TO DISPLAY FLAG FOR NEXT WFM	01121	
1695	QD0E	04D2			CLR	*R2 RESET 'DSW' DISPLAY FLAG FOR THIS WFM	01122	
1696	QD10	0601			DEC	R1 RESET ALL 'DSW' DISPLAY FLAGS	01123	
1697	QD12	15F7			JGT	KEYCLD+4 WAVEFORMS ARE NUMBERED FROM 0 - WFMAXN	01124	
1698	QD14	13F6			JEQ	KEYCLD+4	01125	
1699	QD16	C0A0	D95E		MOV	OPWFMM,R2 SET 'OPW' DISPLAY FLAG	01126	
1700	QD1A	A0A0	3354		A	DISPLA,R2	01127	
1701	QD1E	C4A0	333C		MOV	C2,*R2	01128	
1702	QD22	E820	333C	D972		SOC	CLINE2,ROTFLAG UPDATE READOUT LINE #2	01129
1703	QD28	0380			RTWP		01130	
1704				*				
1705				*	'VS' KEY HANDLER			
1706				*				
1707	QD2A	06A0	698E	KEYVS	BL	POPREG POP X ELEMENT OFF USER STACK	01131	
1708	QD2E	1610			JNE	VSERR X MUST BE A FLOATING POINT NUMBER	01132	
1709	QD30	06A0	75A0		BL	FP2INT CONVERT X TO INTEGER	01133	
1710	QD34	1100			JLT	VSERR X MUST BE IN RANGE 0<=X<=WFMAXN	01134	
1711	QD36	8801	D974		C	R1,WFMAXN	01135	
1712	QD3A	150A			JGT	VSERR	01136	
1713	QD3C	06A0	6984		BL	ADRFWM GET HEADER AND DATA ADDRESSES FOR WFM X	01137	
1714	QD40	05C9			INCT	SOFT HEADER ADDRESS	01138	
1715	QD42	A079			A	*SOFT+,R1 ADD DATA ADDRESS TO VS WFM #	01139	
1716	QD44	C801	D978		MOV	R1,XYWFM SAVE AS VS WFM	01140	
1717	QD48	E820	3362	D9CE		SOC	MWXVSY,DSPFWM SET X-Y DISPLAY FLAG FOR WAVEFORMS	01141
1718	QD4E	1006			JMP	VSRO	01142	
1719	QD50	C020	33D4	VSERR	MOV	HSTK,R0 ON ERROR, PUSH X BACK ONTO USER STACK	01143	
1720	QD54	06A0	6936		BL	PSHSTK	01144	

1721	0058	04E0	D94A		CLR	FATAL	SET ERROR FLAG TO TELL USER	01145
1722	005C	0720	D972	VSRJ	SET0	ROTFLAG	UPDATE ALL 4 LINES OF READOUT	01146
1723	0060	0380			RTWP			01147
1724				*				
1725				*		'TIME' KEY HANDLER		
1726				*				
1727	0062	04E0	D978		KEYTIME CLR	XVWFM	SET VS WFM # TO ZERO VALUE	01148
1728	0066	4820	3362	D9CE	SZC	MMXVSY,DSPWFM	SET Y-T DISPLAY FLAG FOR WAVEFORMS	01149
1729	006C	0720	D972		SET0	ROTFLAG	UPDATE ALL 4 LINES OF READOUT	01150
1730	0070	0380			RTWP			01151
1731				*				
1732				*		'DOTS' KEY HANDLER		
1733				*				
1734	0072	4820	3348	D9CE	KEYDOTS SZC	MMVCTR,DSPWFM	SELECT DOT CURSORS/WAVEFORMS	01152
1735	0078	4820	3348	D9CC		SZC MMVCTR,DSPGRS		01153
1736	007E	0380				RTWP		01154
1737				*				
1738				*		'VECT' KEY HANDLER		
1739				*				
1740	0080	E820	3348	D9CE	KEYVECT SOC	MMVCTR,DSPWFM	SELECT VECTOR CURSORS/WAVEFORMS	01155
1741	0086	E820	3348	D9CC		SOC MMVCTR,DSPGRS		01156
1742	008C	0380				RTWP		01157

1744										
1745									**	**
1746					**	KEY HANDLER FOR FRONT PANEL 3JTTONS			**	**
1747					**				**	**
1748					**	LEVEL 1 ROUTINE			**	**
1749					**				**	**
1750					**	INPUT: NONE			**	**
1751					**	OUTPJT: FRONT PANEL STATUS SETUP			**	**
1752					**	DESTROYS: R1-R2			**	**
1753					**				**	**
1754					**	NOTE ---			**	**
1755					**	THIS ROUTINE CONTROLS THE FRONT PANEL LIGHTS			**	**
1756					**	AND MODE SWITCH FUNCTIONS. BITS 15-9 GO TO			**	**
1757					**	THE MODE SWITCHES AND BITS 8-0 GO TO THE			**	**
1758					**	LIGHTS (3JTTONS).			**	**
1759					**				**	**
1760										
1761										
1762	008E	C820	3350	DA26	KEYVMDL	MOV	VERTL,VERTM	SAVE VERTICAL MODE SWITCH INFORMATION		01159
1763	0094	0201	FF00			LI	R1,\$FF00	BITS TO SET REALTIME MODE SWITCH AND		01160
1764	0098	1017				JMP	KEYVMD	VERTICAL LEFT FRONT PANEL BUTTON		01161
1765	009A	C820	334C	DA26	KEYVMDALT	MOV	VERTALT,VERTM	SAVE VERTICAL MODE SWITCH INFORMATION		01162
1766	00A3	0201	FE83			LI	R1,\$FE80	BITS TO SET REALTIME MODE SWITCH AND		01163
1767	00A4	1011				JMP	KEYVMD	VERTICAL ALTERNATE FRONT PANEL BUTTON		01164
1768	00A5	C820	334E	DA26	KEYVMDADD	MOV	VERTADD,VERTM	SAVE VERTICAL MODE SWITCH INFORMATION		01165
1769	00AC	0201	FE40			LI	R1,\$FE40	BITS TO SET REALTIME MODE SWITCH AND		01166
1770	00B0	1008				JMP	KEYVMD	VERTICAL ADD FRONT PANEL BUTTON		01167
1771	00B2	C820	3346	DA26	KEYVMDCHOP	MOV	VERTCHP,VERTM	SAVE VERTICAL MODE SWITCH INFORMATION		01168
1772	00B8	0201	FE20			LI	R1,\$FE20	BITS TO SET REALTIME MODE SWITCH AND		01169
1773	00BC	1005				JMP	KEYVMD	VERTICAL CHOP FRONT PANEL BUTTON		01170
1774	00BE	C820	333C	DA26	KEYVMDR	MOV	VERTR,VERTM	SAVE VERTICAL MODE SWITCH INFORMATION		01171
1775	00C4	0201	FE10			LI	R1,\$FE10	BITS TO SET REALTIME MODE SWITCH AND		01172
1776	00C8	0202	01F0		KEYVMD	LI	R2,\$01F0	VERTICAL RIGHT FRONT PANEL BUTTON		01173
1777	00CC	1019				JMP	FPBTN	LOAD VERTICAL MASK IN R2		01174
1778										
1779	00CE	C820	333E	DA28	KEYHMDA	MOV	HORZA,HORZM	SAVE HORIZONTAL MODE SWITCH INFORMATION		01175
1780	00D4	0201	FE08			LI	R1,\$FE08	BITS TO SET REALTIME MODE SWITCH AND		01176
1781	00D8	1011				JMP	KEYHMD	HORIZONTAL A FRONT PANEL BUTTON		01177
1782	00DA	C820	3344	DA28	KEYHMDALT	MOV	HORZALT,HORZM	SAVE HORIZONTAL MODE SWITCH INFORMATION		01178
1783	00E0	0201	FE04			LI	R1,\$FE04	BITS TO SET REALTIME MODE SWITCH AND		01179
1784	00E4	1008				JMP	KEYHMD	HORIZONTAL ALT FRONT PANEL BUTTON		01180
1785	00E6	C820	3354	DA28	KEYHMDCHOP	MOV	HORZCHP,HORZM	SAVE HORIZONTAL MODE SWITCH INFORMATION		01181
1786	00EC	0201	FE02			LI	R1,\$FE02	BITS TO SET REALTIME MODE SWITCH AND		01182
1787	00F0	1005				JMP	KEYHMD	HORIZONTAL CHOP FRONT PANEL BUTTON		01183
1788	00F2	C820	3352	DA28	KEYHMDB	MOV	HORZB,HORZM	SAVE HORIZONTAL MODE SWITCH INFORMATION		01184
1789	00F8	0201	FE01			LI	R1,\$FE01	BITS TO SET REALTIME MODE SWITCH AND		01185
1790	00FC	0202	000F		KEYHMD	LI	R2,\$000F	HORIZONTAL B FRONT PANEL BUTTON		01186
1791										
1792	0E00	4802	09C2		FPBTN	SZC	R2,FPTEMP	SET GROUP (VERT/HORZ) LIGHT BITS TO 0		01187
1793	0E04	E801	09C2			SZC	R1,FPTEMP	SET MODE SWITCH BITS AND LIGHT BIT TO 1		01188
1794	0E08	C820	09C2	E010		MOV	FPTEMP,FP	PASS STATUS WORD TO HARDWARE		01189
1795	0E0E	0202	0002			LI	R2,2			01190
1796	0E12	04E0	0938		FPWAIT	CLR	INTFLAG	CLEAR INTERRUPT FLAGS		01191
1797	0E16	C050	0938			MOV	INTFLAG,R1	READ INTERRUPT FLAGS		01192
1798	0E1A	2060	337A			CDC	CLKINT,R1	WAIT FOR TWO 20 MILLISECOND CLOCKS		01193
1799	0E1E	16FB				JNE	*-8			01194
1800	0E20	0602				DEC	R2			01195
1801	0E22	15F7				JST	FPWAIT			01196
1802	0E24	0380				RTWP				01197


```

1804 *****
1805 **
1806 ** HANDLER FOR 20 MILLI-SECOND CLOCK INTERRUPT **
1807 ** **
1808 ** INTERRUPT HANDLER **
1809 ** **
1810 ** NOTE --- **
1811 ** **
1812 ** 1) EVERY 20 MILLISECONDS FRONT PANEL BUTTONS **
1813 ** ARE PULSED TO DETECT IF ONE OF THEM IS BEING **
1814 ** PUSHED. A VALID KEY CLOSURE REQUIRES THAT **
1815 ** BUTTON IS DOWN FOR 2 CONSECUTIVE CLOCK **
1816 ** INTERRUPTS. KEY IS THEN EXECUTED AFTER **
1817 ** SECOND DETECTION. ALL OTHER CONSECUTIVE **
1818 ** DETECTIONS OF THIS BUTTON ARE IGNORED. **
1819 ** 2) EVERY 20 MILLISECONDS CRT DISPLAY IS **
1820 ** REFRESHED TO PREVENT FLICKER. IF REFRESH **
1821 ** TAKES LESS THAN 20 MILLISECONDS DISPLAY **
1822 ** ROUTINE IS INACTIVATED UNTIL NEXT 20 **
1823 ** MILLISECOND INTERRUPT. THIS ALLOWS MAXIMUM **
1824 ** PROCESSOR TIME FOR OTHER FUNCTIONS. IF **
1825 ** REFRESH TAKES MORE THAN 20 MILLISECONDS IT IS **
1826 ** NOT STOPPED BUT TOLD TO CONTINUE AS FLICKER **
1827 ** IS STARTING. **
1828 ** 3) EVERY 100 MILLISECONDS REALTIME READOUT **
1829 ** IS CHECKED IF IT NEEDS UPDATING. **

```

```

1830 **
1831 *****
1832
1833 ***** CHECK GPIB ADDRESS SWITCH FOR UPDATES
1834

```

Address	OpCode	Op1	Op2	Op3	Op4	Comment	Label
1835	DE26	0300	0000			CLOCK	01199
1836	DE2A	C060	D906			MOV INT7854,R1	01200
1837	DE2E	1302				JEQ *+6	01201
1838	DE30	0420	F034			BLWP \$F034	01202
1839	DE34	0300	0008			LIMI 8	01203
1840	DE38	C060	D942			MOV GPIBOPT,R1	01204
1841	DE3C	1174				JLT FPKEY	01205
1842	DE3E	9820	E050	D91A		CB R4R,GPIBADR	01206
1843	DE44	1359				JEQ LEDCHK	01207
1844	DE46	D820	3390	E06C		MOVB RESET,R3W	01208
1845	DE4C	D820	338E	E070		MOV3 CH7FFF,R4W	01209
1846	DE52	C320	33C6			MOV IOLED,R12	01210
1847	DE56	1E00				SBZ 0	01211
1848	DE58	C320	33CA			MOV SRQLED,R12	01212
1849	DE5C	1E00				SBZ 0	01213
1850	DE5E	C320	33CC			MOV REMLED,R12	01214
1851	DE62	1E00				SBZ 0	01215
1852	DE64	D060	E050			MOV3 R4R,R1	01216
1853	DE68	C801	D91A			MOV R1,GPIBADR	01217
1854	DE6C	0541				INV R1	01218
1855	DE6E	C081				MOV R1,R2	01219
1856	DE70	0720	D922			SET0 ONOFF	01220
1857	DE74	2060	3390			CCC ONLINE,R1	01221
1858	DE78	1639				JNE OFFLINE	01222
1859	DE7A	0241	1F00			ANDI R1,\$1F00	01223
1860	DE7E	0281	1F00			CI R1,\$1F00	01224
1861	DE82	1334				JEQ OFFLINE	01225
1862	DE84	04E0	D922			CLR ONOFF	01226

1863	0E88	C0E0	3338		MOV	ADDRMODE,R3	GET BASIC ADDRESS MODE INFO	01227
1864	0E9C	04E0	D926		CLR	TOTLLO	FLAG INDICATING TALK-LISTEN MODE	01228
1865	0E90	0720	D924		SETJ	TRMYPE	FLAG INDICATING <EOI> OR <LF> TERMINATOR	01229
1866	0E94	0242	6000		ANDI	R2,\$6000	KEEP MODE SELECT BITS ONLY	01230
1867	0E98	80A0	3386		C	TALKER,R2	IS THE 7854 SELECTED TO TALK-ONLY?	01231
1868	0E9C	1607			JNE	CHKLD	NO, THEN CHECK FOR LISTEN-ONLY	01232
1869	0E9E	E060	3386		SOC	DAL,R1	YES, DISABLE THE LISTENER	01233
1870	0EA2	E0E0	3386		SOC	TO,R3	SET TO TALK-ONLY MODE	01234
1871	0EA6	0720	D926		SETJ	TOTLLO	FLAG INDICATING TALK-ONLY MODE	01235
1872	0EAA	1010			JMP	LOTO		01236
1873	0EAC	80A0	338A	CHKLD	C	LISTENER,R2	IS THE 7854 SELECTED TO LISTEN-ONLY?	01237
1874	0EB0	1608			JNE	CHKEJIT	NO, THEN 7854 IS TALKER-LISTENER	01238
1875	0EB2	E060	3382		SOC	DAT,R1	YES, DISABLE THE TALKER	01239
1876	0EB6	E0E0	3382		SOC	LO,R3	SET TO LISTEN-ONLY MODE	01240
1877	0EBA	C820	333A	D926	MOV	C1,TOTLLO	FLAG INDICATING LISTEN-ONLY MODE	01241
1878	0EC0	1005			JMP	LOTO		01242
1879	0EC2	80A0	3382	CHKEJIT	C	EOILF,R2	CHECK IF <EOI> OR <LF>	01243
1880	0EC6	1302			JEQ	LOTO		01244
1881	0EC8	04E0	D924		CLR	TRMYPE	FLAG TO <EOI> ONLY	01245
1882	0ECC	D801	E070	LOTO	MOVW	R1,R4W	TELL CHIP 7854'S ADDRESS & MODE	01246
1883	0ED0	04E0	E06C		CLR	R3W	RESET SOFTWARE GPIB CHIP RESET	01247
1884	0ED4	D820	3300	E060	MOVW	INTMASK,R0W	SET GPIB INTERRUPT MASK	01248
1885	0EDA	D803	E068		MOVW	R3,R2W	SET ADDRESS MODE	01249
1886	0EDE	D803	D994		MOVW	R3,AMREG	SAVE THIS DATA	01250
1887	0EE2	20E0	3386		CDC	TO,R3		01251
1888	0EE6	1602			JNE	OFFLINE	IS 7854 IN 'TALK-ONLY' MODE?	01252
1889	0EE8	D020	E040		MOVW	R0R,R0	READ INTERRUPT STATUS REGISTER TO CLEAR BO	01253
1890	0EEC	04E0	D93E	OFFLINE	CLR	IGNORE	DON'T IGNORE GPIB	01254
1891	0EF0	04E0	D918		CLR	CHDINDX	SET CHDBUF INDEX TO BEGINING	01255
1892	0EF4	04E0	D940		CLR	INOUT	FLAG INDICATING GPIB IDLE	01256
1893								
1894							***** TURN ON/OFF I/O, SRQ & REMOTE LED'S *****	
1895								
1896		0EF8		LEDCHK	EQU	*		01257
1897	0EF8	C2E0	D922		MOV	ONOFF,R11	DON'T CHANGE LEDES IF OFFLINE	01258
1898	0EFC	1114			JLT	FPKEY		01259
1899	0EFE	C320	33C6		MOV	IOLEJ,R12	CRU LINE FOR I/O LED	01260
1900	0F02	D2E0	E048		MOVW	R2R,R11	READ GPIB'S ADDRESS STATUS REGISTER	01261
1901	0F06	22E0	3390		CDC	MA,R11	HAS MY ADDRESS OCCURRED?	01262
1902	0F0A	1302			JEQ	*+6		01263
1903	0F0C	1E00			SBZ	0	NO, TJRN I/O LED OFF	01264
1904	0F0E	1001			JMP	*+4		01265
1905	0F10	1D00			SBZ	0	YES, TJRN I/O LED ON	01266
1906	0F12	C320	33CC		MOV	REMLED,R12	CRU LINE FOR REM LED	01267
1907	0F16	D2E0	E044		MOVW	R1R,R11	READ GPIB'S COMMAND STATUS REGISTER	01268
1908	0F1A	22E0	338A		CDC	RLOK,R11	IS 7854 IN RMLS STATE?	01269
1909	0F1E	1302			JEQ	*+6		01270
1910	0F20	1E00			SBZ	0	NO, TJRN REM LED OFF	01271
1911	0F22	1001			JMP	*+4		01272
1912	0F24	1D00			SBZ	0	YES, TJRN REM LED ON	01273
1913								
1914							***** CHECK IF A FRONT PANEL BUTTON IS BEING PUSHED *****	
1915								
1916	0F26	C060	E010	FPKEY	MOV	FP,R1	GET FRONT PANEL STATUS WORD	01274
1917	0F2A	09A1			SRL	R1,10	BUTTON DOWN CODE IS IN BITS 11-14	01275
1918	0F2C	0241	081E		ANDI	R1,\$1E		01276
1919	0F30	C061	378A		MOV	MODESW(R1),R1	GET CORRESPONDING KEYCODE	01277
1920	0F34	1603			JNE	FPKEYON		01278
1921	0F36	04C5			CLR	R5	IF KEYCODE=0 THEN NO KEY IS DEPRESSED	01279

20 MILLI-SECOND CLOCK INTERRUPT HANDLER

01198

1922	0F38	04C6			CLR	R6	SET LAST KEY CODE TO 'NONE'	01280
1923	0F3A	101A			JMP	REFRSH	NEXT, CHECK IF DISPLAY NEEDS REFRESHING	01281
1924	0F3C	8181			FPKEYDN C	R1,R6	CHECK IF PRESENT KEYCODE IS SAME AS	01282
1925	0F3E	1302			JEQ	SAMEKEY	LAST KEYCODE	01283
1926	0F40	04C5			CLR	R5	IF NOT, SET COUNT BACK TO 0	01284
1927	0F42	C181			MOV	R1,R6	SAVE NEW KEYCODE	01285
1928	0F44	0585			SAMEKEY INC	R5	INCREMENT DETECTION COUNT OF THIS KEY DOWN	01286
1929	0F46	0285	0002		CI	R5,2	EXECUTE BUTTON IF DOWN, BUT ONLY ON	01287
1930	0F4A	1612			JNE	REFRSH	SECOND CONSECUTIVE DETECTION OF THAT BUTTON	01288
1931	0F4C	C801	D962		MOV	R1,FKEY	MOVE KEYCODE TO 'FKEY'	01289
1932	0F50	C050	D942		MOV	GPIB OPT,R1	IS GPIB OPTION INSTALLED?	01290
1933	0F54	110B			JLT	LCLFP	NO, EXECUTE LOCAL FRONT PANEL KEY	01291
1934	0F56	D820	337C	E05C	MOV8	RTL,R3W	SEND LOCAL RTL MESSAGE TO INTERFACE	01292
1935	0F5C	C041			MOV	R1,R1	DUMMY INSTRUCTION FOR MORE MC68583 CLOCKS	01293
1936	0F5E	D060	E044		MOV8	R1R,R1	READ COMMAND STATUS REGISTER	01294
1937	0F62	2060	3386		CDC	REM,R1	IS GPIB IN REMOTE	01295
1938	0F66	04E0	E06C		CLR	R3W	REMOVE RTL MESSAGE	01296
1939	0F6A	1302			JEQ	REFRSH	IF REMOTE, IGNORE FRONT PANEL KEYS	01297
1940	0F6C	0420	0040		LCLFP BLW	FPKY	EXECUTE KEY	01298
1941								
1942							***** RESTART DISPLAY EVERY 20 MILLI-SECONDS *****	
1943								
1944	0F70	0720	0A16		REFRSH SET0	FLICKER	FLAG INDICATING FLICKER IF DISPLAY ISN'T	01299
1945	0F74	C020	D9C4		MOV	REFRSH,R0	RESTARTED. CHECK IF DISPLAY IS STILL BEING	01300
1946	0F78	1107			JLT	UPDTR0	REFRESHED. IF SO, FLICKER WILL START	01301
1947	0F7A	0420	0FA4		BLW	DSPINT	IF DISPLAY IS OFF, START IT	01302
1948	0F7E	4820	3382	D938	SZC	DSPINT,INTFLAG	RESET DISPLAY INTERRUPT FLAG	01303
1949	0F84	04E0	0A16		CLR	FLICKER	FLAG INDICATING NO FLICKER	01304
1950								
1951							***** UPDATE READOUT ON MULTIPLES OF THIS INTERRUPT *****	
1952								
1953	0F88	0604			UPDTR0 DEC	R4	DECREMENT COUNTER OF INTERRUPTS TIL. UPDATE	01305
1954	0F8A	1504			JGT	CLKFLG		01306
1955	0F8C	0720	0A18		SET0	ROUPDT		01307
1956	0F90	0204	0006		LI	R4,6	SETUP NEW DELAY COUNT	01308
1957	0F94	E820	337A	D938	CLKFLG SOC	CLKINT,INTFLAG	FLAG INDICATING CLOCK INTERRUPT	01309
1958	0F9A	C320	338C		MOV	CLKRST,R12	RESET REALTIME CLOCK INTERRUPT	01310
1959	0F9E	1E00			SBZ	0		01311
1960	0FA0	1D00			SBO	0		01312
1961	0FA2	0380			RTW			01313

1963					*****		
1964					**		**
1965					** HANDLER FOR 7854 DISPLAY		**
1966					**		**
1967					** INTERRUPT HANDLER		**
1968					**		**
1969					** INPUT- R5 FROM LAST DISPLAY INTERRUPT		**
1970					** OUTPUT- DISPLAY CONTROL WORDS & NEW R5		**
1971					** DESTROYS- R0-R12		**
1972					**		**
1973					** STACK OPERATIONS-		**
1974					** NONE		**
1975					**		**
1976					** NOTE ---		**
1977					** R5 SHOULD BE INITIALIZED TO -2 TO		**
1978					** START/RESTART 7854 DISPLAY		**
1979					**		**
1980					*****		
1981	0FA4		DC20		DSPINIT WORD	WPDSP	DISPLAY WORKSPACE 01315
1982	0FA6		0FA8		WORD	*+2	01316
1983	0FA8	0300	0000		LIMI	0	DISABLE SYSTEM INTERRUPTS 01317
1984	0FAC	0720	09C4		SETJ	REFRESH	FLAG INDICATING REFRESH CYCLE STARTING 01318
1985	0FB0	0720	0A12		SETJ	CYCLE	FLAG INDICATING TO CONTINUE DISPLAYING 01319
1986	0FB4	0205	FFFE		LI	R5,-2	INITIALIZE DISPLAY SEQUENCE INDICATOR 01320
1987	0FB8	0450	0FCA		B	INTDSP	01321
1988	0FBC	0300	0000		DISP LAY	LIMI 0	MASK OFF ALL INTERRUPTS 01322
1989	0FC0	C1E0	D906		MOV	INT7854,R7	SHOULD THIS INTERRUPT BE HANDLED BY 01323
1990	0FC4	1302			JEQ	INTDSP	A DIAGNOSTIC ROUTINE? 01324
1991	0FC6	0420	F038		BLWP	\$F038	YES, TRANSFER CONTROL TO IT 01325
1992	0FCA	0300	0007		INTDSP	LIMI 7	MASK OFF ONLY SYSTEM INTERRUPTS 01326
1993	0FCE	C820	09C8	E000	MOV	DSPRLT,0MDWRD	ALLOW REALTIME DISPLAY DURING SETUP 01327
1994	0FD4	C1E0	D900		MOV	ACQWFM,R7	ACQUIRE MODE WORD 01328
1995	0FD8	21E0	3382		COC	MWACQR,R7	IS A WAVEFORM ACQUIRE IN PROGRESS? 01329
1996	0FDC	1603			JNE	NO.ACQ	NO, CONTINUE REGULARLY 01330
1997	0FDE	21E0	337E		COC	MW2WFM,R7	YES, ARE 2 WAVEFORMS BEING ACQUIRED? 01331
1998	0FE2	1313			JEQ	RSTINT	YES, DISPLAY HAS TO BE OFF 01332
1999	0FE4	05C5			NO.ACQ	INCT R5	IS THIS THE FIRST WAVEFORM TO DISPLAY? 01333
2000	0FE6	151C			JGT	WAVEFORM	NO, CONTINUE THROUGH DISPLAY CYCLE 01334
2001	0FE8	C0A0	DA12		MOV	CYCLE,R2	IS THIS THE START OR END OF THE DISPLAY CYCLE? 01335
2002	0FEC	1119			JLT	WAVEFORM	IF START, DISPLAY FIRST WAVEFORM 01336
2003	0FEE	C020	D9CE		MOV	DSPWFM,R0	IF END, CHECK IF INTERLACING IS NEEDED 01337
2004	0FF2	2020	3380		COC	MWDRLT,R0	ARE REALTIME WAVEFORMS ON? 01338
2005	0FF6	1605			JNE	NO.INTR	NO, DO NOT INTERLACE DISPLAY 01339
2006	0FF8	0246	0002		ANDI	R6,\$0002	YES, TOGGLE INTERLACE ADDRESS OFFSET 01340
2007	0FFC	29A0	333C		XOR	C2,R6	TOGGLE INTERLACE ADDRESS +0,+2,+0,+2,.... 01341
2008	1000	1001			JMP	*+4	01342
2009	1002	04C6			NO.INTR	CLR R6	ADDRESS OFFSET OF +0 FOR NO INTERLACE 01343
2010	1004	C0A0	DA16		MOV	FLICKER,R2	HAS 20 MILLISECOND CLOCK ALREADY INTERRUPTED? 01344
2011	1008	1109			JLT	NXTCYCL	YES, START NEXT CYCLE AS FLICKER IS STARTING 01345
2012	100A	C320	3332		RSTINT	MOV DSPRST,R12	01346
2013	100E	1E00			SBZ	0	NO, RESET DISPLAY DONE INTERRUPT 01347
2014	1010	E820	3382	D938	SDC	DSPINT,INTFLAG	SET DISPLAY DONE INTERRUPT FLAG 01348
2015	1016	04E0	D9C4		CLR	REFRESH	FLAG DISPLAY CYCLE AS COMPLETED 01349
2016	101A	0380			RTWP		01350
2017	101C	04E0	DA16		NXTCYCL	CLR FLICKER	CLEAR FLICKER FLAG 01351
2018	1020	C060	D9C6		WAVEFORM	MOV REALTIME,R1	CHECK IF ONLY READOUT SHOULD 01352
2019	1024	1102			JLT	*+6	BE DISPLAYED 01353
2020	1026	0450	1138		B	READOUT	01354
2021	102A	C065	D97E		MOV	DSPBUF(R5),R1	GET NEXT WFM # 01355

DISPLAY INTERRUPT HANDLER

01314

2022	102E	0241	FF00		ANJI	R1,\$FF00	ADDRESS IS IN HIGH BYTE	01356
2023	1032	1330			JEQ	CURSORS	WFM ADDRESS = 0 ==> CURSOR(S) NEXT	01357
2024	1034	C020	D9CE		MOV	DSPWFM,R0	WAVEFORM DISPLAY MODE WORD	01358
2025	1038	C0A0	D978		MOV	XYWFM,R2	ADDRESS OF VS WAVEFORM	01359
2026	103C	0242	FF00		ANDI	R2,\$FF00	ADDRESS IS IN HIGH BYTE	01360
2027	104J	1602			JNE	*+6	IF Y-T DISPLAY USE YTFWM	01361
2028	1042	D282	D97A		LI	R2,YTFWM	ADDRESS OF Y-T ADXD PRELOADS	01362
2029	1046	A046			A	R6,R1	ADD INTERLACE ADDRESS OFFSET	01363
2030	1048	A086			A	R6,R2		01364
2031	104A	C111			MOV	*R1,R4	VALUE OF FIRST Y POINT	01365
2032	104C	C002			MOV	*R2,R3	VALUE OF FIRST X POINT	01366
2033	104E	2020	3348		COC	MWVCTR,R0	IS DISPLAY IN VECTORS?	01367
2034	1052	1602			JNE	STRTDSP	NO, VALUES ARE OK	01368
2035	1054	05C1			INCT	R1	YES, AWRD MUST GET NEXT POINT	01369
2036	1056	85C2			INCT	R2	3WRD MUST GET NEXT POINT	01370
2037	1058	C320	3382		STRTDSP MOV	DSPRST,R12		01371
2038	105C	1E00			SBZ	0	RESET DISPLAY DONE INTERRUPT	01372
2039	105E	C800	E000		MOV	R0,D4DWRD	SETUP MODE WORD	01373
2040	1062	C801	E002		MOV	R1,AWRD	SETUP PRIMARY WFM/READOUT ADDRESS	01374
2041	1066	21E0	3382		COC	MWACQR,R7	DON'T CHANGE B WORD DURING A WAVEFORM	01375
2042	106A	1302			JEQ	*+6	ACQUISITION	01376
2043	106C	C802	E004		MOV	R2,BWRD	SETUP SECONDARY WFM ADDRESS	01377
2044	1070	C803	E012		MOV	R3,ADXD	SETUP X PRELOAD REGISTER	01378
2045	1074	C804	E014		MOV	R4,ADYD	SETUP Y PRELOAD REGISTER	01379
2046	1078	C320	3336		MOV	DSPSTP,R12		01380
2047	107C	1D08			SBD	0	SET DISLAY STOP CRU BIT	01381
2048	107E	C320	3382		MOV	DSPRST,R12		01382
2049	1082	1D00			SBD	0	FINISH RESETTING DISLAY DONE INTERRUPT	01383
2050	1084	C320	3334		MOV	DSPSTR,R12		01384
2051	1088	1E00			SBZ	0	PULSE DISPLAY START CRU BIT	01385
2052	108A	1D00			SBD	0		01386
2053	108C	E820	3382	D938	SOC	DSPINT,INTFLAG	SET DISPLAY DONE INTERRUPT FLAG	01387
2054	1092	0380			RTWP			01388
2055								
2056	1094	C2A0	D954		CJRSORS MOV	CURSJR,R10	CHECK TO SEE IF CURSORS ARE ON	01389
2057	1098	134F			JEQ	READOUT	CURSOR <= 0 --> NO CURSORS READOUT NEXT	01390
2058	109A	114E			JLT	READOUT		01391
2059	109C	04C7			CLR	R7	DEFAULT TO DOT CURSORS	01392
2060	109E	C020	D9CC		MOV	DSPCRS,R0	GET DISPLAY CURSOR MODE BITS	01393
2061				*	CZC	MWVCTR,R0	SHOULD VECTORS CURSORS BE DISPLAYED?	
2062				*	JEQ	*+6	NO, STAY WITH DOT CURSORS	
2063				*	LI	R7,\$0100	YES, LOAD HEIGHT SIZE	
2064	10A2	0A1A			SLA	R10,1		01394
2065	10A4	C22A	D954		NEXTORS MOV	CURSJR(R10),R8	GET NEXT CURSOR ELEMENT #	01395
2066	10A8	0A18			SLA	R8,1	ADDRESS = (ELEMENT # * 2) + BASE ADDRESS OF	01396
2067	10AA	A220	D960		A	CRSOPWD,R8		01397
2068	10AE	0204	D900		LI	R4,VCRS	ADDRESS OF VERTICAL CURSOR POINTS	01398
2069	10B2	C084			MOV	R4,R2		01399
2070	10B4	C518			MOV	*R8,*R4	FIRST CURSOR POINT	01400
2071	10B6	6D07			S	R7,*R4+	ADJUST FOR VECTOR CURSOR IF NECESSARY	01401
2072	10B8	1902			JND	*+6		01402
2073	10BA	0644			DECT	R4	CHECK FOR OVERFLOW	01403
2074	10BC	A007			A	R7,*R4+	UNDO SUBTRACT IF OVERFLOW	01404
2075	10BE	C518			MOV	*R8,*R4	SECOND CURSOR POINT	01405
2076	10C0	A507			A	R7,*R4	ADJUST FOR VECTOR CURSOR IF NECESSARY	01406
2077	10C2	1901			JND	*+4	CHECK FOR OVERFLOW	01407
2078	10C4	6507			S	R7,*R4	UNDO ADDITION IF OVERFLOW	01408
2079	10C6	C22A	D954		MOV	CURSJR(R10),R8	GET CURSOR ELEMENT #	01409
2080	10CA	C260	D978		MOV	XYWFM,R9	ARE X-Y OR Y-T CURSORS TO BE USED?	01410

2081	10CE	1306			JEQ	YTCRS	Y-T, THEN X MUST BE CALCULATED	01411	
2082	10D0	0249	FF00		ANDI	R9,\$FF00	X-Y, BASE ADDRESS IS IN HIGH BYTE	01412	
2083	10D4	0A18			SLA	R8,1	ADDRESS = (ELEMENT # * 2) + BASE ADDRESS OF	01413	
2084	10D6	A209			A	R9,R8		01414	
2085	10D8	C218			MOV	*R8,R8	X VALUE IS THIS WFM POINT	01415	
2086	10DA	1007			JMP	HCRSR		01416	
2087	10DC	3A20	337C		MPY	C1024,R8	ELEMENT # * 1024	01417	
2088	10E0	3E20	D970		DIV	RESOLV,R8	(ELEMENT # * 1024) / RESOLUTION	01418	
2089	10E4	0A48			SLA	R8,4	((ELEMENT # * 1024) / RESOLUTION) * 16	01419	
2090	10E6	0228	E000		AI	R8,\$E000	((ELEMENT # * 1024) / RESOLUTION) * 15) + \$E0	01420	
2091	10EA	0203	DA00		LI	R3,HCRS	FIRST CURSOR POINT	01421	
2092	10EE	C043			MOV	R3,R1		01422	
2093	10F0	CC08			MOV	R8,*R3+	HORIZONTAL POINTS ARE THE SAME	01423	
2094	10F2	C4C8			MOV	R8,*R3	SECOND CURSOR POINT	01424	
2095	10F4	C320	3382		MOV	DSPRST,R12	RESET DISPLAY DONE INTERRUPT	01425	
2096	10F8	1E00			SBZ	0		01426	
2097	10FA	C800	E000		MOV	R0,DMDWRD	SETUP MODE WORD	01427	
2098	10FE	C802	E002		MOV	R2,AWRD	SETUP PRIMARY CURSOR ADDRESS	01428	
2099	1102	C801	E004		MOV	R1,BWRD	SETUP SECONDARY CURSOR ADDRESS	01429	
2100	1106	C813	E012		MOV	*R3,ADX0	SETUP X PRELOAD REGISTER	01430	
2101	110A	C814	E014		MOV	*R4,ADY0	SETUP Y PRELOAD REGISTER	01431	
2102	110E	C320	3386		MOV	DSPSTP,R12	SET DISPLAY STOP CRU BIT	01432	
2103	1112	1D00			SBJ	0		01433	
2104	1114	C320	3382		MOV	DSPRST,R12		01434	
2105	1118	1D00			SBJ	0		01435	
2106	111A	C320	3384		MOV	DSPSTR,R12		01436	
2107	111E	1E00			SBZ	0	PULSE DISPLAY START CRU BIT	01437	
2108	1120	1D00			SBJ	0		01438	
2109	1122	C320	D970		MOV	RESOLV,R12	CURSORS SHOULD BE BRIGHTER FOR	01439	
2110	1126	093C			SRL	R12,3	LARGER RESOLUTIONS	01440	
2111	1128	060C			DEC	R12	DELAY	01441	
2112	112A	15FE			JGT	*-2		01442	
2113	112C	C320	3386		MOV	DSPSTP,R12		01443	
2114	1130	1E00			SBZ	0	PULSE DISPLAY STOP CRU BIT	01444	
2115	1132	1D00			SBJ	0		01445	
2116	1134	064A			DECF	R10	CHECK FOR MORE CURSORS TO DISPLAY	01446	
2117	1136	1586			JGT	NEXTCRS		01447	
2118									
2119	1138	C020	D9CA		READJUT	MOV	DSPR0,R0	GET DISPLAY READOUT MODE WORD	01448
2120	113C	C050	DADE		MOV	DISHM,R1	A WORD GETS READOUT DISPLAY DATA ADDRESS	01449	
2121	1140	04C2			CLR	R2	CLEAR B WORD	01450	
2122	1142	0203	E002		LI	R3,\$E002	X PRELOAD FOR CHARACTER DISPLAY	01451	
2123	1146	0204	1FF0		LI	R4,\$1FF0	Y PRELOAD FOR CHARACTER DISPLAY	01452	
2124	114A	0205	FFFE		LI	R5,-2	RESET DISPLAY INDICATOR	01453	
2125	114E	04E8	DA12		CLR	CYCLE	FLAG INDICATING LAST CYCLE OF REFRESH	01454	
2126	1152	0460	1058		B	STRTDSP		01455	

TEKCODE TO ASCII READOUT CONVERSION

01456

```

2128 *****
2129 **
2130 ** CONVERT TEKCODE READOUT TO ASCII READOUT **
2131 ** ** **
2132 ** LEVEL & ROUTINE **
2133 ** ** **
2134 ** INPUT- TEKCODE BUFFER ADDRESS **
2135 ** OUTPUT- ASCII READOUT BUFFER SETUP **
2136 ** DESTROYS- R0-R12 **
2137 ** ** **
2138 ** STACK OPERATIONS- **
2139 ** NONE **
2140 ** ** **
2141 ** NOTE --- **
2142 ** THIS ROUTINE USES TABLE 'TEKASC' TO **
2143 ** CONVERT FROM TEKCODE TO ASCII. TABLE **
2144 ** ENTRIES ARE ONE BYTE IN LENGTH AND ARE **
2145 ** NEGATIVE IF THEY ARE CONTROL FUNCTIONS. **
2146 ** ** **
2147 ** TABLE INDEX = TEKCODE - 96 **
2148 ** ** **
2149 *****
2150 1156 DBC0 RLTRD WORD WPLVL4 LEVEL & WORKSPACE 01457
2151 1158 115A WORD *+2 01458
2152 115A C0E0 D9C6 MOV REALTIME,R3 IS REALTIME DISPLAY ON? 01459
2153 115E 1303 JEQ RLTRDQ YES, UPDATE READOUT 01460
2154 1160 0380 RTWP NO, DON'T OVERWRITE CALCULATOR DISPLAY 01461
2155
2156 1162 DBC0 RDA2R WORD WPLVL4 LEVEL & WORKSPACE 01462
2157 1164 1165 WORD *+2 01463
2158 1166 0203 36EC RLTRD LI R3,ASCII INITIALIZE ADDRESS TABLE INDEX 01464
2159 116A 04C0 CLR R0 DECODING WORD 0 01465
2160 116C C26D 0012 MOV 18(R13),SOFT SOFTWARE STACK POINTER 01466
2161 1170 0229 FFFC AI SOFT,-4 01467
2162 1174 05C9 CNVWRD INCT SOFT REINITIALIZE TEXT VARIABLES 01468
2163 1176 C673 MOV *R3+,*SOFT NEXT LINE # 01469
2164 1178 0649 DECT SOFT 01470
2165 117A C673 MOV *R3+,*SOFT NEXT CHARACTER # 01471
2166 117C 0649 DECT SOFT 01472
2167 117E 0208 DA8A LI R8,CNVBUF BUFFER USED FOR CONVERSION 01473
2168 1182 C548 MOV R8,*SOFT TEXT POINTER 01474
2169 1184 0649 DECT SOFT 01475
2170 1186 020C 000A LI R12,10 01476
2171 118A C040 MOV R0,R1 01477
2172 118C 384C MPY R12,R1 NEXT WORD ADDRESS = 10 * (2*WORD #) + TEKCODE 01478
2173 118E 0222 DF00 AI R2,TEKCODE 01479
2174 1192 C64C MOV R12,*SOFT 10 CHARACTERS 01480
2175 1194 DE20 33AC MOV# NULL,*R8+ FILL ASCII WORD WITH NULLS 01481
2176 1198 060C DEC R12 01482
2177 119A 15FC JGT *-6 01483
2178 119C 0208 DA8A LI R8,CNVBUF RESET POINTER TO START OF ASCII WORD 01484
2179 11A0 C312 MOV *R2,R12 GET TIMESLOT 1 CHARACTER 01485
2180 11A2 024C 00F0 ANJI R12,$F0 KEEP ROW NUMBER ONLY 01486
2181 11A6 028C 00C0 CI R12,$C0 CHECK FOR STANDARD FORMAT (TS1=ROW 3) 01487
2182 11AA 162B JNE FREE 01488
2183 11AC 04C6 TS1 CLR R6 SET NULL/SKIP FLAG TO NULL 01489
2184 11AE 04C4 CLR R4 NO DECIMAL POINT USED IN STANDARD FORMAT 01490
2185 11B0 0707 SETO R7 ENSURE DECIMAL POINT FLAGS ARE NOT EQUAL 01491
2186 11B2 06A0 1288 BL TIMESLOT GET TIMESLOT CHARACTER IN R12 01492

```


2187	1186	024C	000F		ANDI	R12,\$F	KEEP COLUMN NUMBER ONLY	01493
2188	118A	028C	000E		CI	R12,\$E	CHECK FOR 'IDENTIFY' COMMAND (ROW 3 COL 10)	01494
2189	118E	1353			JEQ	IDENTFY		01495
2190	11C0	022C	0050		AI	R12,\$60	CREATE 'TEKASC' TABLE INDEX	01496
2191	11C4	D12C	3710		MOV3	TEKASC(R12),R4		01497
2192	11C8	06C4			SWP3	R4		01498
2193	11CA	C144			MOV	R4,R5		01499
2194	11CC	0245	0003		ANDI	R5,\$0003	KEEP ADD ZEROS INFORMATION	01500
2195	11D0	0924			SRL	R4,2		01501
2196	11D2	0244	0001		ANDI	R4,\$0001	KEEP SHIFT PREFIX INFORMATION	01502
2197	11D6	06A0	1288	TS23	BL	TIMESLOT	CONVERT TIMESLOT 2	01503
2198	11DA	06A0	1288		BL	TIMESLOT	CONVERT TIMESLOT 3	01504
2199	11DE	0706		TS4	SET3	R6	SET NULL/SKIP FLAG TO SKIP	01505
2200	11E0	06A0	1288		BL	TIMESLOT	CONVERT TIMESLOT 4	01506
2201	11E4	0605		TS557	DEC	R5		01507
2202	11E6	1103			JLT	TS8	USE TIMESLOTS 5-7 TO ADD ZEROS FROM TIMESLOT	01508
2203	11E8	DE20	3797		MOV3	ASCII0,*R8+	1 CONTROL COMMAND	01509
2204	11EC	10FB			JMP	TS567		01510
2205	11EE	0222	0005	TS8	AI	R2,6	ADJUST TEKCODE POINTER TO TIMESLOT 8	01511
2206	11F2	A484			A	R4,*R2	SHIFT PREFIX AS REQUIRED	01512
2207	11F4	06A0	1288		BL	TIMESLOT	CONVERT TIMESLOT 8	01513
2208	11F8	06A0	1288	TS910	BL	TIMESLOT	CONVERT TIMESLOT 9	01514
2209	11FC	06A0	1288		BL	TIMESLOT	CONVERT TIMESLOT 10	01515
2210	1200	103A			JMP	NXTWRD		01516
2211								
2212	1202	04C5		FREE	CLR	R5	SET DECIMAL POINT POSITION TO 0	01517
2213	1204	C042			MOV	R2,R1		01518
2214	1206	0207	000A		LI	R7,10	CHECK 10 TIMESLOTS FOR DECIMAL POINT	01519
2215	120A	C331		CKDECP	MOV	*R1+,R12	MOVE TEKCODE CHARACTER TO R12	01520
2216	120C	024C	00FF		ANDI	R12,\$FF		01521
2217	1210	022C	FFA0		AI	R12,-96	CREATE TEK TO ASCII TABLE INDEX	01522
2218	1214	D32C	3710		MOV3	TEKASC(R12),R12	MOVE ASCII CHARACTER TO HIGH BYTE OF R12	01523
2219	1218	150F			JGT	NODECP		01524
2220	121A	C10C			MOV	R12,R4		01525
2221	121C	024C	F000		ANDI	R12,\$F000	KEEP HIGH BYTE OF ASCII CODE	01526
2222	1220	028C	D000		CI	R12,\$D000	CHECK IF DECIMAL POINT LOCATION COMMAND	01527
2223	1224	1609			JNE	NODECP		01528
2224	1226	06C4			SWP3	R4		01529
2225	1228	0244	000F		ANDI	R4,\$F	KEEP DECIMAL POINT LOCATION	01530
2226	122C	C144			MOV	R4,R5		01531
2227	122E	0585			INC	R5		01532
2228	1230	0507			NEG	R7	COMMAND LOCATION IS (11-N)	01533
2229	1232	0227	000B		AI	R7,11		01534
2230	1236	1002			JMP	ISDECP		01535
2231	1238	0607		NODECP	DEC	R7		01536
2232	123A	15E7			JGT	CKDECP		01537
2233	123C	04C6		ISDECP	CLR	R6	SET NULL/SKIP FLAG TO NULL	01538
2234	123E	0204	0001		LI	R4,1	START CONVERTING AT TIMESLOT 1	01539
2235	1242	06A0	1288	TS1.3	BL	TIMESLOT	CONVERT NEXT TIMESLOT	01540
2236	1246	0584			INC	R4	INCREMENT TIMESLOT NUMBER	01541
2237	1248	0284	0004		CI	R4,4	CHECK IS FIRST 3 ARE CONVERTED	01542
2238	124C	11FA			JLT	TS1.3		01543
2239	124E	0706			SET3	R6	SET NULL/SKIP FLAG TO SKIP	01544
2240	1250	0144		TS4.10	C	R4,R5	CHECK IF DECIMAL POINT SHOULD BE ADDED	01545
2241	1252	1602			JNE	*+6		01546
2242	1254	DE20	372F		MOV3	DECPNT,*R8+	MOVE DECIMAL POINT TO NEXT LOCATION	01547
2243	1258	06A0	1288		BL	TIMESLOT	CONVERT TIMESLOTS 4 TO 10	01548
2244	125C	0584			INC	R4	INCREMENT TIMESLOT COUNTER	01549
2245	125E	0284	0003		CI	R4,11	CHECK IF ALL 10 TIMESLOTS WERE CONVERTED	01550

2246	1262	11F6		JLT	TS4.10		01551
2247	1264	1008		JMP	NXTWRD		01552
2248							
2249	1266	020C	000A	IDENTIFY	LI	R12,10	01553
2250	126A	0608		DEC	R8	SET CHARACTER COUNTER TO START OF WORD	01554
2251	126C	0206	37B0	LI	R6,IDENTIFY	MOVE ' IDENTIFY ' TO ASCII BUFFER	01555
2252	1270	DE36		MOVB	*R6+,*R8+		01556
2253	1272	060C		DEC	R12		01557
2254	1274	15FD		JGT	*-4		01558
2255							
2256	1276	06A0	136C	NXTWRD	BL	TEXT	01559
2257	127A	05C0		INCT	R0	MOVE ASCII STRING TO READOUT BUFFER	01560
2258	127C	0283	370C	CI	R3,ENDASC	POINT TO NEXT WORD TO DECODE	01561
2259	1280	1402		JHE	*+6		01562
2260	1282	0460	1174	B	CONVWRD	CONVERT ALL 8 WORDS	01563
2261	1286	0380		RTWP		RETURN TO CALLING ROUTINE	01564
2262							
2263	1288	C332		TIMESLOT	MOV	*R2+,*R12	01565
2264	128A	024C	00FF	ANDI	R12,\$FF	MOVE TEKCODE CHARACTER TO R12	01566
2265	128E	022C	FFA0	AI	R12,-96	CREATE TEK TO ASCII TABLE INDEX	01567
2266	1292	032C	3710	MOV3	TEKASC(R12),R12	MOVE ASCII CHARACTER TO HIGH BYTE OF R12	01568
2267	1296	1509		JGT	NOTCTRL		01569
2268	1298	980C	37A0	CB	R12,JUMP	IS THIS A JUMP COMMAND?	01570
2269	129C	13EC		JEQ	NXTWRD	YES, SKIP TO NEXT WORD	01571
2270	129E	81C4		C	R4,R7	CHECK IF LOCATION OF DECIMAL POINT COMMAND?	01572
2271	12A0	1305		JEQ	NOTCTRL+2	YES, THEN IT IS A SKIP	01573
2272	12A2	C186		MOV	R6,R6	IF CTRL COMMAND MOVE NULL/SKIP TO BUFFER	01574
2273	12A4	1103		JLT	NOTCTRL+2		01575
2274	12A6	0588		INC	R8	IF NULL, ADVANCE CHARACTER COUNTER	01576
2275	12A8	1001		JMP	NOTCTRL+2		01577
2276	12AA	DE0C		NOTCTRL	MOVB	R12,*R8+	01578
2277	12AC	022C	0060	AI	R12,96	IF REGULAR CHARACTER MOVE IT TO ASCII BUFFER	01579
2278	12B0	045B		B	*R11	RESET TEKCODE CHARACTER	01580


```

2280 *****
2281 **
2282 ** INITIALIZE TEXT BUFFER **
2283 ** ** **
2284 ** LEVEL 5 ROUTINE **
2285 ** ** **
2286 ** INPUT- SOFT **
2287 ** OUTPUT- SOFT **
2288 ** DESTROYS- R7,R8 **
2289 ** ** **
2290 ** STACK OPERATIONS- **
2291 ** SOFTSTACK - INITXT PUSHES 2 **
2292 ** ** **
2293 ** NOTE --- **
2294 ** THIS ROUTINE INITIALIZES THE TEXT BUFFER TO **
2295 ** DISPLAY A BLANK SCREEN. IT ALSO PUTS A TEXT **
2296 ** POINTER ON THE SOFTSTACK POINTING TO THE HOME **
2297 ** POSITION (LINE #1, CHARACTER #1). **
2298 ** ** **
2299 *****
2300 1282 0649 INITXT DECT SOFT PUT TEXT POINTER OF SOFTSTACK 01582
2301 1284 C660 333A MOV C1,*SOFT 01583
2302 1288 0649 DECT SOFT SET TEXT POINTER TO HOME POSITION (UPPER-LEFT) 01584
2303 128A C660 333A MOV C1,*SOFT 01585
2304 128E C1E0 DADE CLRXT MOV DISMEM,R7 ADDRESS OF DISPLAY MEMORY 01586
2305 12C2 DDE0 3380 MOV3 RS,*R7+ FIRST CHARACTER IS <RESET> (HOME) 01587
2306 12C6 0208 D9D4 LI R8,LINESTR TABLE OF LINE STARTING ADDRESSES 01588
2307 12CA CED7 INITCLR MOV R7,*R8+ INITIALIZE TABLE OF LINE START ADDRESSES 01589
2308 12CC DDE0 3381 MOV3 CRLF,*R7+ INITIALIZE DISPLAY MEMORY 01590
2309 12D0 D5E0 33AD MOV3 ETX,*R7 <END-OF-TEXT> 01591
2310 12D4 0288 D9F4 CI R8,ENDTEXT 01592
2311 12D8 1AF8 JL INITCLR 01593
2312 12DA C607 MOV R7,*R8 01594
2313 12DC 045B B *R11 RETURN TO CALLING ROUTINE 01595
2314 *****
2315 ** ** **
2316 ** ADVANCE TEXT POINTER **
2317 ** ** **
2318 ** LEVEL 5 ROUTINES **
2319 ** ** **
2320 ** INPUT- TEXT POINTER ON SOFTSTACK **
2321 ** OUTPJT- TEXT POINTER UPDATED **
2322 ** DESTROYS- R12 **
2323 ** ** **
2324 ** NOTE --- **
2325 ** TEXT POINTER IS TWO WORDS- **
2326 ** 1) CHARACTER # **
2327 ** 2) LINE # **
2328 ** ** **
2329 *****
2330 12DE CE60 333A ADVLIN MOV C1,*SOFT+ SET CHARACTER NUMBER TO 1 01596
2331 12E2 0599 INC *SOFT INCREMENT LINE NUMBER 01597
2332 12E4 04CC CLR R12 01598
2333 12E6 8819 3358 C *SOFT,C16 CHECK IF BEYOND RANGE 01599
2334 12EA 1204 JLE LINEOK 01600
2335 12EC C660 3358 MOV C16,*SOFT SET LINE NUMBER TO 16 01601
2336 12FD 020C 4000 LI R12,$4000 01602
2337 12F4 0649 LINEOK DECT SOFT REPOINT TEXT POINTER TO CHARACTER NUMBER 01603
2338 12F6 A30C A R12,R12 SET/RESET OVERFLOW STATUS BIT 01604
    
```

01581

2339	12F8	045B		3	*R11		01605	
2340	12FA	C1F9		NULLIV	MOV	*SOFT+,R7	CHARACTER NUMBER TO START BLANKING	01606
2341	12FC	0607			DEC	R7		01607
2342	12FE	C219			MOV	*SOFT,R8	LINE #	01608
2343	1300	0649			DECT	SOFT		01609
2344	1302	0608			DEC	R8		01610
2345	1304	0A18			SLA	R8,1	GET STARTING ADDRESS OF CURRENT LINE	01611
2346	1306	0228	D9D4		AI	R8,LINSTR		01612
2347	130A	A1F8			A	*R8+,R7	START OF CURRENT LINE	01613
2348	130C	C218			MOV	*R8,R8		01614
2349	130E	0608			DEC	R8	END OF CURRENT LINE	01615
2350	1310	8207		NULLFIL	C	R7,R8	ARE WE DONE FILLING?	01616
2351	1312	1A01			JL	*+4	NO, KEEP FILLING.	01617
2352	1314	045B			B	*R11	YES, RETURN	01618
2353	1316	DDE0	33AC		MOV8	NULL,*R7+	FILL WITH <NULL>S	01619
2354	131A	10FA			JMP	NULLFIL		01620
2355	131C	C1F9		OLDTXT	MOV	*SOFT+,R7	GET CURRENT CHARACTER #	01621
2356	131E	0607			DEC	R7		01622
2357	1320	C219			MOV	*SOFT,R8	GET CURRENT LINE #	01623
2358	1322	0608			DEC	R8		01624
2359	1324	0A18			SLA	R8,1	INDEX INTO LINE START ADDRESS TABLES	01625
2360	1326	A1E8	D9D4		A	LINSTR(R8),R7	ACTUAL ADDRESS OF TEXT POINTER	01626
2361	132A	0229	FFFC		AI	SOFT,-4		01627
2362	132E	C647			MOV	R7,*SOFT	PUSH ADDRESS ONTO SOFTSTACK	01628
2363	1330	045B			B	*R11		01629
2364	1332	CE60	333A	DELIN	MOV	C1,*SOFT+	SET CHARACTER NUMBER TO 1	01630
2365	1336	C1D9			MOV	*SOFT,R7	GET CURRENT LINE NUMBER	01631
2366	1338	0649			DECT	SOFT	REPOINT SOFTSTACK TO CHARACTER NUMBER	01632
2367	133A	0607			DEC	R7		01633
2368	133C	0A17			SLA	R7,1	CALCULATE INDEX INTO LINE START TABLE	01634
2369	133E	0227	D9D4		AI	R7,LINSTR		01635
2370	1342	C237			MOV	*R7+,R8	STARTING ADDRESS OF LINE # N	01636
2371	1344	C317			MOV	*R7,R12	STARTING ADDRESS OF LINE # N+1	01637
2372	1346	DE20	33B1		MOV8	CRLF,*R8+	DELETE CURRENT LINE	01638
2373	134A	DE3C		DCMPRSS	MOV8	*R12+,*R8+	COMPRESS TEXT TO DELETE TEXT	01639
2374	134C	8320	D9F4		C	ENDTEXT,R12		01640
2375	1350	1BFC			JH	DCMPRSS		01641
2376	1352	D620	33A3		MOV8	ETX,*R8	GUARANTEE THAT <ETX> IS AT END OF TEXT	01642
2377	1356	53D8			S	R8,R12	COUNT OF CHARACTERS DELETED	01643
2378	1358	5DCC		DADJST	S	R12,*R7+	ADJUST TABLE OF LINE START ADDRESSES	01644
2379	135A	0287	D9F4		CI	R7,ENDTEXT		01645
2380	135E	12FC			JLE	DADJST		01646
2381	1360	045B			B	*R11		01647

2383				*****			
2384				**			**
2385				**	ADD TEXT TO CRT DISPLAY		**
2386				**			**
2387				**	LEVEL 5 ROUTINE		**
2388				**			**
2389				**	INPJT - TEXTS - BYTE COUNT FIRST POP		**
2390				**	DATA NEXT N POPS (N FROM FIRST POP)		**
2391				**	CHARACTER NUMBER NEXT POP		**
2392				**	LINE NUMBER NEXT POP		**
2393				**			**
2394				**	TEXT - BYTE COUNT FIRST POP		**
2395				**	ADDRESS OF DATA SECOND POP		**
2396				**	CHARACTER NUMBER THIRD POP		**
2397				**	LINE NUMBER FOURTH POP		**
2398				**			**
2399				**	OUTPUT - TEXT PUT IN CRT DISPLAY BUFFER		**
2400				**			**
2401				**	STACK OPERATIONS -		**
2402				**	SOFTSTACK - BYTE COUNT & DATA POPPED		**
2403				**	TEXT POINTER UPDATED		**
2404				**			**
2405				*****			
2406	1362	C239		TEXTS	MOV *SOFT+,R8	POP CHARACTER COUNT	01649
2407	1364	C309			MOV SOFT,R12	POINTER TO INPUT STRING	01650
2408	1366	A248			A R8,SOFT		01651
2409	1368	A248			A R8,SOFT	ADJUST SOFTSTACK TO CHARACTER POSITION #	01652
2410	136A	1002			JMP SVREG		01653
2411	136C	C239		TEXT	MOV *SOFT+,R8	POP CHARACTER COUNT	01654
2412	136E	C339			MOV *SOFT+,R12	POP POINTER TO INPUT STRING	01655
2413	1370	C805	DA24	SVREG	MOV R5,TEXT3	SAVE REGISTERS R5,R6	01656
2414	1374	C806	DA20		MOV R6,TEXT1		01657
2415	1378	C179			MOV *SOFT+,R5	POP CHARACTER POSITION #	01658
2416	137A	C199			MOV *SOFT,R6	POP LINE #	01659
2417	137C	C1C8			MOV R8,R7		01660
2418	137E	A1C5			A R5,R7	DETERMINE LINE LENGTH AFTER STRING IS INPUT	01661
2419	1380	0607			DEC R7		01662
2420	1382	81E0	342A		C LINELEN,R7	CHECK IF LINE WILL BE TOO LONG	01663
2421	1386	1409			JHE LENOK		01664
2422	1388	C168	DA24		MOV TEXT3,R5	RESTORE R5 & R6	01665
2423	138C	C1A0	DA20		MOV TEXT1,R6		01666
2424	1390	0649			DECT SOFT	RESET SOFTSTACK TO TEXT POINTER	01667
2425	1392	0208	8000		LI R8,\$8000		01668
2426	1396	A208			A R8,R8	SET CARRY & OVERFLOW STATUS BITS	01669
2427	1398	0458			B *R11		01670
2428	139A	0606		LENOK	DEC R6		01671
2429	139C	0A16			SLA R6,1	POINTER TO ADDRESS OF START OF LINE # N	01672
2430	139E	0226	09D4		AI R6,LINSTR		01673
2431	13A2	A156			A *R6,R5	STARTING ADDRESS OF LINE # N	01674
2432	13A4	0605			DEC R5		01675
2433	13A6	A1F6			A *R6+,R7	ENDING ADDRESS OF INPUT STRING	01676
2434	13A8	0587			INC R7		01677
2435	13AA	8587			C R7,*R6	WILL INPUT STRING FIT?	01678
2436	13AC	121F			JLE TEXTOK	YES, INSERT NEW STRING	01679
2437	13AE	C803	DA1E		MOV R3,TEXT0	SAVE R3	01680
2438	13B2	6106			S *R6,R7	NEEDED EXPANSION ROOM	01681
2439	13B4	C0C7			MOV R7,R3	SAVE EXPANSION COUNT	01682
2440	13B6	C160	09F4		MOV ENDTEXT,R5	EXPAND FROM END FORWARD	01683
2441	13BA	J550	33AD		MOV3 ETX,*R5	GUARANTEE <ETX> IS AT END OF TEXT	01684

ADD TEXT

01640

2442	13BE	A1C5		A	R5,R7	NEW ENDTEXT	01685	
2443	13C0	D5D5		EXPAND	MOV3 *R5,*R7	EXPAND TEXT TO MAKE ROOM FOR INSERTION	01686	
2444	13C2	0605		DEC	R5		01687	
2445	13C4	0607		DEC	R7		01688	
2446	13C6	8156		C	*R6,R5	EXPAND BACK TO START OF LINE # N+1	01689	
2447	13C8	12FB		JLE	EXPAND		01690	
2448	13CA	D5D5		MOV3	*R5,*R7	MOVE <CR-LF> FROM END OF LINE # N	01691	
2449	13CC	0D60	33AC	MOV3	NULL,*R5+	FILL EXPANDED SPACE WITH NULLS	01692	
2450	13D0	81C5		C	R5,R7		01693	
2451	13D2	1AFC		JL	*-6		01694	
2452	13D4	C1C6		MOV	R6,R7	SAVE CURRENT LINE START POINTER	01695	
2453	13D6	AD83		ADJSTR	A	ADJUST LINE START POSITIONS	01696	
2454	13D8	0286	D9F4	CI	R6,ENDTEXT		01697	
2455	13DC	12FC		JLE	ADJSTR		01698	
2456	13DE	0647		DECT	R7		01699	
2457	13E0	C157		MOV	*R7,R5	STARTING ADDRESS OF LINE # N	01700	
2458	13E2	C0E0	DA1E	MOV	TEXT0,R3	RESTORE R3	01701	
2459	13E6	0649		DECT	SOFT		01702	
2460	13E8	A179		A	*SOFT+,R5	STARTING ADDRESS OF INSERTION	01703	
2461	13EA	0605		DEC	R5		01704	
2462	13EC	C1A0	DA20	TEXTOK	MOV	TEXT1,R6	RESTORE R6	01705
2463	13F0	0649		DECT	SOFT		01706	
2464	13F2	A648		A	R8,*SOFT	ADJUST CHARACTER POSITION OF SOFTSTACK	01707	
2465	13F4	04C7		CLR	R7		01708	
2466	13F6	D1FC		INSERT	MOV3 *R12+,R7	INSERT INPUT STRING	01709	
2467	13F8	13FE		JEQ	INSERT	IGNORE NULL CHARACTERS (BYTE=\$00)	01710	
2468	13FA	9807	3433	CB	R7,SPACEBYT	CHECK IF THIS IS A SPACE (\$20)	01711	
2469	13FE	1602		JNE	*+6	IF SO, CHANGE IT TO A NULL (\$80)	01712	
2470	1400	D1E0	33AC	MOV3	NULL,R7		01713	
2471	1404	9807	33AC	CB	R7,NULL	CHECK IF CHARACTER IS A NULL (\$80)	01714	
2472	1408	1302		JEQ	*+6	IF SO, LEAVE IT ALONE	01715	
2473	140A	0247	7F00	ANDI	R7,\$7F00	MASK OFF HIGH ORDER BIT	01716	
2474	140E	DD47		MOV3	R7,*R5+	MOVE GOOD CHARACTERS TO TEXT	01717	
2475	1410	0608		DEC	R8		01718	
2476	1412	15F1		JGT	INSERT		01719	
2477	1414	C150	DA24	MOV	TEXT3,R5	RESTORE R5	01720	
2478	1418	045B		B	*R11		01721	

2480					*****			
2481					**		**	
2482					**	SETUP 7854 4-LINE READOUT DISPLAY BUFFER	**	
2483					**		**	
2484					**	LEVEL 2 ROUTINE	**	
2485					**		**	
2486					**	INPUT- NONE	**	
2487					**	OUTPUT- READOUT BUFFER SETUP	**	
2488					**	DESTROYS- R0-R8,R10,R11	**	
2489					**		**	
2490					**	STACK OPERATIONS-	**	
2491					**	SOFTSTK USED BUT NO AFFECT TO CALLER	**	
2492					**		**	
2493					**	NOTE ---	**	
2494					**	THIS ROUTINE IS CONTROLLED BY A TABLE WHICH DEFINES	**	
2495					**	THE CHARACTER STRING TO BE OUTPUT, I.E. WHERE	**	
2496					**	TO BEGIN OUTPUTING, AND WHAT FORMAT TO CALL	**	
2497					**	THE CONVERSION ROUTINE 'FP2ASC' WITH. EACH	**	
2498					**	VALUE MUST BE COMPUTED AND PUT ON THE SOFTSTK	**	
2499					**	BY INDIVIDUAL ROUTINES IN THIS PROGRAM.	**	
2500					**		**	
2501					*****			
2502					*			
2503					*	LINE #1 - OPW, VZR, VERT SCL, 10RZ SCL		
2504					*****			
2505					*			
2506	141A		0B80	R00T	WORD	WPLVL2	LEVEL 2 WORKSPACE	01723
2507	141C		141E		WORD	*+2		01724
2508	141E	C2A0	0A18	MOV	ROUPDT,R10		CHECK IF 100 MILLISECONDS HAS PASSED	01725
2509	1422	1101		JLT	*+4		SINCE LAST READOUT UPDATE	01726
2510	1424	0380		RTW ³			IF NOT, DON'T UPDATE READOUT THIS TIME	01727
2511	1426	C26D	0012	MOV	18(R13),SOFT		GET CURRENT SOFTSTACK POINTER	01728
2512	142A	04E0	0A18	CLR	ROUPDT		CLEAR READOUT UPDATE FLAG	01729
2513	142E	100A		JMP	CHKMODE			01730
2514	1430		0B80	FR00T	WORD	WPLVL2	LEVEL 2 WORKSPACE	01731
2515	1432		1434		WORD	*+2		01732
2516	1434	C26D	0012	MOV	18(R13),SOFT		GET CURRENT SOFTSTACK POINTER	01733
2517	1438	0550	0AC8	INV	0BUSY		UPDATE ENTIRE STATUS FIELD	01734
2518	143C	0560	0ACA	INV	0KEY			01735
2519	1440	0560	0ACC	INV	0PROGLN			01736
2520	1444	C2A0	09C6	CHKMODE	MOV	REALTIME,R10	CHECK DISPLAY MODE	01737
2521	1448	1108		JLT	CALCJP		CALCULATOR?	01738
2522	144A	1304		JEQ	RLTUP		REALTIME?	01739
2523	144C	1500		JGT	PROGUP		PROGRAM?	01740
2524	144E	0420	42FE	PROGUP	BLWP	TXTSCRN	PUT UP PROGRAM TEXT	01741
2525	1452	0380			RTW ³			01742
2526	1454	0420	1156	RLTJP	BLWP	RLTRO	PUT UP REALTIMEREADOUT	01743
2527	1458	0380			RTW ³			01744
2528	145A	C2A0	0972	CALCJP	MOV	RDTFLAG,R10	GET FLAG INDICATING WHICH LINES TO UPDATE	01745
2529	145E	04E0	0972		CLR	RDTFLAG	INDICATE READOUT HAS BEEN UPDATED	01746
2530	1462	091A		SRL	R10,1		SHIFT TO SEE IF BIT 0 IS A 1	01747
2531	1464	1751		JNC	LINE2		IF NOT, DON'T CHANGE THIS LINE	01748
2532	1466	C26D	0012		MOV	18(R13),SOFT	GET SOFTWARE STACK POINTER FROM CALLER	01749
2533					*****			
2534					*	PJT 'OPW XX' INTO READOUT DISPLAY BUFFER		
2535					*****			
2536	146A	0200	359E	LI	R0,OPW		GET CONTROL TABLE LISTING FOR OPW #	01750
2537	146E	C820	095C	0960	MOV	OPWFND,CRSOPWD	ADDRESS OF DISPLAYED OPW'S DATA	01751
2538	1474	C060	095A		MOV	OPWFN,R1	PUT CURRENT WAVEFORM NUMBER IN R1,R2	01752

2539	1478	06A0	757E	BL	INT2FP		01753
2540	147C	06A0	1816	BL	HRDVAL	OUTPUT STRING 'OPH' AND OPH #	01754
2541				*****			
2542				*	PJT 'VZR SXX,XX' INTO READOUT DISPLAY BUFFER		
2543				*****			
2544	1480	0200	35AA	LI	R0,VZR	GET CONTROL TABLE LISTING FOR VZR	01755
2545	1484	0420	67E6	BLWP	ZROREF	PUT ZERO REFERENCE IN R1,R2	01756
2546	1488	06A0	1816	BL	HRDVAL	OUTPUT STRING 'VZR' AND VERTICAL ZERO	01757
2547				*****			
2548				*	MOVE VERTICAL SCALE INTO READOUT BUFFER		
2549				*****			
2550	148C	0200	35B6	LI	R0,VERT	GET CONTROL TABLE LISTING FOR VERTICAL SCALE	01758
2551	1490	0420	6814	BLWP	VRTSCL	PUT VERTICAL SCALE IN R1,R2	01759
2552	1494	06A0	184A	BL	VALONLY	OUTPUT ONLY VERTICAL SCALE	01760
2553	1498	C120	D95E	MOV	OPWFHM,R4		01761
2554	149C	A120	334C	A	VSCALD,R4	POINTER TO VERTICAL UNITS	01762
2555	14A0	0649		DECT	SOFT		01763
2556	14A2	C644		MOV	R4,*SOFT	PUT POINTER TO UNITS ON STACK	01764
2557	14A4	0649		DECT	SOFT		01765
2558	14A6	C660	333A	MOV	C1,*SOFT	OUTPUT ONE CHARACTERS	01766
2559	14AA	06A0	136C	BL	TEXT	PUT UNITS INTO READOUT BUFFER	01767
2560				*****			
2561				*	PJT HORIZONTAL SCALE INTO READOUT BUFFER		
2562				*****			
2563	14AE	0200	35B0	LI	R0,HORZ	GET CONTROL TABLE LISTING FOR HORIZONTAL SCALE	01768
2564	14B2	C060	D978	MOV	XYWFM,R1	CHECK IF X-Y DISPLAY MODE	01769
2565	14B6	1319		JEQ	HORZYT	NO, PUT UP TIME HORIZONTAL	01770
2566	14B8	C0E0	D95E	HORZXY	MOV	OPWFHM,R3	SAVE OPWFHM
2567	14BC	C120	D95C	MOV	OPWFMD,R4	SAVE OPWFMD	01772
2568	14C0	0241	00FF	ANJI	R1,\$00FF	WFM # IS IN LOW BYTE	01773
2569	14C4	06A0	6984	BL	ADRWFM	GET HEADER AND DATA ADDRESSES OF VS WFM	01774
2570	14C8	C839	D95E	MOV	*SOFT+,OPWFHM	FOOL 'VRTSCL' TO GET VALUE OF VS WFM	01775
2571	14CC	C839	D95C	MOV	*SOFT+,OPWFMD		01776
2572	14D0	0420	6814	BLWP	VRTSCL	PUT VERTICAL SCALE IN R1,R2	01777
2573	14D4	C804	D95C	MOV	R4,OPWFMD	RESTORE OPWFMD	01778
2574	14D8	C120	D95E	MOV	OPWFHM,R4	SAVE VS WFM'S HEADER ADDRESS	01779
2575	14DC	C803	D95E	MOV	R3,OPWFHM	RESTORE OPWFHM	01780
2576	14E0	A120	334C	A	VSCALD,R4		01781
2577	14E4	06A0	184A	BL	VALONLY	OUTPUT VERTICAL SCALE	01782
2578	14E8	1008		JMP	HSCLU		01783
2579	14EA	0420	6844	HORZYT	BLWP	HRZSCL	PUT HORIZONTAL SCALE IN R1,R2
2580	14EE	06A0	184A	BL	VALONLY	OUTPUT ONLY HORIZONTAL SCALE	01785
2581	14F2	C120	D95E	MOV	OPWFHM,R4		01786
2582	14F6	A120	3350	A	HSCALD,R4	POINTER TO VERTICAL UNITS	01787
2583	14FA	0649		HSCLU	DECT	SOFT	01788
2584	14FC	C644		MOV	R4,*SOFT	PUT POINTER TO UNITS ON STACK	01789
2585	14FE	0649		DECT	SOFT		01790
2586	1500	C660	333A	MOV	C1,*SOFT	OUTPUT ONE CHARACTER	01791
2587	1504	06A0	136C	BL	TEXT	PUT UNITS INTO READOUT BUFFER	01792
2588				*			
2589				*	LINE #2 - WFM #S, VS		
2590				*****			
2591				*			
2592	1508	091A		LINE2	SRL	R10,1	SHIFT TO SEE IF BIT 1 IS A 1
2593	150A	172F		JNC	LINE15		IF NOT, DON'T CHANGE THIS LINE
2594	150C	C260	0012	MOV	18(R13),SOFT	GET SOFTWARE STACK POINTER FROM CALLER	01795
2595	1510	0649		DECT	SOFT		01796
2596	1512	C660	333C	MOV	C2,*SOFT	BLANK LINE #2	01797
2597	1516	0649		DECT	SOFT		01798

SETUP 7854 4-LINE READOUT

01722

2598	1518	C660	333A	MOV	C1,*SOFT		01799
2599	151C	05A0	12FA	BL	NULLIN		01800
2600	1520	0229	0004	AI	SOFT,4		01801
2601	1524	0420	1898	BLWP	WFMSRT	SORT WAVEFORM NUMBERS	01802
2602	1528	C060	0980	MOV	DSPBUF+2,R1	CHECK IF ONLY CURRENT WFM IS	01803
2603	152C	1313		JEQ	VSOUT	BEING DISPLAYED. IF SO, DON'T OUTPUT ITS #	01804
2604	152E	0200	35C2	LI	R0,DSW	GET CONTROL TABLE LISTING FOR WFM NUMBERS	01805
2605	1532	06A0	181E	BL	WRDONLY	PUT STRING 'DSW ' INTO READOUT	01806
2606	1536	0203	097E	LI	R3,DSP3UF		01807
2607	153A	C073		NXTWFM	MOV *R3+,R1	GET NEXT WFM # IN R1	01808
2608	153C	1308		JEQ	VSOUT	IF ZERO WE ARE DONE	01809
2609	153E	0241	00FF	ANDI	R1,\$00FF	WFM # IS IN LOW BYTE	01810
2610	1542	06A0	757E	BL	INT2=FP	CONVERT NUMBER TO FLOATING POINT	01811
2611	1546	0205	154E	LI	R5,VNRTRN	RETURN POINTER FOR 'VALNXT'	01812
2612	154A	0460	1854	B	VALNXT	OUTPUT WFM NUMBER	01813
2613	154E	0640		VNRTRN	DECT R0	RESET FORMAT POINTER	01814
2614	1550	0599		INC	*SOFT	1 SPACE BETWEEN WAVEFORM NUMBERS	01815
2615	1552	18F3		JMP	NXTWFM		01816
2616	1554	C060	D978	VSOJT	MOV XYWFM,R1	CHECK IF X-Y DISPLAY	01817
2617	1558	1308		JEQ	LINE15	IF NOT, DON'T INCLUDE VS FIELD	01818
2618	155A	0241	00FF	ANDI	R1,\$00FF	VS # IS IN LOW BYTE	01819
2619	155E	0200	35CE	LI	R0,VS	GET CONTROL TABLE LISTING FOR VS	01820
2620	1562	06A0	757E	BL	INT2=FP	CONVERT TO FLOATING POINT	01821
2621	1566	06A0	1816	BL	WRDVAL	OUTPUT 'VS ##'	01822
2622				*			
2623				*	LINE #15 - VERTICAL CURSOR POSITION, HORIZONTAL CURSOR POSITION		
2624				*****			
2625				*			
2626	156A	091A		LINE15	SRL R10,1	SHIFT TO SEE IF BIT 2 IS A 1	01823
2627	156C	176C		JNC	ULINE16	IF NOT, DON'T CHANGE THIS LINE	01824
2628	156E	C26D	0012	MOV	16(R13),SOFT	GET SOFTWARE STACK POINTER FROM CAL.LER	01825
2629	1572	C0E0	0954	MOV	CURSJR,R3	CHECK IF CURSORS ARE ON	01826
2630	1576	1508		JGT	VCROUT	IF NOT, JUST BLANK LINE	01827
2631	1578	0649		DECT	SOFT		01828
2632	157A	C660	3356	MOV	C15,*SOFT	BLANK LINE #15	01829
2633	157E	0649		DECT	SOFT		01830
2634	1580	C660	333A	MOV	C1,*SOFT		01831
2635	1584	06A0	1332	BL	DELIN		01832
2636	1588	0229	0004	AI	SOFT,4		01833
2637	158C	105C		JMP	ULINE15		01834
2638				*****			
2639				*	PJT 'VCR SX.XXESXX' INTO READOUT BUFFER		
2640				*****			
2641	158E	0200	35DA	VCRJUT	LI R0,VCR	GET CONTROL TABLE LISTING FOR VCR	01835
2642	1592	0643		DECT	R3		01836
2643	1594	1102		JLT	*+6	CHECK IF DELTA POSITION	01837
2644	1596	0200	35FA	LI	R0,DVCR		01838
2645	159A	05C3		INCT	R3		01839
2646	159C	0420	6880	BLWP	VCRPOS	PUT VERTICAL CURSOR POSITION IN R1,R2	01840
2647	15A0	06A0	1816	BL	WRDVAL	OUTPUT STRING '^VCR' AND VCR POSITION	01841
2648	15A4	C120	095E	MOV	OPWFMH,R4		01842
2649	15A8	A120	334C	A	VSCALD,R4	POINTER TO VERTICAL UNITS	01843
2650	15AC	0649		DECT	SOFT		01844
2651	15AE	C644		MOV	R4,*SOFT	PUT POINTER TO UNITS ON STACK	01845
2652	15B0	0649		DECT	SOFT		01846
2653	15B2	C660	333A	MOV	C1,*SOFT	OUTPUT ONE CHARACTER	01847
2654	15B6	06A0	136C	BL	TEXT	PUT UNITS INTO READOUT BUFFER	01848
2655	15BA	0200	35EA	LI	R0,HCR	GET STARTING CHARACTER POSITION FOR	01849
2656	15BE	05C0		INCT	R0	HORIZONTAL CURSOR VALUE	01850

2657	15C0	C010		MOV	*R0,R0		01851
2658	15C2	6019		S	*SOFT,R0	CALCULATE SPACES FROM END OF VERTICAL CURSOR	01852
2659	15C4	C040		MOV	R0,R1	VALUE TO HORIZONTAL CURSOR VALUE	01853
2660	15C6	0649		BLKFL	DECT	SOFT	01854
2661	15C8	C660	34E0	MOV	BLANK,*SOFT	BLANK OUT THIS SPACE	01855
2662	15CC	0601		DEC	R1		01856
2663	15CE	15FB		JGT	BLKFL		01857
2664	15D0	0649		DECT	SOFT		01858
2665	15D2	C640		MOV	R0,*SOFT		01859
2666	15D4	06A0	1362	BL	TEXTS		01860
2667				*****			
2668				*	PJT 'HCR SXX.XKESXX' INTO READOUT BUFFER		
2669				*****			
2670	15D8	0200	35EA	LI	R0,HCR	GET CONTROL TABLE LISTING FOR HCR	01861
2671	15DC	0643		DECT	R3		01862
2672	15DE	1102		JLT	*+6	CHECK IF DELTA POSITION	01863
2673	15E0	0200	360A	LI	R0,DHCR		01864
2674	15E4	C060	D978	MOV	XYWFM,R1	CHECK IF X-Y DISPLAY MODE	01865
2675	15E8	131D		JEQ	HCRYT	NO, PUT TIME HCR	01866
2676	15EA	C0E0	D95E	HCRXY	MOV	OPWFHM,R3	SAVE OPWFHM
2677	15EE	C120	D95C	MOV	OPWFMD,R4	SAVE OPWFMD	01868
2678	15F2	0241	00FF	ANDI	R1,\$0FF	WFM # IS IN LOW BYTE	01869
2679	15F6	06A0	6984	BL	ADRFWM	GET HEADER AND DATA ADDRESSES FOR VS WFM	01870
2680	15FA	0300	0007	LIMI	7	DON'T LET CURSOR DISPLAY GET BAD 'OPWFMD'	01871
2681	15FE	C839	D95E	MOV	*SOFT+,OPWFHM	FOOL 'VCRPOS' TO GET VALUE OF VS WFM	01872
2682	1602	C839	D95C	MOV	*SOFT+,OPWFMD		01873
2683	1606	0420	6880	BLWP	VCRPJS	GET VERTICAL CURSOR POSITION	01874
2684	160A	C804	D95C	MOV	R4,OPWFMD	RESTORE OPWFMD	01875
2685	160E	C120	D95E	MOV	OPWFHM,R4	KEEP VS WFM #	01876
2686	1612	C803	D95E	MOV	R3,OPWFHM	RESTORE OPWFHM	01877
2687	1616	0300	000F	LIMI	\$F		01878
2688	161A	06A0	1816	BL	WRDVAL		01879
2689	161E	A120	334C	A	VSCALD,R4	USE VERTICAL UNITS	01880
2690	1622	1008		JMP	HCRSU		01881
2691	1624	0420	68E0	HCRYT	BLWP	HCRPJS	PUT HORIZONTAL CURSOR POSITION IN R1,R2
2692	1628	06A0	1816	BL	WRDVAL	OUTPUT STRING '^HCR' AND HCR POSITION	01883
2693	162C	C120	D95E	MOV	OPWFHM,R4		01884
2694	1630	A120	3350	A	HSCALD,R4	POINTER TO VERTICAL UNITS	01885
2695	1634	0649		HCRSU	DECT	SOFT	01886
2696	1636	C644		MOV	R4,*SOFT	PUT POINTER TO UNITS ON STACK	01887
2697	1638	0649		DECT	SOFT		01888
2698	163A	C660	333A	MOV	C1,*SOFT	OUTPUT ONE CHARACTER	01889
2699	163E	06A0	136C	BL	TEXT	PUT UNITS INTO READOUT BUFFER	01890
2700	1642	06A0	12FA	BL	NULLIN	BLANK OUT REST OF LINE 15	01891
2701				*			
2702				*	LINE #16 - Y-STACK, X-STACK, KEY, STATUS, PROGRAM LINE #		
2703				*****			
2704				*			
2705	1646	091A		ULINE15	SRL	R10,1	SHIFT TO SEE IF BIT 3 IS A 1
2706	1648	173B		JNC	PLOUT	IF NOT, CHECK LAST 3 FIELDS	01893
2707	164A	C260	0012	MOV	18(R13),SOFT	GET SOFTWARE STACK POINTER FROM CALLER	01894
2708				*****			
2709				*	PUT Y-STACK ELEMENT INTO READOUT BUFFER		
2710				*****			
2711	164E	C0A0	3308	MOV	YSTK,R2	GET POINTER TO Y-STACK ELEMENT	01895
2712	1652	C032		MOV	*R2+,R0	GET FP/WFM # INDICATOR	01896
2713	1654	1607		JNE	WFMY	IF WFM # USE CORRESPONDING FORMAT	01897
2714	1656	0200	361A	LI	R0,YSTCK	GET CONTROL TABLE LISTING FOR FP Y-STACK #	01898
2715	165A	C072		MOV	*R2+,R1	PUT FLOATING POINT Y ELEMENT INTO R1,R2	01899

2716	165C	C092		MOV	*R2,R2		01900
2717	165E	06A0	184A	BL	VALONLY	OUTPUT VALUE OF Y ELEMENT ONLY	01901
2718	1662	100F		JMP	XOUT		01902
2719	1664	0200	3626	WFMY	LI	R0,YWFM	01903
2720	1668	C052		MOV	*R2,R1	GET CONTROL TABLE LISTING FOR WFM # Y-STACK	01904
2721	166A	06A0	757E	BL	INT2FP	PUT WFM # IN R1,R2	01905
2722	166E	06A0	1842	BL	VALWRD	OUTPUT WFM # THEN STRING 'WFM'	01906
2723	1672	0649		DECT	SOFT		01907
2724	1674	C660	34E0	MOV	BLANK,*SOFT	PUSH ' ' ONTO SOFTSTACK	01908
2725	1678	0649		DECT	SOFT		01909
2726	167A	C660	333A	MOV	C1,*SOFT	CHARACTER COUNT IS 1	01910
2727	167E	06A0	1362	BL	TEXTS	PUT SPACE BETWEEN X & Y DISPLAYES	01911
2728							
2729							
2730							
2731	1682	C0A0	D98A	XOUT	MOV	PROGRS,R2	01912
2732	1686	1301		JEQ	*+4	CHECK IF NUMERIC ENTRY IS IN PROGRESS	01913
2733	1688	0380		RTWP		YES, THEN RETURN	01914
2734	168A	C0A0	3306	MOV	XSTK,R2	GET POINTER TO X-STACK ELEMENT	01915
2735	168E	C032		MOV	*R2+,R0	GET FP/WFM # INDICATOR	01916
2736	1690	1607		JNE	WFMX	IF WFM # USE CORRESPONDING FORMAT	01917
2737	1692	0200	3620	LI	R0,XSTCK	GET CONTROL TABLE LISTING FOR FP X-STACK #	01918
2738	1696	C072		MOV	*R2+,R1	PUT FLOATING POINT X ELEMENT INTO R1,R2	01919
2739	1698	C092		MOV	*R2,R2		01920
2740	169A	06A0	184A	BL	VALONLY	OUTPUT VALUE OF X ELEMENT ONLY	01921
2741	169E	1010		JMP	PLOUT		01922
2742	16A0	0200	3632	WFMX	LI	R0,XWFM	01923
2743	16A4	C052		MOV	*R2,R1	GET CONTROL TABLE LISTING FOR WFM # X-STACK	01924
2744	16A6	06A0	757E	BL	INT2FP	PUT WFM # IN R1,R2	01925
2745	16AA	06A0	1842	BL	VALWRD	OUTPUT WFM # THEN STRING 'WFM'	01926
2746	16AE	1008		JMP	PLOUT		01927
2747	16B0		0B80	PRGSTAT	WORD	WPLVL2	01928
2748	16B2		16B4	WORD	*+2		01929
2749	16B4	C0E0	D9C6	MOV	REALTIME,R3	UPDATE PROGRAM STATUS FIELD ONLY	01930
2750	16B8	1103		JLT	PLOUT	WHEN CALCULATOR DISPLAY IS ON	01931
2751	16BA	0720	DAC8	PRGRTWP	SETO	OBUSY	01932
2752	16BE	0380		RTWP		AFTER NUMERIC ENTRY REDISPLAY 'BUSY'	01933
2753							
2754							
2755							
2756	16C0	C260	0012	PLOUT	MOV	18(R13),SOFT	01934
2757	16C4	C020	DAAA	MOV	EDITPNT,R0	GET SOFTWARE STACK POINTER FROM CALLER	01935
2758	16C8	9810	3820	CB	*R0,ENDKEYB	NO PL LINE # IF AT END OF PROGRAM	01936
2759	16CC	130E		JEQ	NOPL		01937
2760	16CE	C060	DAA8	MOV	PROGLN,R1		01938
2761	16D2	8801	DACC	C	R1,OPROGLN	HAS PL # CHANGED?	01939
2762	16D6	130D		JEQ	KEYSTAT	NO, DON'T BOTHER UPDATING IT	01940
2763	16D8	C801	DACC	MOV	R1,OPROGLN	SAVE LINE NUMBER	01941
2764	16DC	0200	36E0	LI	R0,PL	GET CONTROL TABLE LISTING FOR 'PL ##'	01942
2765	16E0	06A0	757E	BL	INT2FP	CONVERT PROGRAM LINE # TO FP	01943
2766	16E4	06A0	1816	BL	WRDVAL	OUTPUT STRING 'PL' AND LINE NUMBER	01944
2767	16E8	1004		JMP	KEYSTAT		01945
2768	16EA	0200	36D4	NOPL	LI	R0,PLBLNK	01946
2769	16EE	06A0	181E	BL	WRDONLY	FIELD OF SPACES	01947
2770							
2771							
2772			16F2	KEYSTAT	EQU	*	01948
2773	16F2	0204	0003	LI	R4,3	MAXIMUM OF THREE CHARACTERS IN MNEMONIC	01949
2774	16F6	C0E0	D968	MOV	LASTKEY,R3	LOAD R3 WITH KEY NUMBER	01950

2775	16FA	C060	D98A		MOV	PROGRS,R1	CHECK IF NUMERIC ENTRY IN PROGRESS	01951
2776	16FE	1628			JNE	STPNOUT	IF SO, CHECK FOR 'STOP IN' STATUS ONLY	01952
2777	1700	C060	D976		MOV	INUSE,R1	CHECK IF 'BUSY' SHOULD BE DISPLAYED	01953
2778	1704	1311			JEQ	ERRUP		01954
2779	1706	8820	D967	3811	C	KEYB,PAUSKEYB	SHOULD 'PAUSE' BE SUBSTITUTED FOR 'BUSY'	01955
2780	170C	1605			JNE	BUSYJP		01956
2781	170E	0720	DAC8		SETO	OBUSY	FLAG INDICATING 'BUSY' NOT DISPLAYED	01957
2782	1712	0200	3658		LI	R0,PAUSE	CONTROL TABLE ENTRY FOR 'PAUSE'	01958
2783	1716	103B			JMP	STSTOUT	PUT 'PAUSE' IN STATUS FIELD	01959
2784	1718	C060	DAC8		BUSYJP	MOV	OBUSY,R1	IS 'BUSY' UP?
2785	171C	1350			JEQ	KEYOJT	YES, DON'T BOTHER PUTTING IT UP AGAIN	01961
2786	171E	04E0	DAC8		CLR	OBUSY	NO, FLAG SAYING IT IS NOW UP	01962
2787	1722	0200	3666		LI	R0,BUSY1	GET CONTROL TABLE LISTING FOR 'BUSY'	01963
2788	1726	1033			JMP	STSTOUT	OUTPUT STRING 'BUSY'	01964
2789	1728	0720	DAC8		ERRUP	SETO	OBUSY	FLAG INDICATING 'BUSY' NOT DISPLAYED
2790	172C	C060	D94A		MOV	FATAL,R1	CHECK ERROR CODE	01966
2791	1730	1103			JLT	WARNOUT	IF NEGATIVE THERE IS NO ERROR	01967
2792	1732	0200	3674		LI	R0,ERR	GET CONTROL TABLE LISTING FOR 'ERR ##'	01968
2793	1736	102B			JMP	STSTOUT	PUT 'ERROR' IN STATUS FIELD	01969
2794	1738	C060	D94C		WARNOJT	MOV	WARNING,R1	CHECK WARNING CODE
2795	173C	1103			JLT	ENDOUT	IF NEGATIVE THERE IS NO WARNING	01971
2796	173E	0200	3682		LI	R0,WARN	GET CONTROL TABLE LISTING FOR 'WARN ##'	01972
2797	1742	1025			JMP	STSTOUT	PUT 'WARNING' IN STATUS FIELD	01973
2798	1744	C060	DAC2		ENDOUT	MOV	EOPFLG,R1	CHECK IF 'END' SHOULD BE PUT UP
2799	1748	1303			JEQ	STPNOUT		01975
2800	174A	0200	364A		LI	R0,ENDSTAT	GET CONTROL TABLE LISTING FOR 'END'	01976
2801	174E	101F			JMP	STSTOUT	PUT 'END' IN STATUS FIELD	01977
2802	1750	C060	DA3A		STPNOUT	MOV	STPFLG,R1	CHECK IF 'STOP IN' SHOULD BE PUT
2803	1754	1103			JLT	TR.FA	IN STATUS FIELD	01979
2804	1756	0200	3690		LI	R0,STOPIN	GET CONTROL TABLE LISTING FOR 'STOP IN'	01980
2805	175A	1019			JMP	STSTOUT	PUT 'STOP IN' IN STATUS FIELD	01981
2806	175C	C060	D98A		TR.FA	MOV	PROGRS,R1	01982
2807	1760	1610			JNE	CLRST	IF IN NUMERIC ENTRY, BLANK STATUS FIELD	01983
2808	1762	C060	D96A		MOV	TRUE,R1	SHOULD 'TRUE' OF 'FALSE' BE DISPLAYED?	01984
2809	1766	1107			JLT	NOSTST	NO, THEN BLANK FIELD	01985
2810	1768	1303			JEQ	SFALSE	0 => 'FALSE'	01986
2811	176A	0200	369E		STRJE	LI	R0,TRUES	1 => 'TRUE'
2812	176E	100F			JMP	STSTOUT	PUT 'TRUE' IN STATUS FIELD	01988
2813	1770	0200	36AC		SFALSE	LI	R0,FALSES	
2814	1774	100C			JMP	STSTOUT	PUT 'FALSE' IN STATUS FIELD	01990
2815			1776		NOSTST	EQU	*	01991
2816	1776	0204	000B		LI	R4,11	MAXIMUM OF ELEVEN CHARACTER IN MNEMONIC	01992
2817	177A	C060	DAC6		MOV	BLNKFLG,R1	IS FIELD ALREADY BLANK?	01993
2818	177E	161F			JNE	KEYOUT	YES, DON'T BLANK FIELD AGAIN	01994
2819	1780	0200	368A		LI	R0,BLANKST	BLANK STATUS FIELD	01995
2820	1784	04E0	DACA		CLR	OKEY	FORCE KEY MNEMONIC DISPLAY	01996
2821	1788	0720	DAC5		SETO	BLNKFLG	SET FIELD BLANK FLAG	01997
2822	178C	1004			JMP	*+10		01998
2823			178E		STSTOUT	EQU	*	01999
2824	178E	04E0	DACA		CLR	OKEY	FORCE KEY TO BE DISPLAYED	02000
2825	1792	04E0	DAC6		CLR	BLNKFLG	INDICATE FIELD NO LONGER BLANK	02001
2826	1796	06A0	181E		BL	WRDONLY	PUT UP STATUS FIELD	02002
2827	179A	1011			JMP	KEYOUT	PUT UP KEY MNEMONIC	02003
2828	179C	0200	36C8		CLRST	LI	R0,BLNKOV	CLEAR LAST FIVE CHARACTERS OF STATUS
2829	17A0	06A0	181E		BL	WRDONLY	DURING NUMERIC ENTRY ONLY	02005
2830	17A4	0380			RTWP			02006
2831	17A6		DB80		KEYJP	WORD	HPLVL2	LEVEL 2 ROUTINE
2832	17A8		17AA		WORD	*+2		02008
2833	17AA	C260	0012		MOV	18(R13),SOFT	GET CURRENT SOFTSTACK POINTER	02009

2834	17AE	00E0	09C6	MOV	REALTIME,R3	PUT UP KEY IN CALC MODE ONLY	02010
2835	17B2	1101		JLT	*+4		02011
2836	17B4	0380		RTWP			02012
2837	17B6	0204	0003	LI	R4,3		02013
2838	17BA	00E0	0966	MOV	KEY,R3	GET KEYCODE OF KEY TO DISPLAY	02014
2839	17BE	0243	00FF	KEYOUT ANDI	R3,\$FF	KEEP ONLY KEY NUMBER	02015
2840	17C2	8803	DACA	C	R3,OKEY	IS THIS KEY MNEMONIC ALREADY DISPLAYED?	02016
2841	17C6	1601		JNE	*+4		02017
2842	17C8	0380		RTWP		YES, RETURN	02018
2843	17CA	0803	DACA	MOV	R3,OKEY	NO, SAVE THIS AS LAST KEY MNEMONIC DISPLAYED	02019
2844	17CE	0A23		SLA	R3,2		02020
2845	17D0	0223	3836	AI	R3,KEYTAB	MNEMONIC ADDRESS = (KEY*4)+KEYTAB+2	02021
2846	17D4	05C3		INCT	R3		02022
2847	17D6	00D3		MOV	*R3,R3	ADDRESS OF MNEMONIC IS INDIRECT	02023
2848	17D8	0200	363E	LI	R0,KEYMNU	BLANK KEY MNEMONIC FIELD	02024
2849	17DC	06A0	181E	BL	WRDONLY		02025
2850	17E0	0200	363E	LI	R0,KEYMNU	GET CONTROL TABLE LISTING FOR MNEMONIC	02026
2851	17E4	0649		DECT	SOFT		02027
2852	17E6	0670		MOV	*R0+,*SOFT	PUT LINE # ON STACK	02028
2853	17E8	0649		DECT	SOFT		02029
2854	17EA	0670		MOV	*R0+,*SOFT	PUT CHARACTER # ON STACK	02030
2855	17EC	0599		INC	*SOFT		02031
2856	17EE	0649		DECT	SOFT		02032
2857	17F0	0643		MOV	R3,*SOFT	PUT POINTER TO MNEMONIC ON STACK	02033
2858	17F2	0708		SETJ	R11		02034
2859	17F4	0588		INC	R11		02035
2860	17F6	9833	3338	CB	*R3+,C0	COUNT CHARACTERS IN MNEMONIC	02036
2861	17FA	16FC		JNE	*-6		02037
2862	17FC	8108		C	R11,R4		02038
2863	17FE	1101		JLT	*+4		02039
2864	1800	02C4		MOV	R4,R11	MAXIMUM SPACE AVAILABLE FOR MNEMONIC	02040
2865	1802	0649		DECT	SOFT		02041
2866	1804	0648		MOV	R11,*SOFT	PUT CHARACTER COUNT ON STACK	02042
2867	1806	06A0	136C	BL	TEXT	PUT MNEMONIC IN READOUT BUFFER	02043
2868	180A	0380		RTWP		RETURN TO CALLER	02044
2869	180C	0200	363E	NOKEY LI	R0,KEYMNU	BLANK KEY MNEMONIC FIELD	02045
2870	1810	06A0	181E	BL	WRDONLY		02046
2871	1814	0380		RTWP			02047
2872				*****			
2873	1816	0148		WRDVAL MOV	R11,R5	SAVE FINAL RETURN ADDRESS	02048
2874	1818	0206	1854	LI	R6,VALNXT	SETUP RETURNS TO OUTPUT STRING THEN VALUE	02049
2875	181C	1001		JMP	WRDONLY+2		02050
2876	181E	0188		WRDONLY MOV	R11,R6		02051
2877	1820	0649		DECT	SOFT		02052
2878	1822	0670		MOV	*R0+,*SOFT	PUT LINE # ON STACK	02053
2879	1824	0649		DECT	SOFT		02054
2880	1826	0670		MOV	*R0+,*SOFT	PUT CHARACTER # ON STACK	02055
2881	1828	0229	FFFC	WRDNXT AI	SOFT,-4		02056
2882	182C	0E70		MOV	*R0+,*SOFT+	PUT CHARACTER COUNT ON STACK	02057
2883	182E	0640		MOV	R0,*SOFT	PUT CHARACTER STRING ADDRESS ON STACK	02058
2884	1830	0649		DECT	SOFT		02059
2885	1832	A019		A	*SOFT,R0	ADJUST R0 TO POINT TO CONVERSION FORMAT WORD	02060
2886	1834	2020	333A	COC	C1,R0		02061
2887	1838	1601		JNE	*+4		02062
2888	183A	0580		INC	R0	FOR ODD SIZES ADD 1 TO INSURE EVEN BOUNDARY	02063
2889	183C	06A0	136C	BL	TEXT	PUT TEXT INTO READOUT BUFFER	02064
2890	1840	0456		B	*R6		02065
2891	1842	0188		VALWRJ MOV	R11,R6	SAVE FINAL RETURN ADDRESS	02066
2892	1844	0205	1828	LI	R5,WRDNXT	SETUP RETURNS TO OUTPUT VALUE THEN STRING	02067

SETUP 7854 4-LINE READOUT

01722

2893	1848	1001		JMP	VALONLY+2		02068
2894	184A	C148		VALONLY MOV	R11,R5		02069
2895	184C	0649		DECT	SOFT		02070
2896	184E	C670		MOV	*R0+,*SOFT	PUT LINE # ON STACK	02071
2897	1850	0649		DECT	SOFT		02072
2898	1852	C670		MOV	*R0+,*SOFT	PUT CHARACTER # ON STACK	02073
2899	1854	0649		VALVXT DECT	SOFT		02074
2900	1856	C642		MOV	R2,*SOFT	PUT EXPONENT ON STACK	02075
2901	1858	0649		DECT	SOFT		02076
2902	185A	C641		MOV	R1,*SOFT	PUT MANTISSA ON STACK	02077
2903	185C	0649		DECT	SOFT		02078
2904	185E	C670		MOV	*R0+,*SOFT	PUT CONVERSION FORMAT ON SOFTSTACK	02079
2905	1860	0420	75E0	BLWP	FP2ASC	CONVERT FLOATING POINT NUMBER TO ASCII	02080
2906	1864	0759		ABS	*SOFT	SAFETY BLOWUP PROTECT	02081
2907	1866	06A0	1362	BL	TEXTS	MOVE VALUE TO READOUT BUFFER	02082
2908	186A	0455		B	*R5		02083

VERTICAL POSITION KEYS' READOUT UPDATE

02084

```

2910 *****
2911 **
2912 ** UPDATE VERTICAL ZERO READOUT FOR 'VPUP' & 'VPDN' KEYS **
2913 **
2914 ** LEVEL 2 ROUTINE **
2915 **
2916 *****
2917 186C DB80 VZRRJ WORD WPLVL2 02085
2918 186E 1870 WORD *+2 02086
2919 1870 C060 D9C5 MOV REALTIME,R1 UPDATE READOUT ONLY IN CALCULATOR MODE 02087
2920 1874 1101 JLT *+4 02088
2921 1876 0380 RTWP 02089
2922 1878 C260 0012 MOV 18(R13),SOFT GET SOFTSTACK POINTER 02090
2923 187C 0200 35AA LI R0,VZR GET TABLE LISTING FOR 'VZR XX.XX' 02091
2924 1880 0420 67E5 BLW ZROREF CALCULATE ZERO REFERENCE 02092
2925 1884 06A0 1816 BL WRDVAL PUT NEW ZERO REFERENCE INTO READOUT BUFFER 02093
2926 1888 C060 094C MOV WARNING,R1 SHOULD WARNING BE ADDED? 02094
2927 188C 1104 JLT NOVZRW NO, RETURN TO CALLER 02095
2928 188E 0200 3682 LI R0,WARN YES, GET TABLE LISTING FOR 'WARN NN' 02096
2929 1892 06A0 181E BL WRDONLY PUT WARNING INTO READOUT BUFFER 02097
2930 1896 0380 NOVZRW RTWP 02098
    
```

SORT WAVEFORMS

02099

```

2932 *****
2933 **
2934 ** SETUP AND SORT WFM NUMBERS BEING DISPLAYED **
2935 ** **
2936 ** LEVEL 3 ROUTINE **
2937 ** **
2938 ** INPUT- NONE **
2939 ** OUTPUT- NONE **
2940 ** DESTROYS- R0-R7 **
2941 ** **
2942 *****
2943 1898 DBA0 WFM SORT WORD WPLVL3 02100
2944 189A 189C WORD *+2 02101
2945 189C C25D 0012 MOV 18(R13),SOFT GET SOFTSTACK POINTER FROM CALLER 02102
2946 18A0 04C1 CLR R1 CLEAR POINTER TO NEXT WAVEFORM TO SORT 02103
2947 18A2 04C2 CLR R2 CLEAR POINTER TO NEXT DSPBUF CELL 02104
2948 18A4 06A0 6984 CHKWFM BL ADRWFM GET HEADER AND DATA ADDRESS OF WFM R1 02105
2949 18A8 C0F9 MOV *SOFT+,R3 MOVE HEADER ADDRESS TO R3 02106
2950 18AA C139 MOV *SOFT+,R4 MOVE DATA ADDRESS TO R4 02107
2951 18AC A0E0 3354 A DISPLA,R3 POINT TO DISPLAY FLAG IN HEADER 02108
2952 18B0 C0D3 MOV *R3,R3 02109
2953 18B2 0243 0003 ANDI R3,3 CHECK IF WAVEFORM IS BEING DISPLAYED 02110
2954 18B6 1607 JNE WFMON 02111
2955 18B8 0581 NEXTWFM INC R1 IF NO, CHECK NEXT WAVEFORM 02112
2956 18BA 8801 0974 C R1,WFMAXN CHECK IF ALL WAVEFORMS HAVE BEEN TESTED 02113
2957 18BE 12F2 JLE CHKWFM IF NOT, CONTINUE 02114
2958 18C0 04E2 097E SORTED CLR DSPBUF(R2) ADD END FLAG ($0000) 02115
2959 18C4 0380 RTWP 02116
2960
2961 18C6 C001 WFMON MOV R1,R0 SAVE NEXT WFM NUMBER 02117
2962 18C8 C0C2 MOV R2,R3 SAVE NEXT AVAILABLE CELL IN DSPBUF 02118
2963 18CA C144 MOV R4,R5 SAVE STARTING ADDRESS OF DATA 02119
2964 18CC 0643 DECT R3 GET NEXT WAVEFORM TO CHECK 02120
2965 18CE 110F JLT INSWFM IF NO MORE INSERT WFM # AT BEGINNING 02121
2966 18D0 C063 097E MOV DSPBUF(R3),R1 GET WFM # 02122
2967 18D4 0241 00FF ANDI R1,$00FF KEEP WFM # ONLY 02123
2968 18D8 06A0 6984 BL ADRWFM GET HEADER AND DATA ADDRESS FOR WFM 02124
2969 18DC 05C9 INCT SOFT HEADER ADDRESS 02125
2970 18DE C189 MOV *SOFT+,R6 DATA ADDRESS 02126
2971 18E0 C1E0 0970 MOV RESOLV,R7 CHECK ALL POINTS IF NECESSARY 02127
2972 18E4 8095 C *R5+,*R6+ CHECK ELEMENTS OF EACH WAVEFORM 1 BY 1 02128
2973 18E6 150A JGT CHKNXTW IF GREATER THAN CHECK AGAINST NEXT WAVEFORM 02129
2974 18E8 1102 JLT INSWFM IF LESS THAN INSERT HERE 02130
2975 18EA 0607 DEC R7 IF EQUAL CHECK NEXT ELEMENT 02131
2976 18EC 15FB JGT *-8 IF ALL ELEMENT EQUAL INSERT WFM # HERE 02132
2977 18EE 05C3 INSWFM INCT R3 02133
2978 18F0 C040 MOV R0,R1 RESTORE POINTER TO NEXT WFM TO SORT 02134
2979 18F2 A004 A R4,R0 ADD DATA ADDRESS TO WFM # 02135
2980 18F4 C8C0 097E MOV R0,DSPBUF(R3) INSERT WFM# AT THIS POINT 02136
2981 18F8 05C2 INCT R2 POINT TO NEXT AVAILABLE CELL 02137
2982 18FA 10DE JMP NEXTWFM 02138
2983 18FC C8E3 097E 0980 C4KNXTW MOV DSPBUF(R3),DSPBUF+2(R3) MOVE WFM # UP ONE SLOT 02139
2984 1902 10E3 JMP WFMON+4 02140

```

WAVEFORM ACQUISITION KEYS

02141

```

2986 *****
2987 **
2988 ** WAVEFORM ACQUISITION KEYS: 'AQR', 'AQS', 'AVG' **
2989 ** **
2990 ** LEVEL 1 ROUTINE **
2991 ** **
2992 ** INPJT: NONE **
2993 ** OUTPUT: NONE **
2994 ** DESTROYS: NONE **
2995 ** **
2996 ** STACK OPERATIONS: **
2997 ** SOFTSTACK: USED BUT NO EFFECT TO CALLER **
2998 ** USERSTACK: 'AVE' POPS 1 IF GOOD ARGUMENT **
2999 **
3000 *****
3001 1904 0200 1CG2 KEYAQR LI R0,AQRWFM 'AQR' KEY WAVEFORM ACQUISITION ROUTINE 02142
3002 1908 0203 0002 LI R3,2 MAXIMUM NUMBER OF WAVEFORMS TO BE ACQUIRED 02143
3003 190C 04C4 CLR R4 SET INTERNALLY CLOCKED ACQUIRE MODE 02144
3004 190E 1033 JMP CHKSTOP 02145
3005 1910 0201 000A KEYAVG10 LI R1,10 USERSTACK GETS 10 02146
3006 1914 1015 JMP PSHANUM 02147
3007 1916 0201 0064 KEYAVG100 LI R1,100 USERSTACK GETS 100 02148
3008 191A 1012 JMP PSHANUM 02149
3009 191C 0201 03E8 KEYAVG1000 LI R1,1000 USERSTACK GETS 1000 02150
3010 1920 100F JMP PSHANUM 02151
3011 1922 C320 D966 KEYAVG MOV KEY,R12 IS THIS A 'STOP' 02152
3012 1926 2320 3382 COC STOPKEY,R12 YES, STOP 'AVG' 02153
3013 192A 132A JEQ AVGSTOP 02154
3014 192C 06A0 690E BL POPREG POP AVERAGE COUNT OFF USER STACK 02155
3015 1930 C000 MOV R0,R0 CHECK IF VALUE IS A FLOATING POINT NUMBER 02156
3016 1932 160E JNE AVGERR ERROR IF WFM # 02157
3017 1934 06A0 75A0 BL FP2INT CONVERT FLOATING POINT NUMBER TO INTEGER 02158
3018 1938 1308 JEQ AVGERR 02159
3019 193A 0281 03FF CI R1,1023 NUMBER MUST BE IN RANGE 1<=X<=1023 02160
3020 193E 1808 JH AVGERR 02161
3021 1940 0649 PSHANUM DECT SOFT PUSH AVERAGE NUMBER ONTO SOFTSTACK 02162
3022 1942 C641 MOV R1,*SOFT 02163
3023 1944 0200 1D00 LI R0,AVGWFM 'AVE' KEY WAVEFORM ACQUISITION ROUTINE 02164
3024 1948 0203 0001 LI R3,1 MAXIMUM NUMBER OF WAVEFORMS TO BE ACQUIRED 02165
3025 194C 04C4 CLR R4 SET INTERNALLY CLOCKED ACQUIRE MODE 02166
3026 194E 1013 JMP CHKSTOP 02167
3027 1950 04E0 D94A AVGERR CLR FATAL SET ERROR FLAG 02168
3028 1954 C020 3304 MOV WSTK,RJ PUSH VALUE BACK ONTO USER STACK 02169
3029 1958 06A0 6936 BL PSHSTK 02170
3030 195C 0380 RTWP 02171
3031 195E 0200 1E8A KEYAQS LI R0,AQSWFM 'AQS' KEY WAVEFORM ACQUISITION ROUTINE 02172
3032 1962 0203 0002 LI R3,2 MAXIMUM NUMBER OF WAVEFORMS TO BE ACQUIRED 02173
3033 1966 C120 3380 MOV MWEXCLK,R4 SET EXTERNALLY CLOCKED ACQUIRE MODE 02174
3034 196A 1005 JMP CHKSTOP 02175
3035 196C 0200 1C52 KEYGND LI R0,GNDWFM 'GND' KEY WAVEFORM ACQUISITION ROUTINE 02176
3036 1970 0203 0002 LI R3,2 MAXIMUM NUMBER OF WAVEFORMS TO BE ACQUIRED 02177
3037 1974 04C4 CLR R4 SET INTERNALLY CLOCKED ACQUIRE MODE 02178
3038 1976 C320 D966 CHKSTOP MOV KEY,R12 IS THIS ACTUALLY A 'STOP' COMMAND? 02179
3039 197A 2320 3382 COC STOPKEY,R12 YES, STOP EXECUTION OF KEY 02180
3040 197E 1607 JNE CNTWFMS NO, THEN THIS IS START OF KEY 02181
3041 1980 04E0 D94C AVGSTOP CLR WARNING SET WARNING WHEN 'STOP'ED 02182
3042 1984 0460 189E B STOPACQ RESTORE VARIABLES 02183
3043 1988 04E0 D94A ACQERR CLR FATAL SET ERROR FLAG FOR USER INDICATION 02184
3044 198C 0380 RTWP 02185

```


WAVEFORM ACQUISITION KEYS

02141

3045	198E	06A0	1F7A	CNTWFMS	BL	CNTWFM	COUNT REALTIME WAVEFORMS	02186
3046	1992	13FA			JEQ	ACQERR	ZERO WAVEFORMS IS INVALID STATE	02187
3047	1994	80D9			C	*SOFT,R3	IS A VALID NUMBER OF WAVEFORMS DISPLAYED?	02188
3048	1996	1BF8			JH	ACQERR	NO, THIS IS AN ERROR	02189
3049	1998	C0D9			MOV	*SOFT,R3	GET NUMBER OF REALTIME WAVEFORMS	02190
3050	199A	C083			MOV	R3,R2	SAVE THIS VALUE FOR FUTURE USE	02191
3051	199C	E120	3382		SOC	MWACQR,R4	SET ACQUIRE MODE BIT	02192
3052	19A0	0283	0002		CI	R3,2	IS THIS A 2 WAVEFORM ACQUIRE?	02193
3053	19A4	1102			JLT	*+6	NO, THEN LEAVE BIT ALONE	02194
3054	19A6	E120	337E		SOC	MW2WFM,R4	YES, SET BIT INDICATING 2 WAVEFORM ACQUIRE	02195
3055	19AA	D160	D9CF		MOVB	DSPWFM+1,R5	GET LOWER BYTE OF DISPLAY WAVEFORM MODE WORD	02196
3056	19AE	0245	0700		ANDI	R5,\$0700	KEEP RESOLUTION BITS ONLY	02197
3057	19B2	E105			SOC	R5,R4	SET RESOLUTION BITS	02198
3058	19B4	0300	0007		LIMI	7	MASK OFF SYSTEM INTERRUPTS	02199
3059	19B8	C320	3386		MOV	DSPSTP,R12	DISPLAY STOP CRU LINE	02200
3060	19BC	1E00			SBZ	0	STOP DISPLAY	02201
3061	19BE	C820	D9C8	E000	MOV	DSPRLT,DMDWRD	SET DISPLAY MODE BITS TO STOP	02202
3062	19C4	1D00			SBZ	0	ENABLE A WORD COUNTER	02203
3063	19C6	C320	3382		MOV	DSPRST,R12	DISPLAY DONE INTERRUPT RESET CRU LINE	02204
3064	19CA	1E00			SBZ	0	RESET DISPLAY DONE INTERRUPT	02205
3065	19CC	04E0	D9C4		CLR	REFRESH	FLAG PERMITTING DISPLAY TO START AGAIN	02206
3066	19D0	0420	1162		BLWP	ROAQR	ACQUIRE REALTIME READOUT INTO TEXT BUFFER	02207
3067	19D4	06A0	1FE2		BL	CHKRD	GET SEARCH PATTERNS FOR REALTIME SCALE FACTOR	02208
3068	19D8	C159			MOV	*SOFT,R5	GET SEARCH PATTERN FOR HORIZONTAL READOUTS	02209
3069	19DA	0A35			SLA	R5,3		02210
3070	19DC	0245	8080		ANDI	R5,\$8000	KEEP BIT INDICATING TIMEBASE: 0=A, 1=B	02211
3071	19E0	E144			SOC	R4,R5	ADD PREVIOUS ACQUIRE MODE INFORMATION	02212
3072	19E2	C805	D900		MOV	R5,ACQWFM	MOVE THIS TO ACQUIRE MODE WORD	02213
3073	19E6	0283	0002		CI	R3,2	ARE 2 WAVEFORMS TO BE ACQUIRED?	02214
3074	19EA	1105			JLT	GHRZSF	NO, THEN USE THIS MODE WORD AGAIN	02215
3075	19EC	C159			MOV	*SOFT,R5	YES, A MODE WORD IS NEEDED FOR BOTH WFMS	02216
3076	19EE	0A85			SLA	R5,11		02217
3077	19F0	0245	8000		ANDI	R5,\$8000	KEEP BIT INDICATING SECOND WFMS' TIMEBASE	02218
3078	19F4	E144			SOC	R4,R5	ADD PREVIOUS ACQUIRE MODE INFORMATION	02219
3079	19F6	C805	D902	GHRZSF	MOV	R5,ACQWFM1	SETUP SECOND WAVEFORMS ACQUIRE MODE	02220
3080	19FA	C150	D9C2		MOV	FPTMP,R5	GET CURRENT FRONT PANEL BIT SETTINGS	02221
3081	19FE	0245	01FF		ANDI	R5,\$01FF	KEEP MODE BUTTON INFO ONLY	02222
3082	1A02	C1A0	0A25		MOV	VERTM,R6	GET CURRENT VERTICAL MODE	02223
3083	1A06	8806	334E		C	R6,VERTADD	IS MAINFRAME IN 'ADD'?	02224
3084	1A0A	1602			JNE	*+6	NO, THEN OK	02225
3085	1A0C	04E0	D94C		CLR	WARNING	YES, THIS IS A WARNING	02226
3086	1A10	F166	233E		SOCB	FPACQ(R6),R5	SET CHANNEL SWITCH BITS FOR THIS MODE	02227
3087	1A14	C1A0	0A28		MOV	HORZM,R6	GET CURRENT HORIZONTAL MODE	02228
3088	1A18	F166	233E		SOCB	FPACQ(R6),R5	SET CHANNEL SWITCH BITS FOR THIS MODE	02229
3089	1A1C	0283	0002		CI	R3,2	IF 2 WAVEFORMS ARE TO BE ACQUIRED THEN	02230
3090	1A20	1302			JEQ	*+6	SET PLUG-IN MODE TO DIGITIZE/2 ELSE	02231
3091	1A22	0225	0600		AI	R5,\$0600	SET PLUG-IN MODE TO REALTIME	02232
3092	1A26	C805	E010		MOV	R5,FP	SETUP CHANNEL SWITCHES FOR ACQUIRE	02233
3093	1A2A	0280	1C52		CI	R0,GNDWFM	DON'T ACQUIRE READOUTS FOR 'GND' KEY	02234
3094	1A2E	137C			JEQ	SAVSTST		02235
3095	1A30	06A0	2048		BL	CHKSF	DETERMINE WHICH READOUTS ARE SCALE FACTORS	02236
3096	1A34	0649			DECT	SOFT		02237
3097	1A36	C643			MOV	R3,*SOFT	PUT NUMBER OF WAVEFORMS ON SOFTSTACK	02238
3098	1A38	06A0	213A		BL	WHCHSF	DETERMINE WHICH SCALE FACTORS TO USE	02239
3099	1A3C	04C1			CLR	R1	CREATE WAVEFORMS NUMBER	02240
3100				*				
3101				*		PROCESS HORIZONTAL SCALE FACTOR		
3102				*				
3103	1A3E	0420	20AE	HORZSF	BLWP	RDIN	GET SCALE FACTOR AS FLOATING POINT NUMBER	02241

WAVEFORM ACQUISITION KEYS

02141

3104	1A42	0280	1E8A		CI	R0,AQSWFM	IS THIS THE KEY 'AQS'?	02242
3105	1A46	1642			JNE	NOTAQS	YES, FURTHER CHECKING IS NEEDED	02243
3106	1A48	8820	DA28	3352	C	HORZM,HORZB	HORIZONTAL MODE MUST BE B	02244
3107	1A4E	1627			JNE	AQSERR		02245
3108	1A50	9820	DF83	370E	CB	TS10.7B87,DD	IF TIMESLOT 10 OF 7B87'S READOUT IS A 8DD	02246
3109	1A56	130E			JEQ	CHKRATE	THEN NORMAL 7B87 MODE	02247
3110	1A58	9820	DF8B	370D	CB	TS10.7B87,CD	IF TIMESLOT 10 OF 7B87'S READOUT IS A 8CD	02248
3111	1A5E	1628			JNE	CHKEXT	THEN /1000 MODE	02249
3112	1A60	C139			MOV	*SOFT+,R4	SAVE HORIZONTAL UNITS	02250
3113	1A62	C1C1			MOV	R1,R7	SAVE R1	02251
3114	1A64	C282			MOV	R2,R8	SAVE R2	02252
3115	1A66	C060	3400		MOV	FP10004,R1	MANTISSA FOR FLOATING POINT 1000	02253
3116	1A6A	C0A0	3402		MOV	FP1000E,R2	EXPONENT FOR FLOATING POINT 1000	02254
3117	1A6E	0420	70AA		BLWP	FPMPY	ADJUST SCALE FACTOR FOR /1000 MODE	02255
3118	1A72	1003			JMP	CHKRAT		02256
3119	1A74	C139			CHKRATE MOV	*SOFT+,R4	SAVE HORIZONTAL UNITS	02257
3120	1A76	C1C1			MOV	R1,R7	SAVE R1	02258
3121	1A78	C202			MOV	R2,R8	SAVE R2	02259
3122	1A7A	C060	3412		CHKRAT MOV	MAXRATM,R1	MANTISSA FOR MAXIMUM SWEEP RATE / 10	02260
3123	1A7E	C0A0	3414		MOV	MAXRATE,R2	EXPONENT FOR MAXIMUM SWEEP RATE / 10	02261
3124	1A82	C2E0	D970		MOV	RESOLV,R11	CURRENT WAVEFORM RESOLUTION	02262
3125	1A86	091B			SRL	R11,1		02263
3126	1A88	0582			INC	R2	MAXIMUM SWEEP RATE / 10 * R	02264
3127	1A8A	091B			SRL	R11,1		02265
3128	1A8C	16FD			JNE	*-4		02266
3129	1A8E	0420	7524		BLWP	FPCMPR	COMPARE SWEEP RATE TO MAXIMUM	02267
3130	1A92	1505			JGT	AQSERR	IF LARGER THEN ERROR	02268
3131	1A94	C047			MOV	R7,R1	RESTORE R1	02269
3132	1A96	C088			MOV	R8,R2	RESTORE R2	02270
3133	1A98	0649			DECT	SOFT		02271
3134	1A9A	C644			MOV	R4,*SOFT	RESTORE HORIZONTAL UNITS	02272
3135	1A9C	1017			JMP	NOTAQS		02273
3136	1A9E	04E0	D9D0		AQSERR CLR	ACQWFM	CLEAR ACQUIRE MODE WORDS	02274
3137	1AA2	04E0	D9D2		CLR	ACQWFM1		02275
3138	1AA6	0720	D972		SETJ	RDTFLAG	UPDATE ENTIRE READOUT	02276
3139	1AAA	04E0	D94A		CLR	FATAL	FLAG THIS AS AN ERROR	02277
3140	1AAE	0380			RTWP			02278
3141	1AB0	9820	DF8B	370C	CHKEXT CB	TS10.7B87,3D	IF TIMESLOT 10 OF 7B87'S READOUT IS A 83D	02279
3142	1AB6	16F3			JNE	AQSERR	THE EXTERNAL CLOCK MODE	02280
3143	1AB8	CE60	3430		MOV	SPACES,*SOFT+	DEFAULT UNITS ARE ' '	02281
3144	1ABC	CE60	3404		MOV	FP1M,*SOFT+	DEFAULT SCALE FACTOR IS 1	02282
3145	1AC0	C660	333A		MOV	FP1E,*SOFT		02283
3146	1AC4	0229	FFFC		AI	SOFT,-4		02284
3147	1AC8	04E0	D94C		CLR	WARNING	FLAG THIS AS A WARNING	02285
3148	1ACC	06A0	6984		NOTAQS BL	ADRWFM	GET HEADER AND DATA ADDRESS FOR WAVEFORM	02286
3149	1AD0	C119			MOV	*SOFT,R4	HEADER ADDRESS	02287
3150	1AD2	A120	3348		A	VOFFA3,R4	POINT TO VERTICAL OFFSET IN HEADER	02288
3151	1AD6	0A11			SLA	R1,1	CREATE INDEX	02289
3152	1AD8	C521	DA2A		MOV	GROUND0(R1),*R4	MOVE VERTICAL ZERO TO HEADER	02290
3153	1ADC	0911			SRL	R1,1	RESTORE BACK TO WFM NUMBER	02291
3154	1ADE	C139			MOV	*SOFT+,R4	HEADER ADDRESS	02292
3155	1AE0	05C9			INCT	SOFT	DATA ADDRESS	02293
3156	1AE2	A120	3350		A	HSCALE,R4	POINT TO HORIZONTAL SCALE UNITS IN HEADER	02294
3157	1AE6	C519			MOV	*SOFT,*R4	MOVE NEW UNITS TO HORIZONTAL SCALE IN HEADER	02295
3158	1AE8	C641			MOV	R1,*SOFT	PUT WAVEFORM NUMBER ON SOFTSTACK	02296
3159	1AEA	0420	6EA8		BLWP	NEWHSC	PUT SCALE FACTOR INTO HEADER	02297
3160	1AEE	0581			INC	R1	INCREMENT WAVEFORM NUMBER	02298
3161	1AF0	80C1			C	R1,R3	ARE THERE MORE WAVEFORMS?	02299
3162	1AF2	11A5			JLT	HORZSF	YES, JJ THEN ALL	02300

WAVEFORM ACQUISITION KEYS

02141

3163	1AF4	06A0	2048		BL	CHKSF	DETERMINE WHICH READOUTS ARE SCALE FACTORS	02301
3164	1AF8	0649			DECT	SOFT		02302
3165	1AFA	C643			MOV	R3,*SOFT	PUT NUMBER OF WAVEFORMS ON SOFTSTACK	02303
3166	1AFC	06A0	213A		BL	WHGHSF	DETERMINE WHICH SCALE FACTORS TO USE	02304
3167	1B00	04C1			CLR	R1	CREATE WAVEFORM NUMBER	02305
3168	1B02	0206	D97E		LI	R6,DSP3UF	DISPLAY WAVEFORM BUFFER	02306
3169	1B06	0420	20AE	VERTSF	BLWP	RDIN	GET SCALE FACTOR AS FLOATING POINT NUMBER	02307
3170	1B0A	06A0	6984		BL	ADRFWM	GET HEADER AND DATA ADDRESS FOR WAVEFORM	02308
3171	1B0E	C139			MOV	*SOFT+,R4	HEADER ADDRESS	02309
3172	1B10	C0B9			MOV	*SOFT+,*R6+	DATA ADDRESS TO DISPLAY BUFFER	02310
3173	1B12	A120	334C		A	VSGALD,R4	POINT TO VERTICAL SCALE UNITS IN HEADER	02311
3174	1B16	C519			MOV	*SOFT,*R4	MOVE NEW UNITS TO VERTICAL SCALE IN HEADER	02312
3175	1B18	C641			MOV	R1,*SOFT	PUT WAVEFORM NUMBER ON SOFTSTACK	02313
3176	1B1A	0420	6EE0		BLWP	NEWVSL	PUT VERTICAL SCALE FACTOR INTO HEADER	02314
3177	1B1E	0581			INC	R1	INCREMENT WAVEFORM NUMBER	02315
3178	1B20	80C1			C	R1,R3		02316
3179	1B22	11F1			JLT	VERTSF		02317
3180	1B24	04D6			CLR	*R6	ENDFLAG FOR DISPLAY WAVEFORM BUFFER	02318
3181	1B26	1002			JMP	*+6		02319
3182	1B28	0229	0004	SAVSTST	AI	SOFT,4	POP READOUT SEARCH PATTERNS IF 'GND'	02320
3183	1B2C	C1A0	D9CE		MOV	DSPWFM,R6	SAVE CURRENT WAVEFORM DISPLAY STATUS	02321
3184	1B30	C150	D954		MOV	CURSOR,R5	SAVE CURRENT CURSOR DISPLAY STATUS	02322
3185	1B34	4820	3362	D9CE	SZC	MWVSY,DSPWFM	SELECT Y-T WAVEFORM DISPLAY	02323
3186	1B3A	4820	3348	D9CE	SZC	MWVCTR,DSPWFM	SELECT DOT WAVEFORM DISPLAY	02324
3187	1B40	04E0	D954		CLR	CURSOR	TURN OFF CURSORS	02325
3188	1B44	0720	D972		SETJ	RDIFLAG		02326
3189	1B48	4820	333C	D972	SZC	CLINE2,RDIFLAG	DON'T UPDATE LINE #2	02327
3190	1B4E	0649			DECT	SOFT		02328
3191	1B50	C660	3358		MOV	C16,*SOFT	CLEAR LINE 16 OF DISPLAY	02329
3192	1B54	0649			DECT	SOFT		02330
3193	1B56	C660	333A		MOV	C1,*SOFT		02331
3194	1B5A	06A0	12FA		BL	NULLIN	CLEAR LINE 15 OF REALTIME READOUT	02332
3195	1B5E	0229	0004		AI	SOFT,4	POP TEXT POINTER OFF SOFTSTACK	02333
3196	1B62	0420	1430		BLWP	FRDOUT	UPDATE CALCULATOR DISPLAY IF SELECTED	02334
3197	1B66	0280	1C52		CI	R0,GNDWFM	DON'T NULL WAVEFORMS FOR 'GND' KEY	02335
3198	1B6A	1313			JEQ	SETSTOP		02336
3199	1B6C	C843			MOV	R3,R1		02337
3200	1B6E	0601			DEC	R1	CREATE WAVEFORM NUMBER	02338
3201	1B70	0280	1000		CI	R0,AVGWFM	FOR 'AVE' ALSO NULL WAVEFORM #1	02339
3202	1B74	1601			JNE	ACQNULL		02340
3203	1B76	0581			INC	R1		02341
3204	1B78	06A0	6984	ACQNJLL	BL	ADRFWM	GET HEADER AND DATA ADDRESS OF WAVEFORM	02342
3205	1B7C	05C9			INCT	SOFT	HEADER ADDRESS	02343
3206	1B7E	C1F9			MOV	*SOFT+,R7	DATA ADDRESS	02344
3207	1B80	C220	D978		MOV	RESOLV,R8	NUMBER OF POINTS TO SET TO NULL (\$8000)	02345
3208	1B84	CDE0	339J		MOV	BADPNT,*R7+	SET WAVEFORM POINT TO NULL (\$8000)	02346
3209	1B88	0608			DEC	R8	NULL ENTIRE WAVEFORM	02347
3210	1B8A	15FC			JGT	*-6		02348
3211	1B8C	0601			DEC	R1	IS THERE ANOTHER WAVEFORM TO NULL?	02349
3212	1B8E	15F4			JGT	ACQNULL	YES, NULL IT ALSO	02350
3213	1B90	13F3			JEQ	ACQNULL		02351
3214	1B92	0300	0007	SETSTOP	LIMI	7	MASK OFF SYSTEM INTERRUPTS	02352
3215	1B96	E820	3382	D966	SOC	STOPKEY,KEY	ALLOW USER TO 'STOP' THIS KEY	02353
3216	1B9C	0410			BLWP	*R8	PROCESS REQUIRED ACQUISITION	02354
3217	1B9E	0300	0007	STOPACQ	LIMI	7	MASK OFF ALL SYSTEM INTERRUPTS	02355
3218	1BA2	04E0	E00A		CLR	AMDWRD	STOP WAVEFORM ACQUISITION	02356
3219	1BA6	04E0	D9D0		CLR	ACQWFM	CLEAR ACQUIRE MODE WORDS	02357
3220	1BAA	04E0	D9D2		CLR	ACQWFM1		02358
3221	1BAE	C820	D9C2	E018	MOV	FPTEMP,FP	RESET CHANNEL SWITCHES TO DISPLAY MODE	02359

3222	18B4	4820	3382	D966	SZC	STOPKEY,KEY	KEY CANNOT BE 'STOP'ED NOW	02360
3223	18BA	0720	0972		SETD	RDTFLAG	UPDATE CALCULATOR READOUT DISPLAY	02361
3224	18BE	C805	0954		MOV	R5,CURSOR	RESTORE CURSOR DISPLAY STATUS	02362
3225	18C2	C806	09CE		MOV	R6,DSPWFM	RESTORE WAVEFORM DISPLAY STATUS	02363
3226	18C6	04C1			CLR	R1	CREATE WAVEFORM ADDRESS INDEX	02364
3227	18C8	C0E1	097E	AQRFILL	MOV	DSPBJF(R1),R3	ADDRESS OF NEXT WAVEFORM	02365
3228	18CC	132B			JEQ	PSHWFMS	ADDRESS OF \$8000 ENDS LOOP	02366
3229	18CE	0649			DECT	SOFT		02367
3230	18D0	C643			MOV	R3,*SOFT	PUSH WAVEFORM ADDRESS ONTO SOFTSTACK	02368
3231	18D2	0420	219C		BLWP	FILLCHK	CHECK THE FILL OF THE WAVEFORM	02369
3232	18D6	C339			MOV	*SOFT+,R12	GET VALID POINT COUNT	02370
3233	18D8	05C9			INCT	SOFT	PDP GAP SIZE	02371
3234	18DA	C1E0	0970		MOV	RESOLV,R7	GET CURRENT RESOLUTION	02372
3235	18DE	39E0	3436		MPY	WARNPER,R7	CALCULATE WARNING PERCENTAGE	02373
3236	18E2	30E0	3374		DIV	C100,R7		02374
3237	18E6	81CC			C	R12,R7	IF THIS PERCENTAGE WASN'T REACHED THIS IS	02375
3238	18E8	1502			JGT	*+6	A WARNING CONDITION	02376
3239	18EA	04E0	094C		CLR	WARNING		02377
3240	18EE	0280	1E8A		CI	R0,AQSWFM	END POINTS ARE GOOD WITH 'AQS'	02378
3241	18F2	1309			JEQ	ENDAQS		02379
3242	18F4	C4E0	3390		MOV	BADPNT,*R3	SET FIRST POINT TO NULL (\$8000)	02380
3243	18F8	A0E0	0970		A	RESOLV,R3		02381
3244	18FC	A0E0	0970		A	RESOLV,R3		02382
3245	1C00	C643			DECT	R3		02383
3246	1C02	C4E0	3390		MOV	BADPNT,*R3	SET LAST POINT TO NULL (\$8000)	02384
3247	1C06	0280	1C52	ENDAQS	CI	R0,GNDWFM	IS THIS THE 'GND' KEY?	02385
3248	1C0A	1305			JEQ	FILLWFM	YES, LEAVE VALUES UNDEFINED	02386
3249	1C0C	C30C			MOV	R12,R12	WERE ANY POINTS ACQUIRED?	02387
3250	1C0E	1503			JGT	*+8	YES, OK	02388
3251	1C10	C4E1	DA2A		MOV	GROUND0(R1),*R3	NO, DEFAULT WAVEFORM TO VZR VALJE	02389
3252	1C14	0513			NEG	*R3		02390
3253	1C16	0649		FILLWFM	DECT	SOFT		02391
3254	1C18	C661	097E		MOV	DSPBJF(R1),*SOFT	PUT WFM ADDRESS ONTO SOFTSTACK	02392
3255	1C1C	0420	68EE		BLWP	PNTFILL	INTERPOLATE ALL UNFILLED POINTS	02393
3256	1C20	05C1			INCT	R1		02394
3257	1C22	10D2			JMP	AQRFILL	FILL ALL WAVEFORMS ACQUIRED	02395
3258	1C24	0280	1C52	PSHWFMS	CI	R0,GNDWFM	DON'T PUSH WFM NUMBERS IF 'GND' KEY	02396
3259	1C28	1311			JEQ	ENDGND		02397
3260	1C2A	0280	10D0		CI	R0,AVGWFM		02398
3261	1C2E	1605			JNE	PSHNUMS		02399
3262	1C30	0649			DECT	SOFT		02400
3263	1C32	C650	333A		MOV	C1,*SOFT		02401
3264	1C36	06A0	6F26		BL	NULLWFM	INITIALIZE WFM #1	02402
3265	1C3A	C042		PSHNUMS	MOV	R2,R1	LOAD NUMBER OF WAVEFORMS ACQUIRED	02403
3266	1C3C	0700			SETD	R0	FLAG INDICATING WAVEFORM ON USER STACK	02404
3267	1C3E	0601			DEC	R1	CREATE WAVEFORM NUMBER	02405
3268	1C40	06A0	6962		BL	PSHREG	PUSH WAVEFORM NUMBER ONTO USER STACK	02406
3269	1C44	0601			DEC	R1	PUSH ALL WAVEFORM NUMBERS	02407
3270	1C46	15FC			JGT	*-6		02408
3271	1C48	13FB			JEQ	*-8		02409
3272	1C4A	0380			RTWP			02410
3273								
3274	1C4C	0420	1048	ENDGND	BLWP	GNDSTOP	TERMINATE 'GND' DIFFERENTLY	02411
3275	1C50	0380			RTWP			02412

'AQR' & 'GND' WAVEFORM ACQUISITION

02413

```

3277 *****
3278 **
3279 ** THE GROUND WAVEFORMS ARE ACQUIRED INTO A SEPARATE MEMORY **
3280 ** SPACE FROM THE OTHER WAVEFORMS. THE GROUND KEY CANNOT **
3281 ** DESTROY ANY WAVEFORMS. REGARDLESS OF THE P/W THE 'GND' **
3282 ** KEY ACQUIRES WAVEFORMS AT 120 P/W. THESE 120 ARE AVERAGED **
3283 ** TO PRODUCE TO GROUND VALUE. **
3284 **
3285 *****
3286 1C52 DB80 GND1WFM WORD WPLV_2 LEVEL 2 ROUTINE 02414
3287 1C54 1C56 WORD *+2 02415
3288 1C56 C260 0012 MOV 10(R13),SOFT GET SOFTSTACK POINTER 02416
3289 1C5A 04E0 0980 CLR DSPBUF+2 END WAVEFORM DISPLAY BUFFER FLAG 02417
3290 1C5E 04E0 0982 CLR DSPBUF+4 END WAVEFORM DISPLAY BUFFER FLAG 02418
3291 1C62 8819 333A C *SOFT,C1 CHECK NUMBER OF WAVEFORMS TO ACQUIRE 02419
3292 1C66 130A JEQ GND1WFM 02420
3293 1C68 C060 33E2 GND2WFM MOV GND1BAS,R1 IF 2 WAVEFORMS NULL AND DISPLAY BOTH 02421
3294 1C6C C801 0980 MOV R1,DSPBUF MOVE ADDRESS TO DISPLAY WAVEFORM BUFFER 02422
3295 1C70 0202 0080 LI R2,120 DEFAULT OF 120 POINTS/WAVEFORM 02423
3296 1C74 CC60 339J MOV BADPNT,*R1+ NULL WAVEFORM 02424
3297 1C78 0602 DEC R2 02425
3298 1C7A 15FC JGT *-6 02426
3299 1C7C C060 33E0 GND1WFM MOV GND0BAS,R1 NULL AND DISPLAY WAVEFORM 02427
3300 1C80 C801 097E MOV R1,DSPBUF MOVE ADDRESS TO DISPLAY WAVEFORM BUFFER 02428
3301 1C84 0202 0080 LI R2,120 DEFAULT OF 120 POINTS/WAVEFORM 02429
3302 1C88 CC60 3390 MOV BADPNT,*R1+ NULL WAVEFORM 02430
3303 1C8C 0602 DEC R2 02431
3304 1C8E 15FC JGT *-6 02432
3305 1C90 0201 0700 LI R1,$0700 FORCE RESOLUTION OF 120 POINTS/WAVEFORM 02433
3306 1C94 E801 0900 SOC R1,ACQWFM 120 POINTS/WAVEFORM FOR ACQUISITION 02434
3307 1C98 E801 0902 SOC R1,ACQWFM1 02435
3308 1C9C E820 3346 D9CE SOC C7,DSPWFM 120 POINTS/WAVEFORM FOR DISPLAY 02436
3309 1CA2 C160 097A MOV YTWFM,R5 SAVE CURRENT Y-T WAVEFORM X PRELOAD 02437
3310 1CA6 C820 3398 D97A MOV CHE008,YTWFM FORCE TO 120 RESOLUTION 02438
3311 1CAC C1A0 097C MOV YTWFM+2,R6 02439
3312 1CB0 C820 33A0 D97C MOV CHE088,YTWFM+2 02440
3313 1CB6 C120 0970 MOV RESOLV,R4 SAVE CURRENT RESOLUTION 02441
3314 1CBA C820 3376 D970 MOV C128,RESOLV FORCE NEW RESOLUTION TO 120 POINTS/WAVEFORM 02442
3315 1CC0 1004 JMP AQR.GND JUMP TO COMMON SECTION 02443
3316
3317 1CC2 DB80 AQRWFM WORD WPLVL2 LEVEL 2 ROUTINE 02444
3318 1CC4 1CC5 WORD *+2 02445
3319 1CC6 C260 0012 MOV 10(R13),SOFT GET SOFTSTACK POINTER 02446
3320 1CCA 0201 097E AQR.GND LI R1,DSPBUF ADDRESSES OF WAVEFORM BUFFERS 02447
3321 1CCE 8819 333A C *SOFT,C1 02448
3322 1CD2 1302 JEQ AQR1WFM 02449
3323 1CD4 C831 E002 AQR2WFM MOV *R1+,AWRD WD'S ADDRESS TO A-WORD (2 WFM AQR ONLY) 02450
3324 1CD8 C831 E004 AQR1WFM MOV *R1+,BWRD W1'S ADDRESS TO B-WORD (WD'S IF 1 WFM AQR) 02451
3325 1CDC C820 0900 E00A MOV ACQWFM,AMDWRD START ACQUISITION HARDWARE 02452
3326 1CE2 C0D9 MOV *SOFT,R3 NUMBER OF WAVEFORM TO ACQUIRE 02453
3327 1CE4 C820 333A D9B2 AQRSWPS MOV C1,SWEEPS SET SWEEPS COUNT TO 1 02454
3328 1CEA 0300 000F AQRSWP? LIM? $F ENABLE ALL SYSTEM INTERRUPTS 02455
3329 1CEE 0720 DA16 SETJ FLICKER TELL DISPLAY THAT IT IS FLICKERING 02456
3330 1CF2 C0A0 09B2 MOV SWEEPS,R2 WAIT UNTIL COMPLETE SWEEP IS ACQUIRED 02457
3331 1CF6 15F9 JGT AQRSWP 02458
3332 1CF8 0603 DEC R3 ACQUIRE 1 COMPLETE SWEEP FOR EACH WAVEFORM 02459
3333 1CFA 15F4 JGT AQRSWPS 02460
3334 1CFC 04C1 CLR R1 CREATE WAVEFORM INDEX 02461
3335 1CFE 0720 DA16 AQRMORE SETJ FLICKER TELL DISPLAY THAT IT IS FLICKERING 02462

```

'AQR' & 'GND' WAVEFORM ACQUISITION

02413

3336	1002	0649		DECT	SOFT		02463	
3337	1004	C661	D97E	AQR1.8	MOV	DSPBJF(R1),*SOFT WFM ADDRESS ON SOFTSTACK FOR 'FILLCHK'	02464	
3338	1008	1601		JNE	**4	IF ADDRESS IS \$0000 THEN NO MORE WAVEFORMS	02465	
3339	100A	0380		RTWP			02466	
3340	100C	0420	219C	BLWP	FILLCHK	GET FILL COUNT AND LARGEST GAP SIZE	02467	
3341	1010	0720	0A16	SETJ	FLICKER	TELL DISPLAY THAT IT IS FLICKERING	02468	
3342	1014	C339		MOV	*SOFT+,R12	NUMBER OF FILLED POINTS	02469	
3343	1016	C1E0	D970	MOV	RESOLV,R7	NUMBER OF TOTAL POINTS IN WAVEFORM	02470	
3344	101A	0937		SRL	R7,3	1/8 OF RESOLUTION IS MINIMUM POINTS ALLOWED	02471	
3345	101C	81CC		C	R12,R7		02472	
3346	101E	11F2		JLT	AQR1.8		02473	
3347	1020	C1E0	D970	MOV	RESOLV,R7	NUMBER OF TOTAL POINTS IN WAVEFORM	02474	
3348	1024	39E0	3434	MPY	PERCENT,R7	NEEDED PERCENTAGE FOR COMPLETE FILL	02475	
3349	1028	3DE0	3374	DIV	C100,R7		02476	
3350	102C	81CC		C	R12,R7	HAS THIS PERCENTAGE BEEN REACHED?	02477	
3351	102E	150A		JGT	AQRFJL	YES, CHECK IF ANOTHER WFM TO CHECK	02478	
3352	1030	1309		JEQ	AQRFJLL		02479	
3353	1032	C1E0	D970	MOV	RESOLV,R7	NUMBER OF TOTAL POINTS IN WAVEFORM	02480	
3354	1036	51F9		S	*SOFT+,R7	SUBTRACT LARGEST GAP FROM TOTAL	02481	
3355	1038	39E0	3434	MPY	PERCENT,R7	GET PERCENTAGE NEEDED WITH GAP	02482	
3356	103C	3DE0	3374	DIV	C100,R7		02483	
3357	1040	81CC		C	R12,R7	IS WAVEFORM FULL BESIDES GAP?	02484	
3358	1042	110D		JLT	AQRMORE	NO, KEEP ACQUIRING	02485	
3359	1044	05C1		AQRFULL	INCT	R1	IS THERE ANOTHER WFM TO CHECK?	02486
3360	1046	100B		JMP	AQRMORE	YES, CHECK IT TOO	02487	
3361								
3362	1048		0B80	GNDSTOP	WORD	WPLVL2	LEVEL 2 ROUTINE	02488
3363	104A		104C		WORD	**2		02489
3364	104C	C26D	0012	MOV	18(R13),SOFT	GET CURRENT SOFTSTACK POINTER	02490	
3365	1050	C804	D970	MOV	R4,RESOLV	RESTORE RESOLUTION	02491	
3366	1054	C805	D97A	MOV	R5,YTWFH	RESTORE X PRELOAD	02492	
3367	1058	C806	D97C	MOV	R6,YTWFH+2		02493	
3368	105C	C0D9		MOV	*SOFT,R3	GET NUMBER OF WAVEFORMS ACQUIRED	02494	
3369	105E	0603		DEC	R3		02495	
3370	1060	0A13		SLA	R3,1	CREATE INDEX FROM WAVEFORM NUMBERS	02496	
3371	1062	C223	33E0	NXTGND	MOV	GND03AS(R3),R8	GET ADDRESS OF GROUND WAVEFORM	02497
3372	1066	0207	0080	LI	R7,128	DEFAULT RESOLUTION OF 128 POINTS	02498	
3373	106A	04C4		CLR	R4	INITIAL VALUE IS 0	02499	
3374	106C	04C5		CLR	R5		02500	
3375	106E	C198		GNDAVE	MOV	*R8,R6	GET NEXT POINT FROM WAVEFORM	02501
3376	1070	08F6		SRA	R6,15	SIGN EXTEND TO CREATE DOUBLE-WORD VALUE	02502	
3377	1072	A106		A	R6,R4	ADD MOST SIGNIFICANT WORDS	02503	
3378	1074	A178		A	*R8+,R5	ADD LEAST SIGNIFICANT WORDS	02504	
3379	1076	1701		JNC	**4	IF CARRY, ADD TO MOST SIGNIFICANT WORD	02505	
3380	1078	0584		INC	R4		02506	
3381	107A	0607		DEC	R7	MORE POINTS?	02507	
3382	107C	15F8		JGT	GNDAVE	YES, CONTINUE	02508	
3383	107E	04C0		CLR	R0	SET SIGN FLAG TO POSITIVE	02509	
3384				*	MOV	R4,R4	CHECK SIGN OF FLAG	02510 DEL
3385				*	JGT	GNDPOS		02511 DEL
3386				*				00007PATCH
3387				*	PROBLEM #6 - PATCH #7 (1 OF 1)			00007PATCH
3388				*				00007PATCH
3389				*	'GND' DOES NOT CALCULATE SMALL POSITIVE GROUND CORRECTLY			00007PATCH
3390				*				00007PATCH
3391	1080	046D	969A	B	PATCH7	BRANCH TO PATCH #7		00007PATCH
3392			1084	BACK7	EQU	*	DEFINE REENTRY POINT	00007PATCH
3393				*				00007PATCH
3394				*	END OF PROBLEM #7			00007PATCH

'AQR' & 'GND' WAVEFORM ACQUISITION

02413

3395	1084	0700		SETJ	R0	SET SIGN FLAG TO NEGATIVE	02512
3396	1086	0544		INV	R4	CREATE POSITIVE VALUE FOR 'DIV'	02513
3397	1088	0545		INV	R5		02514
3398	108A	0585		INC	R5		02515
3399	108C	1701		JNC	GNDPOS		02516
3400	108E	0584		INC	R4		02517
3401	1090	3020	3376	DIV	C128,R4	CALCULATE AVERAGE	02518
3402	1094	0A15		SLA	R5,1	ROUND DIVISION	02519
3403	1096	8160	3376	C	C128,R5		02520
3404	109A	1501		JGT	**4		02521
3405	109C	0584		INC	R4		02522
3406	109E	C000		MOV	R0,R0	CHECK IF GROUND SHOULD BE NEGATIVE	02523
3407	10A0	1101		JLT	**4	GROUNDS ARE STORED AS NGATIVE VALUES	02524
3408	10A2	0504		NEG	R4		02525
3409	10A4	C8C4	DA2A	MOV	R4,GROUND0 (R3)	SAVE THIS FOR FUTURE ACQUISITIONS	02526
3410	10A8	0504		NEG	R4	CREATE CORRECTLY SIGNED GROUND	02527
3411	10AA	0649		DECT	SOFT		02528
3412	10AC	0409		CLR	*SOFT	PUT GROUND ON STACK	02529
3413	10AE	0649		DECT	SOFT		02530
3414	10B0	C644		MOV	R4,*SOFT		02531
3415	10B2	0201	5000	LI	R1,\$5000	PUT 20 IN R1,R2	02532
3416	10B6	0202	0005	LI	R2,5		02533
3417	10BA	0420	70AA	BLWP	FPMPY	VERTICAL OFFSET = VOFFAB * 20	02534
3418	10BE	04C0		CLR	R0		02535
3419	10C0	C879		MOV	*SOFT+,R1	POP VALUE OFF STACK	02536
3420	10C2	C0B9		MOV	*SOFT+,R2		02537
3421	10C4	06A0	6962	BL	PSHREG	PUSH GROUND ONTO USER STACK	02538
3422	10C8	0643		DECT	R3	ARE THERE MORE GROUND?	02539
3423	10CA	15CB		JGT	NXTGND		02540
3424	10CC	13CA		JEQ	NXTGND		02541
3425	10CE	0380		RTWP			02542

'AVE' WAVEFORM ACQUISITION

02543

```

3427 *****
3428 **
3429 ** AVERAGE A WFM INTO MEMORY **
3430 **
3431 ** LEVEL 2 ROUTINE **
3432 **
3433 *****
3434 1000          DB80      AVGWFM WORD WPLVL2      LEVEL 2 ROUTINE      02544
3435 1002          1004          WORD *+2      02545
3436 1004 C260 0012          MOV 18(R13),SOFT  GET SOFTSTACK POINTER  02546
3437 1008 05C9          AVGWFM INCT SOFT      POP NUMBER OF WAVEFORMS 02547
3438 100A C0F9          MOV *SOFT+,R3     GET AVERAGE COUNT     02548
3439 100C 0201 0001          LI R1,1          02549
3440 10E0 06A0 6984          BL ADRWFM        GET HEADER AND DATA ADDRESS FOR WFM #1 02550
3441 10E4 05C9          INCT SOFT        HEADER ADDRESS        02551
3442 10E6 C099          MOV *SOFT,R2     DATA ADDRESS         02552
3443 10E8 C802 E004          MOV R2,BWRJ     ACQUIRE DATA INTO W1 02553
3444 10EC 05C2          INCT R2         SKIP PAST FIRST POINT 02554
3445 10EE C060 DAD2          MOV WOADJ,R1    DATA ADDRESS OF W0   02555
3446 10F2 05C1          INCT R1         SKIP PAST FIRST POINT 02556
3447 10F4 C120 D970          MOV RESOLV,R4   CURRENT RESOLUTION    02557
3448 10F8 C144          MOV R4,R5       02558
3449 10FA 0645          DECT R5         IGNORE ENDPOINTS DURING AVERAGE 02559
3450 10FC 0206 0001          LI R6,1         COUNT OF AVERAGE PASSES 02560
3451 1E00 C820 333A D932          MOV C1,SWEEPS   SET SOFTWARE SWEEPS COUNTER TO 1 02561
3452 1E06 C820 D9D0 E00A          MOV ACQWFM,AMDWRD  START WAVEFORM ACQUISITION 02562
3453 1E0C 0300 000F          NXTPASS LIMB $F  ENABLE SYSTEM INTERRUPTS 02563
3454 1E10 C2E0 D9C6          MOV REALTIME,R11  UPDATE AVERAGES LEFT DISPLAY IN CALC MODE 02564
3455 1E14 1317          JEQ NOAVERO     02565
3456 1E16 1516          JGT NOAVERO     02566
3457 1E18 0208 3620          LI R11,XSTCK    POINTER TO CONVERSION INFO FOR X STACK 02567
3458 1E1C 0649          DECT SOFT       02568
3459 1E1E C67B          MOV *R11+,*SOFT  PUT LINE # ON STACK    02569
3460 1E20 0649          DECT SOFT       02570
3461 1E22 C67B          MOV *R11+,*SOFT  PUT CHARACTER # ON STACK 02571
3462 1E24 0649          DECT SOFT       02572
3463 1E26 C660 3356          MOV C15,*SOFT   PUT EXPONENT ON STACK  02573
3464 1E2A 0649          DECT SOFT       02574
3465 1E2C C643          MOV R3,*SOFT    PUT MANTISSA ON STACK  02575
3466 1E2E 0649          DECT SOFT       02576
3467 1E30 C2DB          MOV *R11,R11    GET CONVERSION FORMAT  02577
3468 1E32 024B 3FFF          ANDI R11,3FFF   CHANGE FROM ENGINEERING TO INTEGER 02578
3469 1E36 C648          MOV R11,*SOFT   PUT CONVERSION FORMAT ON STACK 02579
3470 1E38 0420 75E0          BLWP FP2ASC     CONVERT COUNT REMAINING TO ASCII 02580
3471 1E3C 06A0 1362          BL TEXTS        PUT AVERAGE COUNT UP ON DISPLAY 02581
3472 1E40 0229 0004          AI SOFT,4       POP LINE # AND CHARACTER # OFF STACK 02582
3473 1E44 C2E0 D982          NOAVERO MOV SWEEPS,R11  WAIT FOR A COMPLETE SWEEP TO BE ACQUIRED 02583
3474 1E48 15FD          JGT NOAVERO     02584
3475 1E4A C820 333A D982          MOV C1,SWEEPS   SET SOFTWARE SWEEPS COUNTER TO 1 02585
3476 1E50 C220 3390          NEWGHT MOV BADPNT,R8   OFFSCREEN POINT ($8000) 02586
3477 1E54 04CB          CLR R11         CREATE DOUBLE WORD OF AVERAGE PASS NUMBER 02587
3478 1E56 C306          MOV R6,R12      02588
3479 1E58 3EE0 333E          DIV C3,R11     1/3 OF AVERAGE PASS NUMBER 02589
3480 1E5C A2C6          A R6,R11       1.3333 OF AVERAGE PASS NUMBER 02590
3481 1E5E 0700          SETJ R0        02591
3482 1E60 0588          INC R0         INT (LJS BASE 2 (1.3333 * PASS NUMBER)) 02592
3483 1E62 091B          SRL R11,1      02593
3484 1E64 15FD          JGT *-4        02594
3485 1E66 C000          MOV R0,R0      CHECK VALUE OF R0     02595

```

'AVE' WAVEFORM ACQUISITION

02543

3486	1E68	1502		JGT	PNTCHK	MINIMUM VALUE IS 1	02596	
3487	1E6A	0200	0001	LI	R0,1		02597	
3488	1E6E	0720	DA16	PNTCHK	SET0	TELL DISPLAY THAT IT IS FLICKERING	02598	
3489	1E72	0212		C	*R2,R0	HAS THIS POINT BEEN ACQUIRED	02599	
3490	1E74	130C		JEQ	NOTAQRD	NO, THEN SKIP TO NEXT	02600	
3491	1E76	0211		C	*R1,R0	HAS THIS POINT BEEN ACQUIRED BEFORE	02601	
3492	1E78	1306		JEQ	FRSTPT	NO, JUST TRANSFER POINT FIRST TIME	02602	
3493	1E7A	C2D2		MOV	*R2,R11	YES, NEW POINT MUST BE WEIGHTED IN	02603	
3494	1E7C	CC88		MOV	R0,*R2+	SET POINT TO UNDEFINED AGAIN	02604	
3495	1E7E	62D1		S	*R1,R11	NEW POINT - AVERAGE	02605	
3496	1E80	000B		SRA	R11,R0	(NEW POINT - AVERAGE) / N, N IS POWER OF 2	02606	
3497	1E82	AC4B		A	R11,*R1+	ADJUST NEW AVERAGE	02607	
3498	1E84	1007		JMP	NXTPNT		02608	
3499	1E86	CC52		FRSTPT	MOV	*R2,*R1+	JUST TRANSFER POINT FIRST TIME ACQUIRED	02609
3500	1E88	CC88		MOV	R0,*R2+	SET ACQUIRED POINT TO UNDEFINED	02610	
3501	1E8A	1004		JMP	NXTPNT		02611	
3502	1E8C	C488		SAMPNT	MOV	R0,*R2		02612
3503	1E8E	05C1		NOTAQRD	INCT	R1	ADVANCE TO NEXT POINT	02613
3504	1E90	05C2			INCT	R2		02614
3505	1E92	0504			INC	R4	ADJUST ACQUIRED POINT COUNT	02615
3506	1E94	0605		NXTPNT	DEC	R5	CHECK IF DONE WITH PASS THROUGH WAVEFORM	02616
3507	1E96	1509		JGT	CHKDIV		02617	
3508	1E98	C160	0970	MOV	RESOLV,R5	IF SO START BACK AT BEGINNING	02618	
3509	1E9C	0645		DECT	R5	IGNORE ENDPOINTS DURING AVERAGE	02619	
3510	1E9E	C060	DA02	MOV	WOADD,R1		02620	
3511	1EA2	05C1		INCT	R1	SKIP FIRST POINT	02621	
3512	1EA4	C099		MOV	*SOFT,R2		02622	
3513	1EA6	05C2		INCT	R2	SKIP FIRST POINT	02623	
3514	1EA8	10B1		JMP	NXTPASS		02624	
3515	1EAA	0604		CHKDIV	DEC	R4	DECREMENT ACQUIRED POINT COUNT	02625
3516	1EAC	15E0		JGT	PNTCHK		02626	
3517	1EAE	C120	0970	MOV	RESOLV,R4	EACH ACQUISITION PASS IS RESOLUTION POINTS	02627	
3518	1EB2	0586		INC	R6	INCREMENT COUNT OF PASSES	02628	
3519	1EB4	0603		DEC	R3	DECREMENT REMAINING PASS COUNT	02629	
3520	1EB6	15CC		JGT	NEWGHT		02630	
3521	1EB8	0380		RTWP			02631	

'AQS' WAVEFORM ACQUISITION

02632

3523	1EBA		D880		AQSWFM	WORD	WPLVL2	LEVEL 2 ROUTINE	02633
3524	1EBC		1EBE			WORD	*+2		02634
3525	1EBE	C260	0012			MOV	18(R13),SOFT	GET CURRENT SOFTSTACK POINTER	02635
3526	1EC2	04E0	D9D2			CLR	ACQWFM1	STOP ACQUIRE AFTER ACQUIRE DONE INTERRUPT	02636
3527	1EC6	0201	D97E			LI	R1,DSPBUF	WAVEFORM ADDRESS(ES)	02637
3528	1ECA	8819	333A			C	*SOFT,C1	CHECK NUMBER OF WAVEFORMS TO ACQUIRE	02638
3529	1ECE	1306				JEQ	AQS1WFM		02639
3530	1ED0	C831	E002		AQS2WFM	MOV	*R1+,AWRD	A WORD GETS ADDRESS OF SECOND WFM	02640
3531	1ED4	C811	E004			MOV	*R1,3WRD	B WORD GET ADDRESS OF FIRST WFM	02641
3532	1ED8	0641				DECT	R1	REPOINT TO A WORD'S PRELOAD	02642
3533	1EDA	1004				JMP	WFMAQS		02643
3534	1EDC	C811	E002		AQS1WFM	MOV	*R1,AWRD	BOTH A WORD AND B WORD GET ADDRESS	02644
3535	1EE0	C811	E004			MOV	*R1,3WRD	FIRST WFM	02645
3536	1EE4	4820	3386	D938	WFMAQS	SZC	ACQINT,INTFLAG	REFLAG INDICATING ACQUIRE DONE	02646
3537	1EEA	04E0	D9D2			CLR	ACQWFM1	STOP EXTERNAL ACQUIRE AFTER 1ST INTERRUPT	02647
3538	1EEE	C820	D9D0	E00A		MOV	ACQWFM,AMDWRD	START EXTERNAL ACQUISITION	02648
3539							*****		
3540							*		
3541							*	AN EXTERNAL ACQUISITION STOPS THE 9900 FROM EXECUTING UNTIL THE	
3542							*	ACQUISITION HAS COMPLETED. IT DOES THIS BY NEVER RELEASING THE	
3543							*	SYSTEM'S BUS UNTIL THE ACQUISITION HAS FINISHED. THE NEXT	
3544							*	INSTRUCTION IS A DUMMY INSTRUCTION TO GUARANTEE THAT THE EXTERNAL	
3545							*	ACQUISITION HAS COMPLETED BY THE TIME THE A WORD IS READ.	
3546							*		
3547							*****		
3548	1EF4	C000				MOV	R0,R1	DUMMY INSTRUCTION (JUST FOR SAFETY'S SAKE)	02649
3549	1EF6	C0A0	E002			MOV	AWRD,R2	YES, READ ADDRESS OF LAST ACQUIRED POINT	02650
3550	1EFA	04E0	E00A			CLR	AMDWRD	STOP EXTERNAL ACQUIRE	02651
3551	1EFE	6091				S	*R1,R2	RELATIVE DISPLACEMENT OF LAST POINT IN WFM	02652
3552	1F00	0203	D97E			LI	R3,DSPBUF	BUFFER CONTAINING ADDRESSES OF ACQUIRED WFMS	02653
3553	1F04	C073			AQSS4FT	MOV	*R3+,R1	GET NEXT WAVEFORM TO ADJUST	02654
3554	1F06	1307				JEQ	AQSDONE	ADDRESS = 0 => DONE	02655
3555	1F08	0420	56AA			BLWP	ROTATE	ROTATE WAVEFORM TO CORRECT POSITION	02656
3556	1F0C	0649				DECT	SOFT		02657
3557	1F0E	C641				MOV	R1,*SOFT	PUSH WAVEFORM ADDRESS ONTO SOFTSTACK	02658
3558	1F10	0420	6BEE			BLWP	PNTFILL	INTERPOLATE IN UNFILLED POINTS	02659
3559	1F14	10F7				JMP	AQSSHFT		02660
3560	1F16	0380			AQSDONE	RTWP			02661

'RDOU' KEY HANDLER - READOUT ACQUISITION

02662

```

3562 *****
3563 **
3564 ** 'RDOU' KEY HANDLER **
3565 ** ** **
3566 ** LEVEL 1 ROUTINE **
3567 ** ** **
3568 ** INPUT- USER **
3569 ** OUTPUT- READOUT(X) ON USER **
3570 ** ** **
3571 ** STACK OPERATIONS- **
3572 ** SOFTSTACK- USED BUT NO AFFECT TO CALLER **
3573 ** USERSTACK- POPS 1 PUSHES 1 **
3574 ** ** **
3575 ** NOTE --- **
3576 ** VALID INPUTS- **
3577 ** ** **
3578 ** 0 1 2 3 **
3579 ** 4 5 6 7 **
3580 ** ** **
3581 *****
    
```

3582	1F18	06A0	690E	KEYRDOU BL	POPREG	POP X ELEMENT FROM USER STACK	02663
3583	1F1C	1627		JNE	RDERR	MFM # IN X IS ERROR	02664
3584	1F1E	06A0	75A0	BL	FP2INT	CONVERT FLOATING POINT NUMBER TO INTEGER	02665
3585	1F22	1124		JLT	RDERR	RANGE IS 0<=X<=7	02666
3586	1F24	0281	0007	CI	R1,7		02667
3587	1F28	1521		JST	RDERR		02668
3588	1F2A	0300	0007	LIHI	7	MASK OFF SYSTEM INTERRUPTS	02669
3589	1F2E	0420	1162	BLWP	ROAQR	DISPLAY REALTIME READOUT	02670
3590	1F32	0649		DECT	SOFT		02671
3591	1F34	C641		MOV	R1,*SOFT	PUSH READOUT POSITION ONTO SOFTSTACK	02672
3592	1F36	0420	20AE	BLWP	RDIN	DECODE REALTIME READOUT WORD X	02673
3593	1F3A	05C9		INCT	SOFT	POP UNITS OFF SOFTSTACK	02674
3594	1F3C	04C0		CLR	R0	FLAG INDICATING FLOATING POINT NUMBER	02675
3595	1F3E	C860	D94C	MOV	WARNING,R1	CHECK IF WARNING FLAG HAS BEEN SET	02676
3596	1F42	1103		JLT	*+8	IF NOT, THEN A NUMBER WAS ACQUIRED	02677
3597	1F44	04C1		CLR	R1	IF SO, THEN NO NUMBER WAS ACQUIRED SO	02678
3598	1F46	04C2		CLR	R2	DEFAULT VALUE IS 0	02679
3599	1F48	1002		JMP	*+6		02680
3600	1F4A	C079		MOV	*SOFT+,R1	POP FLOATING POINT READOUT NUMBER	02681
3601	1F4C	C0B9		MOV	*SOFT+,R2		02682
3602	1F4E	06A0	6962	BL	PSHREG		02683
3603	1F52	0649		DECT	SOFT		02684
3604	1F54	C660	3358	MOV	C16,*SOFT	CLEAR LINE 16 OF DISPLAY	02685
3605	1F58	0649		DECT	SOFT		02686
3606	1F5A	C660	333A	MOV	C1,*SOFT		02687
3607	1F5E	06A0	12FA	BL	NULLIN		02688
3608	1F62	0560	DACC	INV	OPROGLN		02689
3609	1F66	0720	D972	SETJ	RDTFLAG	PUT UP OLD READOUT	02690
3610	1F6A	0380		RTWP			02691
3611	1F6C	04E0	D94A	RJTERR CLR	FATAL	SET ERROR FLAG	02692
3612	1F70	C020	3304	MOV	WSTK,R0		02693
3613	1F74	06A0	6936	BL	PSHSTK	PUSH OLD VALUE BACK ONTO STACK	02694
3614	1F78	0380		RTWP			02695

```

3516 *****
3517 **
3518 ** COUNT CURRENT NUMBER OF REALTIME WAVEFORMS **
3519 ** **
3520 ** LEVEL 5 ROUTINE **
3521 ** **
3522 ** INPUT: NONE **
3523 ** OUTPUT: NONE **
3524 ** DESTROYS: R7,R8 **
3525 ** **
3526 ** STACK OPERATIONS: **
3527 ** SOFTSTACK: PUSHES 1 **
3528 ** **
3529 ** NOTE --- **
3530 ** THIS ROUTINE USES THREE TABLES TO COUNT THE NUMBER OF **
3531 ** REALTIME WAVEFORMS CURRENTLY BEING DISPLAYED. TWO **
3532 ** OF THESE TABLES ARE USED TO COMPUTE AN INDEX INTO THE **
3533 ** THIRD TABLE WHICH CONTAINS THE NUMBER OF REALTIME **
3534 ** WAVEFORMS BEING DISPLAYED. THE NUMBER FOUND IN THE **
3535 ** THE THIRD TABLE MUST BE DOUBLED IF THE DELAY MODE IS **
3536 ** ON. IF THIS ROUTINE DETECTS AN INVALID STATE A COUNT **
3537 ** OF ZERO IS RETURNED. **
3538 ** **
3539 *****
    
```

3640	1F7A	C1E0	DA26	CNTWFM	MOV	VERTM,R7	MAINFRAME'S CURRENT VERTICAL DISPLAY MODE	02697	
3641	1F7E	2227	2236		MOV3	MF.WFM(R7),R8	FIRST COMPONENT OF INDEX	02698	
3642	1F82	1129			JLT	CWERR	IF NEGATIVE THEN INVALID STATE	02699	
3643	1F84	C1E0	DA28		MOV	HORZM,R7	MAINFRAME'S CURRENT HORIZONTAL DISPLAY MODE	02700	
3644	1F88	B227	2236		AB	MF.WFM(R7),R8	ADD SECOND COMPONENT OF INDEX TO FIRST	02701	
3645	1F8C	1124			JLT	CWERR	IF NEGATIVE THEN INVALID STATE	02702	
3646	1F8E	1823			JCC	CWERR	IF CARRY THEN INVALID STATE	02703	
3647	1F90	C1E0	E010		MOV	FP,R7	FRONT PANEL READ DATA WORD	02704	
3648	1F94	0977			SRL	R7,7	SHIFT BITS 7-10 TO RIGHT 4 BITS	02705	
3649	1F96	0247	000F		ANDI	R7,\$F	KEEP BITS 7-10 (VERTICAL PLUG-INS' MODES)	02706	
3650	1F9A	8820	DA26	3350	C	VERTM,VERTL	IF VERTICAL LEFT THEN	02707	
3651	1FA0	1602			JNE	*+6	DEFAULT RIGHT PLUG-IN TO 1+2	02708	
3652	1FA2	0247	000C		ANDI	R7,\$C		02709	
3653	1FA6	8820	DA26	333C	C	VERTM,VERTR	IF VERTICAL RIGHT THEN	02710	
3654	1FAC	1602			JNE	*+6	DEFAULT LEFT PLUG-IN TO 1+2	02711	
3655	1FAE	0247	0003		ANDI	R7,\$3		02712	
3656	1FB2	B227	2246		AB	PI.WFM(R7),R8	ADD LAST COMPONENT OF INDEX TO FIRST TWO	02713	
3657	1FB6	110F			JLT	CWERR	IF NEGATIVE THEN INVALID STATE	02714	
3658	1FB8	183E			JCC	CWERR	IF CARRY THEN INVALID STATE	02715	
3659	1FBA	0988			SRL	R8,8	PUT INDEX IN LOW BYTE, ZERO HIGH BYTE	02716	
3660	1FBC	D228	2256		MOV3	NUMWFMS(R8),R8	GET NUMBER OF REALTIME WAVEFORMS	02717	
3661	1FC0	0988			SRL	R8,8	PUT NUMBER IN LOW BYTE, ZERO HIGH BYTE	02718	
3662	1FC2	2220	3358		CDC	C16,R8	ARE PLUGINS IS SLAVE MODE?	02719	
3663	1FC6	1608			JNE	NOSLAVE	NO, THEN NUMBER IS CORRECT	02720	
3664	1FC8	C1E0	E010		MOV	FP,R7	YES, GET FRONT PANEL READ DATA WORD	02721	
3665	1FCC	21E0	3340		CDC	DLYMODE,R7	IS THE DELAY MODE ON?	02722	
3666	1FD0	1603			JNE	NOSLAVE	NO, THEN NUMBER IS CORRECT	02723	
3667	1FD2	0A18			SLA	R8,1	YES, NUMBER MUST BE DOUBLED	02724	
3668	1FD4	1001			JMP	NOSLAVE		02725	
3669	1FD6	0438			CWERR	CLR	R8	ON INVALID STATE NUMBER IS ZERO	02726
3670	1FD8	0248	000F		NOSLAVE	ANDI	R8,\$000F	KEEP ONLY NUMBER OF WAVEFORMS	02727
3671	1FDC	0549			DECT	SOFT		02728	
3672	1FDE	C648			MOV	R8,*SOFT	PUSH RESULT ONTO SOFTSTACK	02729	
3673	1FE0	045B			B	*R11	RETURN TO CALLING ROUTINE	02730	

```

3675 *****
3676 **
3677 ** GET SEARCH PATTERNS FOR VERTICAL AND HORIZONTAL **
3678 ** REALTIME SCALE FACTORS **
3679 **
3680 ** INPUT: NONE **
3681 ** OUTPUT: NONE **
3682 ** DESTROYS: R7,R8 **
3683 **
3684 ** STACK OPERATIONS: **
3685 ** SOFTSTACK: PUSHES 2 **
3686 **
3687 ** NOTE --- **
3688 ** THIS ROUTINE FINDS THE SEARCH PATTERNS TO MATCH THE **
3689 ** REALTIME SCALE FACTORS TO THE REALTIME WAVEFORMS. FOUR **
3690 ** TABLES ARE USED TO FIND THESE PATTERNS. THE VERTICAL **
3691 ** READOUT SEARCH PATTERNS ARE FOUND USING THREE TABLES **
3692 ** TWO WHICH DEFINE AN INDEX INTO A THIRD WHICH CONTAINS **
3693 ** THE PATTERN. THE HORIZONTAL READOUT PATTERNS ARE FOUND **
3694 ** USING ALL FOUR TABLES THREE WHICH DEFINE AN INDEX INTO **
3695 ** A FOURTH WHICH CONTAINS THE PATTERN. **
3696 **
3697 ** THE SEARCH PATTERNS HAVE THE FOLLOWING FORM: **
3698 **
3699 ** $WXYZ - WHERE WX RELATE TO W0, YZ TO W1 **
3700 **
3701 ** THE HIGH BYTE DEFINES THE SEARCH PATTERN FOR THE FIRST **
3702 ** WAVEFORM (W0) AND THE LOW BYTE DEFINES THE SEARCH **
3703 ** PATTERN FOR THE SECOND WAVEFORM (W1) IF IT EXISTS. **
3704 ** EACH SEARCH PATTERN CONSISTS OF TWO NUMBERS WHICH **
3705 ** RELATE TO THE POSITION OF THE REALTIME READOUT AS **
3706 ** SHOWN BELOW. **
3707 **
3708 ** CRT- 0 1 2 3 **
3709 ** 4 5 6 7 **
3710 **
3711 ** THE SEARCH FOR THE SCALE FACTOR IS TO BEGIN IN THE FIRST **
3712 ** READOUT POSITION AND CONTINUE TO THE SECOND READOUT **
3713 ** POSITION. IF BOTH POSITIONS CONTAIN A VALID SCALE **
3714 ** FACTOR THE 'ADD' MODE IS SELECTED. AN ENTRY OF A $ **
3715 ** IN ANY POSITION OF THE SEARCH PATTERN INDICATES THAT **
3716 ** THE SEARCH IS TO STOP AS NO MORE POSITIONS EXIST TO **
3717 ** SEARCH FURTHER. TWO $FFFF VALUES ARE RETURNED IF AN **
3718 ** INVALID STATE IS FOUND. **
3719 **
3720 **
3721 *****

```

```

3722 1FE2 C1E0 DA25 CHKRD MOV VERTM,R7 MAINFRAME'S CURRENT VERTICAL DISPLAY MODE 02732
3723 1FE6 D227 2292 MOVB MF.RJ(R7),R6 FIRST COMPONENT OF VERTICAL'S INDEX 02733
3724 1FEA 1129 JLT CRERRD IF NEGATIVE THEN INVALID STATE 02734
3725 1FEC C1E0 ED10 MOV FP,R7 FRONT PANEL READ DATA WORD 02735
3726 1FF0 0977 SRL R7,7 SHIFT BITS 7-10 TO RIGHT 4 BITS 02736
3727 1FF2 0247 000F ANDI R7,$F KEEP BITS 7-10 (VERTICAL PLUG-INS' MODES) 02737
3728 1FF6 8820 DA25 3350 C VERTM,VERTL IF VERTICAL LEFT THEN 02738
3729 1FFC 1602 JNE *+6 DEFAULT RIGHT PLUG-IN TO 1+2 02739
3730 1FFE 0247 000C ANDI R7,$C 02740
3731 2002 8820 DA25 333C C VERTM,VERTR IF VERTICAL RIGHT THEN 02741
3732 2008 1602 JNE *+6 DEFAULT LEFT PLUG-IN TO 1+2 02742
3733 200A 0247 0003 ANDI R7,$3 02743

```

3734	200E	B227	22A2	AB	PI.RD(R7),R8	ADD SECOND COMPONENT OF INDEX TO FIRST	02744	
3735	2012	1115		JLT	CRERR0	IF NEGATIVE THEN INVALID STATE	02745	
3736	2014	1814		JOC	CRERR0	IF CARRY THEN INVALID STATE	02746	
3737	2016	0988		SRL	R8,B	PUT INDEX IN LOW BYTE, ZERO HIGH BYTE	02747	
3738	2018	0649		DECT	SOFT		02748	
3739	201A	C668	2282	MOV	V.RD(R8),*SOFT	PUSH VERTICAL SEARCH PATTERN ONTO SOFTSTACK	02749	
3740	201E	1118		JLT	CRERR1	IF NEGATIVE THEN INVALID STATE	02750	
3741	2020	05C8		INCF	R8	POINT TO FIRST HORIZONTAL INDEX COMPONENT	02751	
3742	2022	C228	2282	MOV	V.RD(R8),R8	GET FIRST COMPONENT OF HORIZONTAL'S INDEX	02752	
3743	2026	110C		JLT	CRERR1	IF NEGATIVE THEN INVALID STATE	02753	
3744	2028	C1E0	DA28	MOV	HORZM,R7	GET MAINFRAME'S CURRENT HORIZONTAL DISPLAY MO	02754	
3745	202C	D1E7	2292	MOV3	HF.RD(R7),R7	GET SECOND COMPONENT OF HORIZONTAL'S INDEX	02755	
3746	2030	1107		JLT	CRERR1	IF NEGATIVE THEN INVALID STATE	02756	
3747	2032	0987		SRL	R7,B	PUT COMPONENT IN LOW BYTE, ZERO HIGH BYTE	02757	
3748	2034	A207		A	R7,R8	ADD SECOND COMPONENT TO FIRST COMPONENT	02758	
3749	2036	0649		DECT	SOFT		02759	
3750	2038	C668	2302	MOV	H.RD(R8),*SOFT	PUSH HORIZONTAL SEARCH PATTERN ONTO SOFTSTACK	02760	
3751	203C	0458		B	*R11	RETURN TO CALLING ROUTINE	02761	
3752	203E	0649		CRERR0	DECT	SOFT	02762	
3753	2040	0719		CRERR1	SET3	*SOFT	PUSH NEGATIVE VALUE IF INVALID STATE	02763
3754	2042	0649		DECT	SOFT		02764	
3755	2044	0719		SET3	*SOFT	PUSH 2 NEGATIVE VALUES IF INVALID STATE	02765	
3756	2046	0458		B	*R11		02766	

```

3758 *****
3759 **
3760 ** CHECK IF READOUT POSITION IS VALID SCALE FACTOR **
3761 ** ** **
3762 ** LEVEL 5 ROUTINE **
3763 ** ** **
3764 ** INPUT: READOUT POSITIONS ON SOFTSTACK- 1 WORD **
3765 ** OUTPUT: VALID SCALE FACTOR POSITIONS ON SOFTSTACK- **
3766 ** 1 WORD **
3767 ** DESTROYS: R7,R8,R12 **
3768 ** ** **
3769 ** STACK OPERATIONS: **
3770 ** SOFTSTACK: POPS 1, PUSHES 1 **
3771 ** ** **
3772 ** NOTE --- **
3773 ** THIS ROUTINE CHECKS THE FOUR READOUT POSITIONS FOUND **
3774 ** IN THE TOP WORD OF THE SOFTSTACK. FOUR BITS ARE NEEDED **
3775 ** TO IDENTIFY A READOUT POSITIONS AS SHOWN: **
3776 ** ** **
3777 ** SOFTSTACK - $WXYZ **
3778 ** ** **
3779 ** WHERE W, X, Y, & Z EACH REPRESENT A READOUT POSITION **
3780 ** TO BE CHECKED. THESE POSITIONS MUST BE A NUMBER FROM **
3781 ** 0-7 OR $F. A $F INDICATES NO POSITION TO CHECK. AL **
3782 ** OTHER CODES WILL HAVE A $F SUBSTITUTED. ALL POSITIONS **
3783 ** WHICH ARE NOT VALID SCALE FACTORS WILL HAVE A $F **
3784 ** SUBSTITUTED FOR THEIR POSITION NUMBERS. **
3785 ** **

```

```

3786 *****
3787 2048 C1D9 CHKSF MOV *SOFT,R7 POP READOUT POSITIONS TO CHECK OFF SOFTSTACK 02768
3788 204A 0208 0004 LI R8,4 CHECK FOUR POSITIONS 02769
3789 204E C307 NEXTSF MOV R7,R12 02770
3790 2053 024C 000F ANDI R12,$F KEEP NEXT POSITION TO CHECK ONLY 02771
3791 2054 028C 0007 CI R12,7 IS THIS A VALID READOUT POSITION NUMBER? 02772
3792 2058 1522 JGT NOTSF NO, THEN SUBSTITUTE A $F INDICATING NOT SCALE 02773
3793 205A 0A1C SLA R12,1 CREATE WORD INDEX 02774
3794 205C C2AC 230E MOV TEKRD(R12),R10 GET ADDRESS OF TEKCODE READOUT 02775
3795 2060 C31A MOV *R10,R12 GET TIMESLOT 1 OF READOUT 02776
3796 2062 024C 00F0 ANDI R12,$00F0 KEEP TIMESLOT'S ROW INFORMATION ONLY 02777
3797 2066 028C 00C0 CI R12,$00C0 IS THIS A VALID SCALE FACTOR (ROW-3)? 02778
3798 206A 1619 JNE NOTSF NO, THIS IS NOT A SCALE FACTOR 02779
3799 206C 022A 0006 AI R10,6 ADDRESS OF TIMESLOT 4 (1-2-5 INFORMATION) 02780
3800 2070 C31A MOV *R10,R12 GET TIMESLOT 4 INFORMATION 02781
3801 2072 024C 080F ANDI R12,$F KEEP COLUMN INFORMATION ONLY 02782
3802 2076 028C 000D CI R12,$D IS THIS A SKIP COMMAND 02783
3803 207A 1311 JEQ NOTSF YES, THIS IS NOT A GOOD SCALE FACTOR 02784
3804 207C 022A FFFC AI R10,-4 POINT TO TIMESLOT #2 02785
3805 2080 C33A MOV *R10+,R12 GET TIMESLOT 2'S INFO 02786
3806 2082 024C 00FF ANDI R12,$FF KEEP GOOD INFO ONLY 02787
3807 2086 028C 00D7 CI R12,$D7 IS THIS A INVERTED INDICATOR? 02788
3808 208A 1306 JEQ WARNSF YES, THIS IS A WARNING 02789
3809 208C C31A MOV *R10,R12 GET TIMESLOT 3'S INFO 02790
3810 208E 024C 00FF ANDI R12,$FF KEEP GOOD INFO ONLY 02791
3811 2092 028C 00DE CI R12,$DE IS THIS A UNCAL INDICATOR? 02792
3812 2096 1606 JNE ISSF NO, EVERYTHING IS OK 02793
3813 2098 04E0 094C WARNSF CLR WARNING YES, THIS IS A WARNING 02794
3814 209C 1003 JMP ISSF 02795
3815 209E 020C 000F NOTSF LI R12,$000F FLAG INDICATING UNDEFINED READOUT POSITION 02796
3816 20A2 E1CC SOC R12,R7 NO, SET POSITION TO $F 02797

```

02767

3817	20A4	0B47	ISSF	SRC	R7,4	SHIFT NEXT POSITION TO RIGHT NYBBLE	02798
3818	20A6	0608		DEC	R8	MORE TO CHECK?	02799
3819	20A8	15D2		JGT	NEXTSF	YES, CONTINUE	02800
3820	20AA	C647		MOV	R7,*SOFT	NO, PUSH RESULT ONTO SOFTSTACK	02801
3821	20AC	0458		B	*R11	RETURN TO CALLING ROUTINE	02802

3882	2130	0649	DECT	SOFT		02850
3883	2132	0640	MOV	RD,*SOFT	PUSH UNITS ONTO SOFTSTACK	02851
3884	2134	0B49 0012	MOV	SOFT,16(R13)	RETURN CURRENT SOFTSTACK POINTER	02852
3885	2138	0380	RTWP			02853

LOCATE SCALE FACTOR

02854

```

3887 *****
3888 **
3889 **
3890 *****
3891 213A C339 W4CHSF MOV *SOFT+,R12 GET NUMBER OF WAVEFORMS BEING ACQUIRED 02855
3892 213C C109 MOV *SOFT,R7 GET SEARCH PATTERN FOR SCALE FACTORS 02856
3893 213E C207 MOV R7,R8 02857
3894 2140 0908 SRL R8,8 PUT SEARCH PATTERN FOR W0 WFM IN LOW BYTE 02858
3895 2142 C648 MOV R8,*SOFT PUSH THIS VALUE ONTO SOFTSTACK 02859
3896 2144 060C DEC R12 MORE THAN 1 WFM? 02860
3897 2146 1304 JEQ W0SF YES, DO W1 TOO 02861
3898 2148 0247 00FF ANDI R7,$00FF KEEP W1'S SEARCH PATTERN IN LOW BYTE 02862
3899 214C 0649 DECT SOFT 02863
3900 214E C647 MOV R7,*SOFT PUSH W1'S PATTERN ONTO SOFTSTACK 02864
3901 2150 050C W0SF INC R12 RESTORE COUNT OF WAVEFORMS 02865
3902 2152 020C 0002 CI R12,2 TWO WAVEFORMS? 02866
3903 2156 1101 JLT SFNEXT YES, ALREADY POINTING TO W0'S SEARCH PATTERN 02867
3904 2158 0509 INCT SOFT NO, POINT TO W0'S SEARCH PATTERN 02868
3905 215A C219 SFNEXT MOV *SOFT,R8 POP NEXT SEARCH PATTERN 02869
3906 215C 0948 SRL R8,4 SHIFT Y POSITION TO Z POSITION 02870
3907 215E 0248 000F ANDI R8,$000F KEEP Y OF $WXYZ PATTERN 02871
3908 2162 0208 0007 CI R8,$0007 IS THIS A VALID READOUT POSITION? 02872
3909 2166 150A JGT NOTYSF NO, LOOK AT NEXT 02873
3910 2168 C109 MOV *SOFT,R7 YES, GET OLD SEARCH PATTERN 02874
3911 216A 0247 000F ANDI R7,$000F KEEP Z OF $WXYZ PATTERN 02875
3912 216E 0207 0007 CI R7,$0007 IS THIS A VALID READOUT POSITION? 02876
3913 2172 1502 JGT LOADSF NO, THIS IS A WARNING 02877
3914 2174 04E0 D94C CLR WARNING 02878
3915 2176 C648 LJAJSF MOV R8,*SOFT PUSH SCALE FACTOR POSITION ONTO SOFTSTACK 02879
3916 217A 1008 JMP NXTSF CHECK IF MORE WAVEFORMS ARE TO BE DONE 02880
3917 217C C219 NOTYSF MOV *SOFT,R8 GET SEARCH PATTERN 02881
3918 217E 0248 000F ANDI R8,$000F KEEP Z PATTERN ONLY 02882
3919 2182 0208 0008 CI R8,$0008 IS THIS A VALID READOUT POSITION? 02883
3920 2186 11F8 JLT LOADSF YES, USE IT 02884
3921 2188 04E0 D94C CLR WARNING NO, THIS IS A WARNING 02885
3922 218C 0208 000F LI R8,$000F LOAD SF IF NO VALID READOUT POSITION 02886
3923 2190 10F3 JMP LJAJSF 02887
3924 2192 0649 NXTSF DECT SOFT POINT TO NEXT SEARCH PATTERN 02888
3925 2194 060C DEC R12 MORE WAVEFORMS? 02889
3926 2196 15E1 JGT SFNEXT YES, CONTINUE 02890
3927 2198 0509 INCT SOFT NO, BACKUP TO LAST SEARCH PATTERN 02891
3928 219A 0458 B *R11 RETURN TO CALLING ROUTINE 02892

```

WAVEFORM FILL CHECK

02893

ADDRESS	HEX	ASCII	DATA	OPERATION	COMMENT	ADDRESS
3930	219C		DBA0	FILLCHK	WORD	02894
3931	219E		21A0		WORD	02895
3932	21A0	C26D	0012	MOV	16(R13),SOFT	02896
3933	21A4	C0B9		MOV	*SOFT+,R2	02897
3934	21A6	04C3		CLR	R3	02898
3935	21A8	04C4		CLR	R4	02899
3936	21AA	04C5		CLR	R5	02900
3937	21AC	C1A0	D970	MOV	RESOLV,R6	02901
3938	21B0	8832	3390	CHK.PNT	C	02902
3939	21B4	1305		JEQ	NOTFILL	02903
3940	21B6	0583		INC	R3	02904
3941	21B8	8105		C	R5,R4	02905
3942	21BA	1101		JLT	*+4	02906
3943	21BC	C105		MOV	R5,R4	02907
3944	21BE	0705		SETJ	R5	02908
3945	21C0	0585		NOTFILL	INC	02909
3946	21C2	0606		DEC	R6	02910
3947	21C4	15F5		JGT	CHK.PNT	02911
3948	21C6	8105		C	R5,R4	02912
3949	21C8	1101		JLT	*+4	02913
3950	21CA	C105		MOV	R5,R4	02914
3951	21CC	0649		DECT	SOFT	02915
3952	21CE	C644		MOV	R4,*SOFT	02916
3953	21D0	0649		DECT	SOFT	02917
3954	21D2	C643		MOV	R3,*SOFT	02918
3955	21D4	C849	0012	MOV	SOFT,16(R13)	02919
3956	21D8	0300		RTW		02920

SWEEP INTERRUPT HANDLER

02921

3958	21DA	0300	0000		SWEEP	LIMI	0	MASK OFF ALL INTERRUPTS	02922
3959	21DE	C320	D906			MOV	INT7854,R12	SHOULD THIS INTERRUPT BE HANDLED BY	02923
3960	21E2	1302				JEQ	*+6	THE DIAGNOSTIC ROUTINE	02924
3961	21E4	0420	F044			BLWP	\$F044	YES, TRANSFER CONTROL TO IT	02925
3962	21E8	0300	000C			LIMI	\$C	RESET INTERRUPT MASK TO INITIAL LEVEL	02926
3963	21EC	C820	D902	E00A		MOV	ACQWFH1,ANDWRD	SET HARDWARE SWEEP COUNTER TO 1	02927
3964	21F2	0620	D932			DEC	SWEEPS	DECREMENT REMAINING SOFTWARE SWEEP COUNT	02928
3965	21F6	1502				JGT	*+6		02929
3966	21F8	04E0	D982			CLR	SWEEPS	MINIMUM VALUE IS 0	02930
3967	21FC	02CC				STST	R12	GET CURRENT STATUS	02931
3968	21FE	024C	000F			ANDI	R12,\$000F	KEEP CURRENT INTERRUPT MASK ONLY	02932
3969	2202	024F	FFF0			ANDI	R15,\$FFF0	REPLACE RETURN INTERRUPT MASK WITH CURRENT	02933
3970	2206	E3CC				SOC	R12,R15	INTERRUPT MASK TO KEEP SWEEP INTERRUPT MASKE	02934
3971	2208	E820	3386	D938		SOC	ACQINT,INTFLAG	OFF	02935
3972	220E	0380				RTW			02936

3974					*****			
3975					**		**	
3976					**	INTERRUPT HANDLER FOR UNDEFINED INTERRUPTS	**	
3977					**		**	
3978					**	NOTE ---	**	
3979					**	ONLY REGISTER 9-15 HAVE BEEN DEFINED FOR EACH OF	**	
3980					**	THESE ROUTINES.	**	
3981					**		**	
3982					*****			
3983	2210	0420	2215	INTU	BLWP	INTU1	'BLWP' TRANSFERS STATUS TO R15 & PREVENTS	02938
3984	2214	0380			RTWP		MORE INTERRUPTS UNTIL AFTER 'LIMI 0'	02939
3985	2216		0380	INTU1	WORD	WPINT1		02940
3986	2218		221A		WORD	*+2		02941
3987	221A	0300	0000		LIMI	0	MASK OFF ALL INTERRUPTS	02942
3988	221E	0320	0904		MOV	DIAGINT,R12	SHOULD THIS INTERRUPT BE HANDLED BY	02943
3989	2222	1601			JNE	*+4	A DIAGNOSTIC ROUTINE?	02944
3990	2224	0380			RTWP		NO, JUST RETURN	02945
3991	2226	030F			MOV	R15,R12	GET MPU STATUS PRIOR TO INTERRUPT	02946
3992	2228	024C	000F		ANDI	R12,\$000F	KEEP INTERRUPT MASK ONLY	02947
3993	222C	058C			INC	R12	INTERRUPT NUMBER	02948
3994	222E	0A2C			SLA	R12,2	CREATE TABLE INDEX	02949
3995	2230	042C	F010		BLWP	\$F010(R12)	TRANSFER CONTROL DIAGNOSTIC INTERRUPT HANDLER	02950
3996	2234	0380			RTWP			02951

TABLE - REALTIME WAVEFORMS

02952

3998 *
 3999 * THE NUMBER OF REALTIME WAVEFORMS CURRENTLY BEING DISPLAYED CAN
 4000 * BE DETERMINED FROM THE MAINFRAME'S VERTICAL AND HORIZONTAL
 4001 * DISPLAY MODES AND THE VERTICAL PLUG-INS' DISPLAY MODES. THIS
 4002 * INFORMATION IS IMPORTANT IN THE 7854 SYSTEM IN THAT DURING A
 4003 * WAVEFORM ACQUISITION ONLY 1 OR 2 REALTIME WAVEFORMS CAN BE
 4004 * ACQUIRED AT ANY ONE TIME.
 4005 *

4006 * THE FOLLOWING THREE TABLES ARE USED TO DETERMINE THE NUMBER
 4007 * OF REALTIME WAVEFORMS CURRENTLY BEING DISPLAYED. THE FIRST
 4008 * TWO ARE USED TO CALCULATE AN INDEX INTO THE THIRD WHICH LISTS
 4009 * THE NUMBER OF REALTIME WAVEFORMS. IT SHOULD BE NOTED THAT
 4010 * THE NUMBERS FOUND IN THE THIRD TABLE MUST BE DOUBLED IF A
 4011 * DELAYING-DELAYED TIMEBASE MODE IS BEING USED. ALSO SINCE
 4012 * THE ACQUISITION HARDWARE CANNOT DIGITIZE WAVEFORMS CORRECTLY
 4013 * WHEN ANY PLUG-IN IS IN THE 'CHOP' MODE THESE ENTRIES ARE NOT
 4014 * INCLUDED IN THE TABLES.
 4015 *

4016 * THE INDEX IS THE SUM OF THREE ENTRIES FROM THE FIRST TWO TABLES
 4017 * TWO ENTRIES FROM THE FIRST TABLE AND ONE ENTRY FROM THE SECOND.
 4018 * THE MAINFRAME'S VERTICAL AND HORIZONTAL MODES EACH DEFINE AN
 4019 * ENTRY IN THE FIRST TABLE WHILE THE VERTICAL PLUG-INS' MODES
 4020 * DEFINE THE ENTRY IN THE SECOND TABLE.
 4021 *

4022 *
 4023 * TABLE 1 - MAINFRAME'S VERTICAL AND HORIZONTAL DISPLAY MODES

4024 *
 4025 * INDEX = AS DEFINED BY BITS 11-14 OF THE FRONT PANEL
 4026 * READ DATA WORD
 4027 *

Address	Hex	Byte	Hex	Index	Description	Address
4028	2236	FF	MF.WFM	0		02953
4029	2237	FF		1		02954
4030	2238	30		2	VERTICAL RIGHT	02955
4031	2239	00		3	HORIZONTAL A	02956
4032	223A	FF		4		02957
4033	223B	FF		5		02958
4034	223C	04		6	HORIZONTAL ALT	02959
4035	223D	24		7	VERTICAL CHOP	02960
4036	223E	FF		8		02961
4037	223F	FF		9		02962
4038	2240	0C		A	VERTICAL ALT	02963
4039	2241	18		B	VERTICAL ADD	02964
4040	2242	00		C	VERTICAL LEFT	02965
4041	2243	00		D	HORIZONTAL B	02966
4042	2244	08		E	HORIZONTAL CHOP	02967
4043	2245	FF		F		02968

4044 *
 4045 * TABLE 2 - PLUG-INS' VERTICAL DISPLAY MODES

4046 *
 4047 * INDEX = AS DEFINED BY BITS 7-10 OF THE FRONT PANEL
 4048 * READ DATA WORD
 4049 *

Address	Hex	Byte	Hex	Index	Description	Address
4050	2246	00	PI.WFM	0	LEFT 1+2, RIGHT 1+2	02969
4051	2247	FF		1		02970
4052	2248	01		2	LEFT 1+2, RIGHT ALT	02971
4053	2249	FF		3		02972
4054	224A	FF		4		02973
4055	224B	FF		5		02974
4056	224C	FF		6		02975

TABLE - REALTIME WAVEFORMS

02952

4057	224D	FF	BYTE	\$FF	7		02976	
4058	224E	02	BYTE	2	8	- LEFT ALT, RIGHT 1+2	02977	
4059	224F	FF	BYTE	\$FF	9		02978	
4060	2250	03	BYTE	3	A	- LEFT ALT, RIGHT ALT	02979	
4061	2251	FF	BYTE	\$FF	B		02980	
4062	2252	FF	BYTE	\$FF	C		02981	
4063	2253	FF	BYTE	\$FF	D		02982	
4064	2254	FF	BYTE	\$FF	E		02983	
4065	2255	FF	BYTE	\$FF	F		02984	
4066		*						
4067		*	TABLE 3 - NUMBER OF DISPLAYED REALTIME WAVEFORMS					
4068		*						
4069		*	INDEX = SUM OF THREE ENTRIES FROM ABOVE TWO TABLES					
4070		*	TWO ENTRIES FROM TABLE 1 AND ONE ENTRY FROM					
4071		*	TABLE 2					
4072		*						
4073		*	ORDER/LINE = LEFT 1+2, RIGHT 1+2					
4074		*	LEFT 1+2, RIGHT ALT					
4075		*	LEFT ALT, RIGHT 1+2					
4076		*	LEFT ALT, RIGHT ALT					
4077		*						
4078		*	BIT 4 INDICATES SLAVE MODE (0 = NOT SLAVED, 1 = SLAVED)					
4079		*						
4080		*	BITS 0-3 INDICATE NUMBER OF REALTIME WAVEFORMS					
4081		*						
4082	2256	01	NUMWFMS	BYTE	\$01,\$01,\$02,\$02	VERTICAL LEFT , HORIZONTAL A/B	02985	
	2257	01						
	2258	02						
	2259	02						
4083	225A	02	BYTE	\$02,\$02,\$12,\$12	" "	, " ALT	02986	
	225B	02						
	225C	12						
	225D	12						
4084	225E	02	BYTE	\$02,\$02,\$12,\$12	" "	, " CHOP	02987	
	225F	02						
	2260	12						
	2261	12						
4085	2262	02	BYTE	\$02,\$03,\$03,\$04	VERTICAL ALT	, HORIZONTAL A/B	02988	
	2263	03						
	2264	03						
	2265	04						
4086	2266	12	BYTE	\$12,\$13,\$13,\$14	" "	, " ALT	02989	
	2267	13						
	2268	13						
	2269	14						
4087	226A	12	BYTE	\$12,\$13,\$13,\$14	" "	, " CHOP	02990	
	226B	13						
	226C	13						
	226D	14						
4088	226E	01	BYTE	\$01,\$02,\$02,\$02	VERTICAL ADD	, HORIZONTAL A/B	02991	
	226F	02						
	2270	02						
	2271	02						
4089	2272	02	BYTE	\$02,\$12,\$12,\$12	" "	, " ALT	02992	
	2273	12						
	2274	12						
	2275	12						
4090	2276	02	BYTE	\$02,\$12,\$12,\$12	" "	, " CHOP	02993	
	2277	12						

	2278	12						
	2279	12						
4091	227A	02	BYTE	\$02,\$03,\$03,\$04	VERTICAL CHOP ,	HORIZONTAL A/B		02994
	227B	03						
	227C	03						
	227D	04						
4092	227E	04	BYTE	\$04,\$14,\$14,\$14	" " , "	ALT		02995
	227F	14						
	2280	14						
	2281	14						
4093	2282	04	BYTE	\$04,\$14,\$14,\$14	" " , "	CHOP		02996
	2283	14						
	2284	14						
	2285	14						
4094	2286	01	BYTE	\$01,\$02,\$01,\$02	VERTICAL RIGHT,	HORIZONTAL A/B		02997
	2287	02						
	2288	01						
	2289	02						
4095	228A	02	BYTE	\$02,\$12,\$02,\$12	" " , "	ALT		02998
	228B	12						
	228C	02						
	228D	12						
4096	228E	02	BYTE	\$02,\$12,\$02,\$12	" " , "	CHOP		02999
	228F	12						
	2290	02						
	2291	12						

TABLE - REALTIME READOUTS

03000

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		2292	FF	MF.RD	BYTE	\$FF	0	03001
4125		2293	FF		BYTE	\$FF	1	03002
4126		2294	40		BYTE	64	2 - VERTICAL RIGHT	03003
4127		2295	00		BYTE	0	3 - HORIZONTAL A	03004
4128		2296	FF		BYTE	\$FF	4	03005
4129		2297	FF		BYTE	\$FF	5	03006
4130		2298	04		BYTE	4	6 - HORIZONTAL ALT	03007
4131		2299	30		BYTE	48	7 - VERTICAL CHOP	03008
4132		229A	FF		BYTE	\$FF	8	03009
4133		229B	FF		BYTE	\$FF	9	03010
4134		229C	10		BYTE	16	A - VERTICAL ALT	03011
4135		229D	20		BYTE	32	B - VERTICAL ADD	03012
4136		229E	00		BYTE	0	C - VERTICAL LEFT	03013
4137		229F	02		BYTE	2	D - HORIZONTAL B	03014
4138		22A0	04		BYTE	4	E - HORIZONTAL CHOP	03015
4139		22A1	FF		BYTE	\$FF	F	03016
4140	*							
4141	*							
4142	*							
4143	*							
4144	*							
4145	*							
4146		22A2	00	PI.RD	BYTE	0	0 - LEFT 1+2, RIGHT 1+2	03017
4147		22A3	FF		BYTE	\$FF	1	03018
4148		22A4	04		BYTE	4	2 - LEFT 1+2, RIGHT ALT	03019
4149		22A5	FF		BYTE	\$FF	3	03020
4150		22A6	FF		BYTE	\$FF	4	03021
4151		22A7	FF		BYTE	\$FF	5	03022
4152		22A8	FF		BYTE	\$FF	6	03023
4153		22A9	FF		BYTE	\$FF	7	03024
4154		22AA	08		BYTE	8	8 - LEFT ALT, RIGHT 1+2	03025
4155		22AB	FF		BYTE	\$FF	9	03026
4156		22AC	0C		BYTE	12	A - LEFT ALT, RIGHT ALT	03027

TABLE - REALTIME READOUTS

03000

4157	22AD	FF		BYTE	\$FF	B			03028
4158	22AE	FF		BYTE	\$FF	C			03029
4159	22AF	FF		BYTE	\$FF	D			03030
4160	22B0	FF		BYTE	\$FF	E			03031
4161	22B1	FF		BYTE	\$FF	F			03032

TABLE 3 - VERTICAL SCALE FACTOR SEARCH PATTERN, HORIZONTAL INDEX

INDEX = SUM OF THREE ENTRIES FROM ABOVE TWO TABLES:
TWO ENTRIES FROM TABLE 1 AND ONE ENTRY FROM TABLE 2

4169	22B2 22B4	0404 0000	V.RD	WORD	\$0404,0	VERTICAL LEFT,	1+2	1+2	03033
4170	22B6 22B8	0404 0000		WORD	\$0404,0	" "	, 1+2	ALT	03034
4171	22BA 22BC	4F0F 0006		WORD	\$4F0F,6	" "	, ALT	1+2	03035
4172	22BE 22C0	4F0F 0005		WORD	\$4F0F,6	" "	, ALT	ALT	03036
4173	22C2 22C4	1504 0006		WORD	\$1504,6	VERTICAL ALT,	1+2	1+2	03037
4174	22C6 22C8	FFFF FFFF		WORD	\$FFFF,-1	" "	, 1+2	ALT	03038
4175	22CA 22CC	FFFF FFFF		WORD	\$FFFF,-1	" "	, ALT	1+2	03039
4176	22CE 22D0	FFFF FFFF		WORD	\$FFFF,-1	" "	, ALT	ALT	03040
4177	22D2 22D4	0404 0000		WORD	\$0404,0	VERTICAL ADD,	1+2	1+2	03041
4178	22D6 22D8	5010 0005		WORD	\$5010,6	" "	, 1+2	ALT	03042
4179	22DA 22DC	4101 0006		WORD	\$4101,6	" "	, ALT	1+2	03043
4180	22DE 22E0	4501 0006		WORD	\$4501,6	" "	, ALT	ALT	03044
4181	22E2 22E4	1504 0006		WORD	\$1504,6	VERTICAL CHOP,	1+2	1+2	03045
4182	22E6 22E8	FFFF FFFF		WORD	\$FFFF,-1	" "	, 1+2	ALT	03046
4183	22EA 22EC	FFFF FFFF		WORD	\$FFFF,-1	" "	, ALT	1+2	03047
4184	22EE 22F0	FFFF FFFF		WORD	\$FFFF,-1	" "	, ALT	ALT	03048
4185	22F2 22F4	1515 0000		WORD	\$1515,0	VERTICAL RIGHT,	1+2	1+2	03049
4186	22F6 22F8	5F1F 0005		WORD	\$5F1F,6	" "	, 1+2	ALT	03050
4187	22FA 22FC	1515 0000		WORD	\$1515,0	" "	, ALT	1+2	03051
4188	22FE 2300	5F1F 0005		WORD	\$5F1F,6	" "	, ALT	ALT	03052

TABLE 4 - HORIZONTAL SCALE FACTOR SEARCH PATTERN

INDEX = SUM OF ENTRY FROM TABLE 1 AND ENTRY FROM TABLE 3

4195	2302	26FF	H.RD	WORD	\$26FF	HORIZONTAL A, 1 VERTICAL WAVEFORM			03053
------	------	------	------	------	--------	-----------------------------------	--	--	-------

TABLE - REALTIME READOUTS

03000

4196	2304	37FF	WORD	\$37FF	"	B, 1 VERTICAL WAVEFORM	03054
4197	2306	2637	WORD	\$2637	"	ALT/CHOP, 1 VERTICAL WAVEFORM	03055
4198	2308	2626	WORD	\$2626	HORIZONTAL	A, 2 VERTICAL WAVEFORMS	03056
4199	230A	3737	WORD	\$3737	"	B, 2 VERTICAL WAVEFORMS	03057
4200	230C	2637	WORD	\$2637	"	ALT/CHOP, 2 VERTICAL WAVEFORMS	03058

4202			*									
4203			*	THIS TABLE IS USED TO FIND THE ADDRESS OF A READOUT POSITION								
4204			*	IN THE ACQUIRED TEKCODE COPY. THE INDEX IS THE NUMBER OF								
4205			*	THE READOUT POSITION WANTED:								
4206			*									
4207			*		0	1	2	3				
4208			*	CRT-								
4209			*		4	5	6	7				
4210			*									
4211	230E	DF00	TEKRD	WORD	TEKCODE+00	POSITION	0					03060
4212	2310	DF28		WORD	TEKCODE+40	"	1					03061
4213	2312	DF58		WORD	TEKCODE+80	"	2					03162
4214	2314	DF78		WORD	TEKCODE+120	"	3					03063
4215	2316	DF14		WORD	TEKCODE+20	"	4					03064
4216	2318	DF3C		WORD	TEKCODE+60	"	5					03065
4217	231A	DF64		WORD	TEKCODE+100	"	6					03066
4218	231C	DF8C		WORD	TEKCODE+140	"	7					03067

4219 *
 4220 * THIS TABLE IS USED TO FIND THE LINE NUMBER AND CHARACTER NUMBER
 4221 * WHERE A READOUT POSITION IS LOCATED IN THE ASCII COPY. THIS
 4222 * DATA CAN BE USED IN CALLING THE ROUTINE 'OLDTXT' TO GET THE
 4223 * ACTUAL ADDRESS OF THE ASCII READOUT. THE INDEX IS THE NUMBER
 4224 * OF THE READOUT POSITION WANTED:

4225			*									
4226			*		0	1	2	3				
4227			*	CRT-								
4228			*		4	5	6	7				
4229			*									
4230	231E	01	ASCRO	BYTE	1,1	POSITION	0,	LINE	#1,	CHARACTER	#1	03068
	231F	01										
4231	2320	01		BYTE	1,11	"	1,	"	#1,	"	#11	03069
	2321	0B										
4232	2322	01		BYTE	1,21	"	2,	"	#1,	"	#21	03070
	2323	15										
4233	2324	01		BYTE	1,31	"	3,	"	#1,	"	#31	03071
	2325	1F										
4234	2326	10		BYTE	16,1	"	4,	"	#16,	"	#1	03072
	2327	01										
4235	2328	10		BYTE	16,11	"	5,	"	#16,	"	#11	03073
	2329	0B										
4236	232A	10		BYTE	16,21	"	6,	"	#16,	"	#21	03074
	232B	15										
4237	232C	10		BYTE	16,31	"	7,	"	#16,	"	#31	03075
	232D	1F										

4238 *
 4239 * THIS TABLE IS USED TO FIND THE LINE NUMBER AND CHARACTER NUMBER
 4240 * OF THE ASCII READOUT WHICH CORRESPONDS TO ITS TEKCODE READOUT
 4241 * COUNTER-PART. THE INDEX IS THE READOUT'S POSITION IN THE
 4242 * TEKCODE READOUT.

4243			*									
4244			*		0	1	2	3				
4245			*	CRT-								
4246			*		4	5	6	7				
4247			*									
4248			*	TEKCODE COPY -	0	4	1	5	2	6	3	7
4249			*									
4250	232E	01	TEKASCRO	BYTE	1,1	POSITION	0					03076
	232F	01										
4251	2330	10		BYTE	16,1	"	4					03077

	2331	01				
4252	2332	01	BYTE 1,11	"	1	03078
	2333	0B				
4253	2334	10	BYTE 16,11	"	5	03079
	2335	0B				
4254	2336	01	BYTE 1,21	"	2	03080
	2337	15				
4255	2338	10	BYTE 16,21	"	6	03081
	2339	15				
4256	233A	01	BYTE 1,31	"	3	03082
	233B	1F				
4257	233C	10	BYTE 16,31	"	7	03083
	233D	1F				

FRONT PANEL ACQUISITION MODES

03084

4259	233E	FF	FPACQ	BYTE	\$FF	0		03085
4260	233F	FF		BYTE	\$FF	1		03086
4261	2340	80		BYTE	\$80	2	- VERTICAL RIGHT	03087
4262	2341	08		BYTE	\$08	3	- HORIZONTAL A	03088
4263	2342	FF		BYTE	\$FF	4		03089
4264	2343	FF		BYTE	\$FF	5		03090
4265	2344	00		BYTE	\$00	6	- HORIZONTAL ALT	03091
4266	2345	00		BYTE	\$00	7	- VERTICAL CHOP	03092
4267	2346	FF		BYTE	\$FF	8		03093
4268	2347	FF		BYTE	\$FF	9		03094
4269	2348	00		BYTE	\$00	A	- VERTICAL ALT	03095
4270	2349	60		BYTE	\$60	B	- VERTICAL ADD	03096
4271	234A	40		BYTE	\$40	C	- VERTICAL LEFT	03097
4272	234B	10		BYTE	\$10	D	- HORIZONTAL B	03098
4273	234C	00		BYTE	\$00	E	- HORIZONTAL CHOP	03099
4274	234D	FF		BYTE	\$FF	F		03100

>>>>> 7854 GPIB NOTES <<<<<<

03102

```

4277 *****
4278 **
4279 ** SPECIAL LEVEL 5 (A,B,C) SUBROUTINE CALLING PROTOCOL **
4280 **
4281 ** SEVERAL LEVEL 5 GPIB I/O SUBROUTINES HAVE SPECIAL RULES **
4282 ** FOR THEIR CALLING PROTOCOL. THESE ROUTINES HAVE A LETTER **
4283 ** APPENDED TO THE NUMBER 5 (5A, 5B, 5C FOR EXAMPLE). THE **
4284 ** LETTER INDICATES WHICH RULES MUST BE FOLLOWED WHEN USING **
4285 ** THIS SUBROUTINE. **
4286 **
4287 ** LEVEL 5A CAN CALL 5B & 5C. LEVEL 5B CAN CALL 5C. LEVEL 5C **
4288 ** IS A TERMINAL ROUTINE. **
4289 **
4290 ** LEVEL AVAILABLE REGISTERS RETURN REGISTER **
4291 ** 5A R0,R1,R2 R10 **
4292 ** 5B R0,R1,R2 R12 **
4293 ** 5C R1,R2 R11 **
4294 **
4295 *****
4296 *****
4297 **
4298 **
4299 *****
4300 *****
4301 **
4302 **
4303 *****
4304 *****
4305 **
4306 **
4307 *****
4308 *****
4309 **
4310 **
4311 *****

```

4313						*****		
4314						**		**
4315						**	HANDLER FOR GPIB INTERRUPT	**
4316						**		**
4317						**	INTERRUPT HANDLER	**
4318						**		**
4319						**	NOTE ---	**
4320						**	THIS ROUTINE DECODES THE TYPE OF GPIB INTERRUPT WHICH	**
4321						**	OCCURRED AND BRANCHES TO AN APPROPRIATE ROUTINE TO	**
4322						**	HANDLE THAT PARTICULAR INTERRUPT.	**
4323						**		**
4324						**	THIS ROUTINE CANNOT DESTROY ANY REGISTERS WHILE DECODING	**
4325						**	I/O OTHER THAN COMMAND AND QUERY INPUT.	**
4326						**		**
4327						*****		
4328	234E	0300	0000			GPIB	LIMI 0	MASK OFF ALL INTERRUPTS 03104
4329	2352	080C	092C				MOV R12,GPIBR12	SAVE R12 03105
4330	2356	0320	0906				MOV INT7854,R12	SHOULD THIS INTERRUPT BE HANDLED BY 03106
4331	235A	1302					JEQ *+6	A DIAGNOSTIC ROUTINE? 03107
4332	235C	0420	F03C				BLWP \$F03C	YES, TRANSFER CONTROL TO IT 03108
4333	2360	0300	000A				LIMI \$A	RESET INTERRUPT MASK TO INITIAL LEVEL 03109
4334	2364	0320	33C6				MOV IOLED,R12	CRU ADDRESS FOR I/O LED 03110
4335	2368	1000					SBO 0	TURN I/O LED ON 03111
4336	236A	0320	333A				MOV GPIBE0I,R12	CRU ADDRESS FOR <EOI> LINE 03112
4337	236E	1E08					SBZ 0	TURN <EOI> OFF 03113
4338	2370	0320	E040			RJSTST	MOVB R0R,R12	READ INTERRUPT STATUS & CLEAR INTERRUPT 03114
4339	2374	2320	337C				COB CMD,R12	IS IT A COMMAND INTERRUPT? 03115
4340	2378	160C					JNE NOTCMD	NO, CHECK OTHER INTERRUPTS 03116
4341	237A	0320	E044				MOVB R1R,R12	READ COMMAND STATUS & CLEAR CMD BIT 03117
4342	237E	2320	337C				COB SPAS,R12	IS THIS A SERIAL POLL ACTIVE STATE INT? 03118
4343	2382	1319					JEQ SPASIN	YES, NO HANDSHAKE TO COMPLETE 03119
4344	2384	0820	3380	E06C			MOV3 DACR,R3W	COMPLETE HANDSHAKE FOR COMMAND 03120
4345	238A	2320	337A				COB DCAS,R12	IS IT A 'DCL' OR 'SDC' INTERRUPT? 03121
4346	238E	1328					JEQ DCLSTOP	YES, PROCESS THE DEVICE CLEAR 03122
4347	2390	10EF					JMP R0STST	NO, CHECK FOR MORE GPIB INTERRUPTS 03123
4348								
4349	2392	2320	3382			NOTCMD	COB GET,R12	IS IT A GROUP EXECUTE TRIGGER INTERRUPT? 03124
4350	2396	1362					JEQ GETRUN	YES, THIS IS A 'RUN' COMMAND 03125
4351								
4352	2398	2320	3386				COB B0,R12	IS IT AN OUTPUT INTERRUPT? 03126
4353	239C	1368					JEQ BYTEOUT	YES, PROCESS OUTGOING DATA BYTE 03127
4354	239E	0820	092A	092A			MOV SE0I,SE0I	DID LAST OUTPUT FINISH? 03128
4355	23A4	1303					JEQ *+8	03129
4356	23A6	0820	3320	E07C			MOV3 SPACEB,R7W	NO, REPLACE LAST BYTE IN MC68488 WITH A ' ' 03130
4357	23AC	04E0	092A				CLR SE0I	NO, CLEAR LAST BYTE OUT FLAG 03131
4358								
4359	23B0	2320	3378				COB BI,R12	IS IT AN INPUT INTERRUPT? 03132
4360	23B4	136A					JEQ BYTEIN	YES, PROCESS INCOMING DATA BYTE 03133
4361								
4362	23B6	0320	E054			SPASIN	MOV3 R5R,R12	GET CURRENT SERVICE REQUEST STATUS 03134
4363	23BA	2320	3386				COB SRQS,R12	IS 7854 IN SERVICE REQUEST STATE? 03135
4364	23BE	1310					JEQ RQSON	YES, DON'T CHANGE RQS STATUS 03136
4365	23C0	0320	E044				MOV3 R1R,R12	NO, IS 7854 IS SERIAL POLL ACTIVE STATE? 03137
4366	23C4	2320	337C				COB SPAS,R12	03138
4367	23C8	13FB					JEQ *-8	YES, WAIT UNTIL IT IS NOT 03139
4368	23CA	0320	E054				MOVB R5R,R12	03140
4369	23CE	4320	3386				SZC RSV,R12	NO, RESET RSV BIT IS STATUS BYTE 03141
4370	23D2	080C	E074				MOV3 R12,R5W	03142
4371	23D6	04E0	DAE6				CLR RQSNJM	CLEAR SERVICE REQUEST NUMBER 03143

GPIB INTERRUPT HANDLER

03103

4372	23DA	C320	33CA	MOV	SRQLED,R12		03144	
4373	23DE	1E00		SBZ	0	TURN SRQ LED OFF	03145	
4374	23E0	C320	D92C	RSJON	MOV	GPIBR12,R12	RESTORE R12	03146
4375	23E4	0380		RTW ^D			03147	

(SELECTIVE) DEVICE CLEAR

03148

4377										
4378										
4379								'DCL' & 'SDC' DEVICE CLEAR GPIB COMMANDS		
4380										
4381								GPIB INTERRUPT ROUTINE		
4382										
4383								NOTE ---		
4384								A 'DCL' OR 'SDC' FUNCTION IDENTICALLY AS A 'STOP'		
4385								EXCEPT ANY GPIB TRANSFER IN PROGRESS IS ALSO		
4386								TERMINATED.		
4387										
4388										
4389	23E6	D320	E040					DCLSTOP MOVB R0R,R12	CLEAR PENDING GPIB INTERRUPTS	03149
4390	23EA	D320	E044					NOVB R1R,R12	CLEAR 'CMD' INTERRUPT	03150
4391	23EE	D820	D994	E068				NOVB AMREG,R2W	CLEAR ADDRESS MODE REGISTER	03151
4392	23F4	D4E0	E06C					CLR R3W	CLEAR AUXILIARY COMMAND REGISTER	03152
4393	23F8	D320	E05C					NOVB R7R,R12	CLEAR 'BYTE-IN' INTERRUPT	03153
4394	23FC	D820	3320	E07C				NOVB SPACEB,R7W	CLEAR 'BYTE-OUT' INTERRUPT	03154
4395	2402	D4E0	D92A					CLR SEOI	CLEAR LAST BYTE OUT FLAG	03155
4396	2406	8820	DAE5	3344				C RQSNUM,C6	IS A POWER-ON RQS PENDING?	03156
4397	240C	1306						JEQ KEEPQRS	YES, DON'T CLEAR IT	03157
4398	240E	8820	DAE5	3346				C RQSNJM,C7		03158
4399	2414	1302						JEQ KEEPQRS		03159
4400	2416	06A0	3304					BL RQSOFF	TURN OFF ANY SERVICE REQUESTS	03160
4401	241A	C320	D944					KEEPQRS MOV GPIBKEY,R12	IS A GPIB KEY IN PROGRESS?	03161
4402	241E	1306						JEQ NOGPIBK	NO, PASS 'STOP' TO SYSTEM CONTROL	03162
4403	2420	028C	0004					CI R12,4	IS A GPIB QUERY IN PROGRESS?	03163
4404	2424	1303						JEQ NOGPIBK		03164
4405	2426	E820	3382	D966				SJC STOPKEY,KEY	YES, ALLOW THE GPIB KEY TO BE STOPPED	03165
4406	242C	C820	3832	D962				NOGPIBK MOV PANIC,FKEY	LOAD 'STOP' KEYCODE	03166
4407	2432	D4E0	D928					CLR RENKEY	DISABLE OPC SRQ	03167
4408	2436	D4E0	D918					CLR CMDINDX	SET COMMAND BUFFER INDEX TO 0	03168
4409	243A	0720	D92E					SETJ STOPJCL	SET FLAG INDICATING THIS STOP IS A DCL	03169
4410	243E	D4E0	D944					CLR GPIBKEY	RESET QUERIE FLAG	03170
4411	2442	D4E0	D93A					CLR ACPTCMD	NO LONGER INPUTTING GPIB COMMANDS	03171
4412	2446	D420	0044					BLWP GPIBK	PASS 'STOP' TO SYSTEM CONTROL	03172
4413	244A	028C	0004					CI R12,4	IS A GPIB QUERY IN PROGRESS?	03173
4414	244E	1603						JNE NOQUER	NO, RETURN	03174
4415	2450	C347						MOV R7,R13	YES, RETURN TO ROUTINE INTERRUPTED	03175
4416	2452	C388						MOV R8,R14	BY QUERY	03176
4417	2454	C3C9						MOV R9,R15		03177
4418	2456	C320	D92C					NOQUER MOV GPIBR12,R12	RESTORE R12	03178
4419	245A	0380						RTWP		03179

```

4421 *****
4422 **
4423 ** 'GET' GROUP EXECUTE TRIGGER GPIB COMMAND **
4424 ** **
4425 ** GPIB INTERRUPT ROUTINE **
4426 ** **
4427 ** NOTE --- **
4428 ** A 'GET' FUNCTIONS IDENTICALLY TO A 'RUN' COMMAND **
4429 ** **
4430 *****
4431 245C D820 3380 E06C GETRUN MOVB DACR,R3W COMPLETE HANDSHAKE FOR 'GET' COMMAND 03181
4432 2462 2820 380E D952 MOV RUNKEY,FKEY MOVE KEYCODE FOR 'RUN' TO KEY BUFFER 03182
4433 2468 04E0 D918 CLR CMDINDX SET COMMAND BUFFER INDEX TO 0 03183
4434 246C 0420 0044 BLWP GPIBKY PASS 'RJR' COMMAND TO SYSTEM CONTROL 03184
4435 2470 0460 237J B ROSTST CHECK FOR MORE GPIB INTERRUPTS 03185
    
```

BYTE OUT

03186

```

4437 *****
4438 **
4439 ** GPIB DATA OUTPUT **
4440 **
4441 ** GPIB INTERRUPT ROUTINE **
4442 **
4443 ** NOTE --- **
4444 **
4445 ** A GPIB OUTPUT INTERRUPT OCCURS WHEN THE 7854 IS **
4446 ** ADDRESSED AS A TALKER AND AFTER EVERY COMPLETED **
4447 ** HANDSHAKE OF AN OUTPUT DATA TRANSMISSION. **
4448 ** A CHECK MUST ALSO BE MADE TO HANDLE THE LAST OUTPUT **
4449 ** DATA BYTE. ON THE LAST BYTE THE INTERNAL DATA ACCEPTED **
4450 ** FLAG IS NOT SET AND THE TRANSMISSION IS COMPLETE. **
4451 **
4452 *****
4453 2474 C320 D92A BYTEOUT MOV SEOI,R12 WAS OUTPUT JUST COMPLETED? 03187
4454 2478 04E0 D92A CLR SEOI 03188
4455 247C 1103 JLT LASTOUT NO, LEAVE OUTPUT FLAG ALONE 03189
4456 247E C820 333A D940 MOV C1,INOUT FLAG INDICATING GPIB OUTPUT OCCURING 03190
4457 2484 C320 D92C LASTOJT MOV GPIBR12,R12 RESTORE R12 03191
4458 2488 0380 RTW 03192
    
```

```

4460 *****
4461 **
4462 ** GPIB DATA INPUT **
4463 ** ** **
4464 ** GPIB INTERRUPT ROUTINE **
4465 ** ** **
4466 ** NOTE --- **
4467 ** THIS ROUTINE CANNOT DESTROY R7-R12 AS THESE **
4468 ** ARE USED TO SAVE THE RETURN ADDRESS FOR QUERIE **
4469 ** ROUTINES. **
4470 ** ** **
4471 *****
4472 248A C80C D930 BYTEIN MOV R12,INTGPIB SAVE GPIB INTERRUPT STATUS 03194
4473 248E C320 D93E MOV IGNORE,R12 SHOULD THE REST OF THIS INPUT BE IGNORED? 03195
4474 2492 1318 JEQ GPIBOK NO, PROCESS THIS DATA BYTE 03196
4475 * MOVB R7R,R12 YES, READ DATA BYTE 03197 DEL
4476 * 00001PATCH
4477 * PROBLEM #1 - PATCH #1 (1 OF 2) 00001PATCH
4478 * 00001PATCH
4479 * WITH GPIB TERMINATOR MODE OF EOI/LF, COMMANDS ARE NOT CORRECTLY 00001PATCH
4480 * TERMINATED. THE CHECK FOR LF BEING A TERMINATOR WAS NOT MADE. 00001PATCH
4481 * 00001PATCH
4482 2494 0460 9600 3 PATCH1 BRANCH TO PATCH #1 00001PATCH
4483 BACK1 EQU * DEFINE REENTRY POINT 00001PATCH
4484 * 00001PATCH
4485 * END OF PATCH #1 OF PROBLEM #1 00001PATCH
4486 * 00001PATCH
4487 2498 C320 D930 MOV INTGPIB,R12 RESTORE GPIB INTERRUPT STATUS 03198
4488 249C 2320 337A COC END,R12 CHECK FOR END OF TRANSMISSION 03199
4489 24A3 1606 JNE NOTEND 03200
4490 24A2 04E0 D93E CLR IGNORE START LOOKING AT INPUT WITH NEXT RECORD 03201
4491 24A6 04E0 D93A CLR ACPTCMD CLEAR ACCEPTING COMMANDS FLAG 03202
4492 24AA 04E0 D940 CLR INOUT GPIB IS NOW IDLE 03203
4493 24AE 04E0 D918 NOTEND CLR CMDINDX SET INDEX BACK TO ZERO FOR NEXT COMMAND 03204
4494 24B2 0820 D994 E058 MOVB AMREG,R2W RESET RFD HOLDOFF 03205
4495 24B8 0820 3386 E05C MOVB RFD,R3W COMPLETE HANDSHAKE STOPPED BY RFD HOLDOFF 03206
4496 24BE C320 D92C MOV GPIBR12,R12 RESTORE R12 03207
4497 24C2 0380 RTWP RETURN TO INTERRUPTED ROUTINE 03208
4498 *
4499 * INPUT COMMANDS ARE TO PROCESSED BY THIS ROUTINE. INPUT DATA
4500 * ARE TO BE PROCESSED BY OTHER ROUTINES.
4501 *
4502 24C4 0720 D940 GPIBOK SETB INOUT FLAG INDICATING GPIB INPUT OCCURING 03209
4503 24C8 C320 D944 MOV GPIBKEY,R12 SHOULD THIS BYTE BE USED BY 03210
4504 24CC 1302 JEQ COMMAND A GPIB INPUT ROUTINE? 03211
4505 24CE 1503 JGT COMMAND+4 NO, THEN DECODE AS A COMMAND 03212
4506 24D0 10EE JMP NOTEND 03213
4507 *
4508 * INPUT DATA UNTIL A COMMAND TERMINATOR IS RECEIVED
4509 *
4510 24D2 04E0 D940 COMMAND CLR INOUT FLAG INDICATING GPIB I/O IDLE 03214
4511 24D6 C320 D930 MOV INTGPIB,R12 03215
4512 24DA C0A0 D918 MOV CMDINDX,R2 NEXT AVAILABLE LOCATION IN BUFFER 03216
4513 24DE D0E0 D994 MOVB AMREG,R3 GET CURRENT ADDRESS MODE REGISTER INFO 03217
4514 24E2 E0E0 337C SOB HDLA,R3 ADD RFD HOLDOFF 03218
4515 24E6 0803 E058 MOVB R3,R2W HOLDOFF RFD ON ALL DATA 03219
4516 * MOVB R7R,R1 READ DATA BYTE 03220 DEL
4517 * 00002PATCH
4518 * PROBLEM #1 - PATCH #2 (2 OF 2) 00002PATCH
    
```

4519					*				00002PATCH
4520					*	WITH GPIB TERMINATOR MODE OF EOI/LF, COMMANDS ARE NOT CORRECTLY			00002PATCH
4521					*	TERMINATED. THE CHECK FOR LF BEING A TERMINATOR WAS NOT MADE.			00002PATCH
4522					*				00002PATCH
4523	24EA	0460	961C			B	PATCH2	BRANCH TO PATCH #2	00002PATCH
4524		24EE				BACK2	EQU *	DEFINE REENTRY POINT	00002PATCH
4525					*				00002PATCH
4526					*	END OF PATCH #2 OF PROBLEM #1			00002PATCH
4527					*				00002PATCH
4528	24EE	C0E0	D926			MOV	TOTLLO,R3	IS 7854 IN TALK-LISTEN MODE?	03221
4529	24F2	160B				JNE	TOLO	NO, THEN ALWAYS ACCEPT COMMANDS	03222
4530	24F4	D0E0	E044			MOV3	R1R,R3	READ COMMAND STATUS REGISTER	03223
4531	24F8	20E0	3386			COC	REM,R3	IS GPIB IN REMOTE MODE?	03224
4532	24FC	1306				JEQ	TOLO	NO, THEN ACCEPT LOCAL COMMANDS ONLY	03225
4533	24FE	2320	337A			COC	END,R12	DID <EOI> COME WITH THIS BYTE	03226
4534	2502	16D5				JNE	NOTEND	NO, CONTINUE ACCEPTING COMMANDS	03227
4535	2504	04E0	D93A			CLR	ACPTCMD	YES, CLEAR ACCEPTING COMMANDS FLAG	03228
4536	2508	10D2				JMP	NOTEND		03229
4537	250A	0720	D93A			TOLJ	SET0	ACPTCMD	SET FLAG INDICATING ACCEPTING GPIB COMMANDS
4538	250E	2060	3386			COC	CH4000,R1	IS THIS UPPER/LOWER CASE CHARACTER?	03231
4539	2512	1602				JNE	*+6	NO, LEAVE IT ALONE	03232
4540	2514	4060	3382			SZC	CH2000,R1	YES, SHIFT LOWER CASE TO UPPER CASE	03233
4541					*	LI	R4,DLMTBL	LOAD DELIMITER TABLE ADDRESS	03234 DEL
4542					*				00022PATCH
4543					*	PROBLEM #18 - PATCH #22 (1 OF 2)			00022PATCH
4544					*				00022PATCH
4545					*	CORRECT INCOMPATIBILITIES WITH TEK 4924 GPIB TAPE DRIVE			00022PATCH
4546					*				00022PATCH
4547	2518	0204	97E4			LI	R4,DLMTBL1	LOAD ALTERNATE DELIMETER TABLE ADDRESS	00022PATCH
4548					*				00022PATCH
4549					*	END OF PATCH #22 OF PROBLEM #18			00022PATCH
4550					*				00022PATCH
4551	251C	C174				MOV	*R4+,R5	COUNT OF ENTRIES IN TABLE	03235
4552	251E	9D01				CMDDLM	CB	R1,*R4+	IS THIS BYTE A DELIMITER?
4553	2520	1317				JEQ	CMDIN	YES, COMMAND IS COMPLETE	03237
4554	2522	0605				DEC	R5	NO, CHECK THE REST	03238
4555	2524	15FC				JGT	CMDDLM		03239
4556	2526	8820	D944	3340		C	GPIBKEY,C4	IS A QUERIE EXECUTING?	03240
4557	252C	1605				JNE	NOTQ		03241
4558	252E	C347				MOV	R7,R13	YES, ABORT IT	03242
4559	2530	C388				MOV	R8,R14		03243
4560	2532	C3C9				MOV	R9,R15		03244
4561	2534	04E0	D944			CLR	GPIBKEY		03245
4562	2538	D881	D90C			NOTJ	MOV3	R1,CMD3UF(R2)	MOVE NEXT CHARACTER TO BUFFER
4563	253C	0582				INC	R2	POINT TO NEXT EMPTY LOCATION	03247
4564	253E	0282	000A			CI	R2,CMD5Z	HAS MAXIMUM LENGTH HAS BEEN REACHED?	03248
4565	2542	1101				JLT	*+4	YES, THEN POINT BACK TO LAST BUFFER BYTE	03249
4566	2544	0602				DEC	R2	PUT REST OF INPUT INTO LAST BUFFER BYTE	03250
4567	2546	C802	D918			MOV	R2,CMDINDX		03251
4568	254A	2320	337A			COC	END,R12	CHECK IF EOI WITH THIS CHARACTER	03252
4569	254E	16B1				JNE	NOTEND+4		03253
4570	2550	2320	337A			CMDDIN	COC	END,R12	CHECK IF EOI WITH THIS CHARACTER
4571	2554	1602				JNE	*+6		03255
4572	2556	04E0	D93A			CLR	ACPTCMD	RESET FLAG INDICATING ACCEPTING GPIB COMMANDS	03256
4573	255A	C0A0	D918			MOV	CMDINDX,R2	GET SIZE OF INPUT MNEMONIC	03257
4574	255E	1603				JNE	NOTTRM	IF SIZE EQUAL 0 THEN TERMINATOR ONLY	03258
4575	2560	04E0	D940			CLR	INOUT	FLAG INDICATING GPIB I/O IDLE	03259
4576	2564	10A4				JMP	NOTEND		03260
4577	2566	D8A0	3338	D90C		NOTTRM	MOV3	C0,CMD3UF(R2)	TERMINATE COMMAND WITH \$00

BYTE IN

03193

4578	256C	04E0	D918	CLR	CMDINDX	SET INDEX BACK TO ZERO FOR NEXT COMMAND	03262
4579	2570	04C3		CLR	R3	INITIALIZE KEY CODE	03263
4580	2572	0700		SETO	R0	SET DUPLICATE CODE TO NEGATIVE	03264
4581	2574	04CA		CLR	R10	ZERO COUNT OF MATCHES	03265
4582	2576	C103		CMDSRCH	MOV R3,R4	CREATE INDEX INTO KEYTAB	03266
4583	2578	0A24		SLA	R4,2		03267
4584	257A	05C4		INCT	R4	MNU ADDRESS = KEYCODE * 4 + 2	03268
4585	257C	C124	3836	MOV	KEYTAB(R4),R4	ADDRESS OF MNEMONIC	03269
4586	2580	C142		MOV	R2,R5	INPUT MNEMONIC SIZE	03270
4587	2582	0206	D90C	LI	R6,CMDBUF	ADDRESS OF INPUT MNEMONIC	03271
4588	2586	90B4		CMPRCMD	CB *R4+,*R6+	CHECK IF MNEMONICS ARE THE SAME	03272
4589	2588	1607		JNE	NXTCMD		03273
4590	258A	0605		DEC	R5	CHECK ALL INPUT CHARACTERS	03274
4591	258C	15FC		JGT	CMPRCMD		03275
4592	258E	9814	3338	CB	*R4,C0	DOES COMMAND MATCH EXACTLY	03276
4593	2592	133B		JEQ	FNDCMD	IF SO, THEN THIS IS THE KEYCODE	03277
4594	2594	C003		MOV	R3,R0	SAVE THIS CODE	03278
4595	2596	058A		INC	R10	INCREMENT MATCH COUNT	03279
4596	2598	0583		NXTCMD	INC R3	INCREMENT TO NEXT KEYCODE	03280
4597	259A	0283	007F	CI	R3,\$7F	SKIP LAST BAD GPIB INPUT	03281
4598	259E	1309		JEQ	SKIPKEY		03282
4599	25A0	0283	0000	CI	R3,\$00	SKIP CALCULATOR 'PROGRAM' KEY	03283
4600	25A4	1306		JEQ	SKIPKEY		03284
4601	25A6	0283	0080	CI	R3,\$80	SKIP CALCULATOR 'EXECUTE' KEY	03285
4602	25AA	1303		JEQ	SKIPKEY		03286
4603	25AC	0283	0053	CI	R3,\$53	SKIP CALCULATOR 'NEXT' KEY	03287
4604	25B0	1601		JNE	SKIPKEY+2		03288
4605	25B2	0583		SKIPKEY	INC R3		03289
4606	25B4	0283	0100	CI	R3,\$0100	NOT FOUND IF NO MORE TO SEARCH	03290
4607	25B8	11DE		JLT	CMDSRCH		03291
4608	25BA	060A		DEC	R10	HOW MANY MATCHES WERE FOUND?	03292
4609	25BC	1517		JGT	BADGKEY	2 OR MORE IS ERROR	03293
4610	25BE	1102		JLT	*+6	0, THEN CHECK FOR QUERY	03294
4611	25C0	C0C0		MOV	R0,R3	YES, USE IT	03295
4612	25C2	1023		JMP	FNDCMD		03296
4613				*			
4614				*		IF INPUT IS NOT A 7854 COMMAND, CHECK IF IT IS A	
4615				*		7854 QUERY.	
4616				*			
4617	25C4	0204	3000	LI	R4,QUETABL	ADDRESS OF VALID 7854 QUERIES	03297
4618	25C8	C142		QJESRCH	MOV R2,R5	COUNT OF GPIB INPUT CHARACTERS	03298
4619	25CA	0206	D90C	LI	R6,CMDBUF	ADDRESS OF INPUT GPIB DATA	03299
4620	25CE	9036		CMPRQUE	CB *R6+,*R4+	DOES INPUT MATCH THIS VALID QUERY?	03300
4621	25D0	1603		JNE	NXTQJE	NO, THEN CHECK THE NEXT ONE	03301
4622	25D2	0605		DEC	R5	ARE THERE MORE CHARACTERS TO CHECK?	03302
4623	25D4	15FC		JGT	CMPRQUE	YES, CONTINUE	03303
4624	25D6	103A		JMP	FNDQUE	NO, PROCESS THE QUERY	03304
4625	25D8	9834	3338	NXTQJE	CB *R4+,C0	FIND ADDRESS OF NEXT QUERY NAME	03305
4626	25DC	16FD		JNE	*-4		03306
4627	25DE	9834	3338	CB	*R4+,C0		03307
4628	25E2	13FD		JEQ	*-4		03308
4629	25E4	0584		INC	R4		03309
4630	25E6	0284	3016	CI	R4,ENDQTBL	IS THIS THE END OF THE QUERY TABLE?	03310
4631	25EA	1AEE		JL	QUESRCH	NO, CONTINUE COMPARISONS	03311
4632				*			
4633				*		IF NOT A VALID COMMAND OR QUERY, THEN LOAD THE KEYCODE	
4634				*		FOR AN INVALID GPIB COMMAND.	
4635				*			
4636	25EC	0203	007F	BADGKEY	LI R3,\$7F		03312

BYTE IN

03193

4637	25F0	C820	D90C	D934	MOV	CMDBUF,MNUGPIB	MOVE FIRST THREE CHARACTERS OF MNEMONIC	03313
4638	25F6	D4E0	D936		CLR	MNUGPIB+2	TO MNUGPIB FOR ERROR DISPLAY. TERMINATE WITH	03314
4639	25FA	D820	D90E	D936	MOVB	CMDBUF+2,MNUGPIB+2	BYTE \$00	03315
4640	2600	2320	337A		CJC	END,R12	SET IGNORE IF THIS IS NOT THE END	03316
4641	2604	1302			JEQ	FND CMD	OF THE RECORD	03317
4642	2606	D720	D93E		SETJ	IGNORE		03318
4643	268A	D300	0007		FND CMD	LIMI	7	03319
4644	260E	C0A0	D964		MOV	BUFAVL,R2	GET NUMBER OF BUFFER SPOTS AVAILABLE	03320
4645	2612	1503			JGT	PASSCMD	IF THERE ARE SOME, SEND GPIB COMMAND	03321
4646	2614	D720	D992		SETJ	SUSGPIB	IF NONE, SUSPEND GPIB WITH NRFD	03322
4647	2618	D380			RTWP			03323
4648	261A	C803	D962		PASSCMD	MOV	R3,FKEY	03324
4649	261E	1602			JNE	*+6		03325
4650	2620	D5A8	D962		INC	FKEY	IF 'STOP' 0 MAKE IT 'STOP' 1	03326
4651	2624	8820	D962	333A	C	FKEY,C1	IS THIS A 'STOP' COMMAND?	03327
4652	262A	1307			JEQ	SUSSTOP	YES, SUSPEND GPIB UNTIL 'STOP' IS COMPLETED	03328
4653	262C	D820	D994	E068	MOV3	AMREG,R2W	RESET RFD HOLDOFF	03329
4654	2632	D820	3386	E06C	MOV3	RFDR,R3W	COMPLETE HANDSHAKE STOPPED BY RFD HOLDOFF	03330
4655	2638	1002			JMP	SUSSTOP+4		03331
4656	263A	D720	D93C		SUSSTOP	SETJ	STPGPIB	03332
4657	263E	D720	D928		SETJ	REMKEY	SET FLAG INDICATING REMOTE COMMAND	03333
4658	2642	D420	0044		BLWP	GPIBKY	PROCESS THIS KEYCODE	03334
4659	2646	D380			RTWP			03335
4660								
4661	2648		D360		RESSPIB	WORD	WPGPIB	03336
4662	264A		268A		WORD	FND CMD	GPIB WORKSPACE	03337
4663					*			
4664					*		IF THIS IS A 7854 QUERY THEN OUTPUT REQUESTED DATA	
4665					*			
4666	264C	D820	D994	E068	FNDQUE	MOV3	AMREG,R2W	03338
4667	2652	D820	3386	E06C		MOV3	RFDR,R3W	03339
4668	2658	C060	D926			MOV	TOTLLO,R1	03340
4669	265C	1524				JGT	NOQUE	03341
4670	265E	9834	3338			CB	*R4+,C0	03342
4671	2662	16FD				JNE	*-4	03343
4672	2664	9834	3338			CB	*R4+,C0	03344
4673	2668	13FD				JEQ	*-4	03345
4674	266A	D604				DEC	R4	03346
4675	266C	C314				MOV	*R4,R12	03347
4676	266E	D4E0	D948			CLR	INOUT	03348
4677	2672	8820	3340	D944		C	C4,GPIBKEY	03349
4678	2678	1306				JEQ	QUEWAIT	03350
4679	267A	C1C0				MOV	R13,R7	03351
4680	267C	C20E				MOV	R14,R8	03352
4681	267E	C24F				MOV	R15,R9	03353
4682	2680	C820	3340	D944		MOV	C4,GPIBKEY	03354
4683	2686	D300	000F		QUEWAIT	LIMI	\$F	03355
4684	268A	C060	D940			MOV	INOUT,R1	03356
4685	268E	13FB				JEQ	QUEWAIT	03357
4686	2690	1101				JLT	ABORTQ	03358
4687	2692	D69C				BL	*R12	03359
4688	2694	D300	0007		ABORTQ	LIMI	7	03360
4689	2698	D4E0	D944			CLR	GPIBKEY	03361
4690	269C	C347				MOV	R7,R13	03362
4691	269E	C388				MOV	R8,R14	03363
4692	26A0	C3C9				MOV	R9,R15	03364
4693	26A2	D4E0	D93A			CLR	ACPTCMD	03365
4694	26A6	D380			NOQUE	RTWP		03366

4696	26A8	4820	32A2	D91E	KEYRQSOFF	SZC	RQSENBL,RSVENBL	ALL SERVICE REQUESTS OFF	03368
4697	26AE	0380				RTWP			03369
4698	26B0	E820	32A2	D91E	KEYRQSON	SOC	RQSENBL,RSVENBL	ALL SERVICE REQUESTS ON	03370
4699	26B6	0380				RTWP			03371
4700	26B8	4820	32AC	D91E	KEYOPCOFF	SZC	OPCENBL,RSVENBL	OPC SERVICE REQUESTS OFF	03372
4701	26BE	0380				RTWP			03373
4702	26C0	E820	32AC	D91E	KEYOPCON	SOC	OPCENBL,RSVENBL	OPC SERVICE REQUESTS ON	03374
4703	26C6	0380				RTWP			03375
4704	26C8	4820	32AA	D91E	KEYCEROFF	SZC	CERENBL,RSVENBL	CER SERVICE REQUESTS OFF	03376
4705	26CE	0380				RTWP			03377
4706	26D0	E820	32AA	D91E	KEYCERON	SOC	CERENBL,RSVENBL	CER SERVICE REQUESTS ON	03378
4707	26D6	0380				RTWP			03379
4708	26D8	4820	32AB	D91E	KEYEXROFF	SZC	EXRENBL,RSVENBL	EXR SERVICE REQUESTS OFF	03380
4709	26DE	0380				RTWP			03381
4710	26E0	E820	32AB	D91E	KEYEXRON	SOC	EXRENBL,RSVENBL	EXR SERVICE REQUESTS ON	03382
4711	26E6	0380				RTWP			03383
4712	26E8	4820	32A5	D91E	KEYREMOFF	SZC	REMENBL,RSVENBL	REM SERVICE REQUESTS OFF	03384
4713	26EE	0380				RTWP			03385
4714	26F0	E820	32A6	D91E	KEYREMON	SOC	REMENBL,RSVENBL	REM SERVICE REQUESTS ON	03386
4715	26F6	0380				RTWP			03387
4716	26F8	4820	32A4	D91E	KEYIOCOFF	SZC	IOCENBL,RSVENBL	IOC SERVICE REQUESTS OFF	03388
4717	26FE	0380				RTWP			03389
4718	2700	E820	32A4	D91E	KEYIOCON	SOC	IOCENBL,RSVENBL	IOC SERVICE REQUESTS ON	03390
4719	2706	0380				RTWP			03391

KEY 'READX'

03392

```

4721 *****
4722 **
4723 ** GPIB KEY 'READX' **
4724 ** **
4725 ** LEVEL 1 ROUTINE **
4726 ** **
4727 ** INPUT: DATA (WFM OR CNS) FROM GPIB **
4728 ** OUTPUT: DATA (WFM OR CNS) TO X **
4729 **
4730 *****
4731 2708 C060 D992 KEYREADX MOV SUSGPIB,R1 IS GPIB SUSPENDED IN COMMAND INPUT? 03393
4732 270C 1119 JLT READERR YES, ERROR (PREVENT GPIB FROM HANGING) 03394
4733 270E C060 D93A MOV ACPTCMD,R1 IS A COMMAND MESSAGE STILL COMING? 03395
4734 2712 11FA JLT KEYREADX YES, WAIT FOR IT TO COMPLETE 03396
4735 2714 C820 3334 D944 MOV CN2,GPIBKEY FLAG INDICATING 'READX' KEY IS EXECUTING 03397
4736 271A 0720 D972 SETO RDTFLAG 03398
4737 271E 0420 1430 BLWP FROOUT UPDATE CALCULATOR READOUT TO CURRENT DISPLAY 03399
4738 2722 C820 334E DAE6 MOV C11,RQSNUM LOAD RQS NUMBER FOR 'READX' KEY 03400
4739 2728 06A0 3016 BL STRLSTN START GPIB LISTEN FUNCTION 03401
4740 272C 04E0 DA0C CLR YZFLAG SET FLAG INDICATING NO YZERO INPUT 03402
4741 2730 06A0 3124 NXTRBYT BL GPIBIN INPUT NEXT 'READX' DATA BYTE FROM GPIB 03403
4742 2734 06A0 3112 BL CHKOLM IS THIS BYTE A DELIMITER? 03404
4743 2738 1606 JNE TRYPRE NO, BEGIN PARSING INPUT 03405
4744 273A 06A0 3206 BL CHKEOI YES, IGNORE IT 03406
4745 273E 16F8 JNE NXTRBYT 03407
4746 2740 04E0 D94A READERR CLR FATAL ERROR FLAG 03408
4747 2744 1068 JMP STPREAD 03409
4748
4749 ***** FIRST CHECK FOR 'WFMPRE' TO START PREAMBLE *****
4750
4751 2746 C820 333A D91C TRYPRE MOV C1,CNSWFM FLAG INDICATING WAVEFORM BEING INPUT 03410
4752 274C 0200 2F76 LI R0,WFMPRE ADDRESS OF 'WFMPRE' TABLE ENTRY 03411
4753 2750 C080 MOV *R0+,R2 BYTE COUNT OF STRING 03412
4754 2752 0602 DEC R2 03413
4755 2754 1005 JMP FRSTPRE 03414
4756 2756 06A0 3206 NXTPRE BL CHKEOI DID <EOI> COME WITH THIS BYTE? 03415
4757 275A 13F2 JEQ READERR YES, CHECK FOR 'WFMPRE' 03416
4758 275C 06A0 3124 BL GPIBIN INPUT NEXT DATA BYTE FROM GPIB 03417
4759 2760 9C01 FRSTPRE CB R1,*R0+ COMPARE INPUT AGAINST 'WFMPRE' 03418
4760 2762 1603 JNE TRYCRV IF NO MATCH, TRY 'CURVE' 03419
4761 2764 0602 DEC R2 DECREMENT BYTE COUNT 03420
4762 2766 15F7 JGT NXTPRE 03421
4763 2768 1060 JMP IWFMPRE INPUT PREAMBLE 03422
4764
4765 ***** SECOND CHECK FOR 'CURVE' TO START WAVEFORM *****
4766
4767 *TRYCRV C R2,WFMPRE DID ANY CHARACTERS MATCH 'WFMPRE' ? 03423 DEL
4768 * 00016PATCH
4769 * PROBLEM #13 - PATCH #16 (1 OF 1) 00016PATCH
4770 * 00016PATCH
4771 * 'READX' WILL NOT ACCEPT ONLY A WAVEFORM CURVE (NO PREAMBLE) 00016PATCH
4772 * 00016PATCH
4773 276A 0460 9710 TRYCRV B PATC#15 BRANCH TO PATCH16 00016PATCH
4774 276E 275E BACK16 EQU * DEFINE REENTRY POINT 00016PATCH
4775 * 00016PATCH
4776 * END OF PROBLEM #13 00016PATCH
4777 * 00016PATCH
4778 276E 1114 JLT READCNS YES, CANNOT MATCH 'CURVE' 03424
4779 2770 0200 2FF2 LI R0,CURVE ADDRESS OF 'CURVE' TABLE ENTRY 03425

```

4780	2774	C0B9		MOV	*R0+,R2	BYTE COUNT OF STRING	03426	
4781	2776	0602		DEC	R2		03427	
4782	2778	1005		JMP	FRSTCRV		03428	
4783	277A	06A0	3206	NXTCRV	BL	CHKEDI	DID <EDI> COME WITH THIS BYTE?	03429
4784	277E	13E0		JEQ	READERR	YES, CHECK FOR 'CURVE'	03430	
4785	2780	06A0	3124		BL	GPIBIN	INPUT NEXT DATA BYTE FROM GPIB	03431
4786	2784	9C01		FRSTCRV	CB	R1,*R0+	COMPARE INPUT AGAINST 'CURVE'	03432
4787	2786	1608		JNE	READCNS	IF NO MATCH, MUST BE CONSTANT	03433	
4788	2788	0602		DEC	R2	DECREMENT BYTE COUNT	03434	
4789	278A	15F7		JGT	NXTCRV		03435	
4790	278C	0420	69A8	BLWP	OPWH2W0	COPY HEADER FROM OPW TO W0	03436	
4791	2790	06A0	6974	BL	ZEROW0	SET W0 TO 0	03437	
4792	2794	0460	2980	B	ICURVE	INPUT CURVE	03438	
4793								
4794				***** THIRD CHECK FOR CONSTANT INPUT *****				
4795								
4796	2798	04C3		READCNS	CLR	R3	03439	
4797	279A	1007		JMP	FRSTDGT		03440	
4798	279C	0983		NXTDGT	SRL	R3,8	SHIFT LAST BYTE TO LOW BYTE	03441
4799	279E	00C1		MOVW	R1,R3	MOVE THIS BYTE TO HIGH BYTE	03442	
4800	27A0	06A0	3206	BL	CHKEDI	DID <EDI> COME WITH THIS BYTE	03443	
4801	27A4	13CD		JEQ	READERR	YES, THIS IS AN ERROR	03444	
4802	27A6	06A0	3124		BL	GPIBIN	GET NEXT GPIB DATA BYTE	03445
4803	27AA	9801	3797	FRSTDGT	CB	R1,ASCII0	A VALID CONSTANT MUST CONTAIN A DIGIT	03446
4804	27AE	11F6		JLT	NOTDGT		03447	
4805	27B0	9801	379E	CB	R1,ASCII9		03448	
4806	27B4	15F3		JGT	NOTDGT		03449	
4807	27B6	0720	091C	SET0	CNSWFM	FLAG INDICATING CONSTANT BEING INPUT	03450	
4808	27BA	C020	33E0	MOV	GPIBBUF,R0	BUFFER FOR GPIB INPUT	03451	
4809	27BE	DC20	3781	MOVW	PLUS,*R0+	ASSUME VALUE IS POSITIVE	03452	
4810	27C2	9803	372F	CB	R3,DECPNT	HAS PREVIOUS BYTE A DECIMAL POINT	03453	
4811	27C6	1602		JNE	*+6	NO, CHECK SIGN	03454	
4812	27C8	DC03		MOVW	R3,*R0+	YES, ADD IT TO BYTE STRING	03455	
4813	27CA	06C3		SWPB	R3		03456	
4814	27CC	9803	3782	CB	R3,MINUS	IS VALUE ACTUALLY NEGATIVE?	03457	
4815	27D0	1604		JNE	ISPOS	NO, POSITIVE WAS ALREADY ASSUMED	03458	
4816	27D2	C0A0	33E0	MOV	GPIBBUF,R2		03459	
4817	27D6	04A0	3782	MOVW	MINUS,*R2	YES, REPLACE PLUS WITH MINUS	03460	
4818	27DA	0202	00FA	ISPJS	LI	R2,250	MAXIMUM GPIB BYTES ACCEPTED	03461
4819	27DE	DC01		NXTDGT	MOVW	R1,*R0+	MOVE NUMBER INTO BUFFER	03462
4820	27E0	0602		JEC	R2		03463	
4821	27E2	1587		JGT	NOTFULL		03464	
4822	27E4	0720	093E	SET0	IGNORE	IGNORE THE REST IF MORE THAN 255	03465	
4823	27E8	0060	E05C	MOVW	R7R,R1		03466	
4824	27EC	04E0	094C	CLR	WARNING		03467	
4825	27F0	1006		JMP	FPISIN		03468	
4826	27F2	06A0	3206	NOTFULL	BL	CHKEDI	DID <EDI> COME WITH THIS DATA BYTE?	03469
4827	27F6	1303		JEQ	FPISIN	YES, CONVERT TO INTERNAL REPRESENTATION	03470	
4828	27F8	06A0	3124		BL	GPIBIN	GET NEXT GPIB DATA BYTE	03471
4829	27FC	10F0		JMP	NXTDGT		03472	
4830	27FE	0420	3321	FPISIN	MOVW	COMHAB,*R0	ENSURE THAT A CONSTANT DELIMITER IS PRESENT	03473
4831	2802	C020	33E0	MOV	GPIBBUF,R0	BUFFER WHICH CONTAINS NUMBER	03474	
4832	2806	0420	6518	BLWP	SCANIN	CONVERT NUMBER TO INTERNAL REPRESENTATION	03475	
4833	280A	04C0		CLR	R0	SET FLAG INDICATING FLOATING POINT NUMBER	03476	
4834	280C	1005		JMP	PSHRVAL		03477	
4835	280E	04E0	094C	W=MINW	CLR	WARNING	SET WARNING FLAG	03478
4836	2812	0700		W=MIN	SET0	R0	FLAG INDICATING WFM #	03479
4837	2814	04C1		CLR	R1		03480	
4838	2816	04C2		CLR	R2	PUSH '0 WFM' ONTO USERSTACK	03481	

4839	2818	06A0	6962	PSHRVAL	BL	PSHREG	PUSH NUMBER INPUT ONTO USERSTACK	03482
4840	281C	04E0	091C	STPREAD	CLR	CNSH=M	FLAG INDICATING NEITHER WAVEFORM OR CONSTANT	03483
4841	2820	04E0	0944		CLR	GPIBKEY	FLAG INDICATING GPIB KEY IS FINISHED	03484
4842	2824	0720	0972		SETJ	RDTFLAG	UPDATE ENTIRE CALCULATOR READOUT	03485
4843	2828	0380			RTWP			03486
4844								
4845				*****		INPUT WAVEFORM PREAMBLE		*****
4846								
4847	282A	0420	69A8	INFMPRE	BLWP	OPWH2W0	COPY HEADER FROM OPW TO W0	03487
4848	282E	06A0	6974		BL	ZEROW0	SET W0 TO ALL 0'S	03488
4849	2832	06A0	3206		BL	CHKEJI	IS 'WFMPRE' ENTIRE MESSAGE	03489
4850	2836	13E0			JEQ	WFMIN	YES, WAVEFORM IS DEFAULT	03490
4851	2838	C020	33E0	INLABEL	MOV	GPIBBUF,R0	ADDRESS OF BUFFER FOR GPIB INPUT	03491
4852	283C	06A0	3124	L3LC1R	BL	GPIBIN	GET NEXT GPIB CHARACTER	03492
4853	2840	06A0	3206		BL	CHKEOI	DID <EOI> COME WITH THIS DATA BYTE?	03493
4854	2844	1604			JNE	CHKFDLM	NO, CHECK IF IT IS A DELIMITER	03494
4855	2846	8020	33E0		C	GPIBBUF,R0		03495
4856	284A	13E3			JEQ	WFMIN		03496
4857	284C	10E0			JMP	WFMINW	WARNING IS STRAY LAST BYTES	03497
4858								
4859	284E	06A0	3112	CHKFDLM	BL	CHKDLM	IS THE A LABEL DELIMITER?	03498
4860	2852	1302			JEQ	LABELIN	YES, LABEL IS IN	03499
4861	2854	DC01			MOVB	R1,*R0+	MOVE INTO GPIB INPUT BUFFER	03500
4862	2856	10F2			JMP	L3LC1R	GET NEXT LABEL CHARACTER	03501
4863	2858	8020	33E0	LABELIN	C	GPIBBUF,R0	CHECK IF JUST DELIMITER	03502
4864	285C	13EF			JEQ	L3LC1R	IF SO, IGNORE IT	03503
4865	285E	DC01			MOVB	R1,*R0+	ADD DELIMITER TO LABEL	03504
4866	2860	D420	3338		MOVB	C0,*R0	END INPUT WITH A BYTE \$00	03505
4867	2864	0201	2F75		LI	R1,WFMI0	ADDRESS OF PREAMBLE TABLE	03506
4868	2868	C020	33E0	NXTLBL	MOV	GPIBBUF,R0	ADDRESS OF INPUT LABEL	03507
4869	286C	C0B1			MOV	*R1+,R2	BYTE COUNT OF STRING TO MATCH	03508
4870	286E	1313			JEQ	BADL3L	IF AT END OF TABLE, NO MATCH	03509
4871	2870	9C70		CHKL3L	CB	*R0+,*R1+	CHECK FOR A LABEL MATCH	03510
4872	2872	160B			JNE	NOTLBL	IF NO MATCH TRY NEXT LABEL	03511
4873	2874	0602			DEC	R2		03512
4874	2876	15FC			JGT	CHKL3L		03513
4875	2878	0501			INC	R1		03514
4876	287A	0241	FFFE		ANDI	R1,\$FFFE	POINT TO ADDRESS OF LABEL ROUTINE	03515
4877	287E	C051			MOV	*R1,R1	GET ADDRESS OF LABEL ROUTINE	03516
4878	2880	0691			BL	*R1	INPUT ARGUMENT FOR THIS LABEL	03517
4879	2882	06A0	3206		BL	CHKEJI	DID <EOI> COME WITH LAST BYTE?	03518
4880	2886	13C5			JEQ	WFMIN	YES, INPUT IS FINISHED	03519
4881	2888	10D7			JMP	INLABEL	INPUT NEXT LABEL	03520
4882	288A	A042		NOTL3L	A	R2,R1	POINT TO NEXT LABEL TO CHECK	03521
4883	288C	0221	0004		AI	R1,4		03522
4884	2890	0241	FFFE		ANDI	R1,\$FFFE		03523
4885	2894	10E9			JMP	NXTL3L	TRY NEXT LABEL	03524
4886	2896	04E0	094C	BADL3L	CLR	WARNING	SET WARNING FLAG	03525
4887	289A	10CE			JMP	INLABEL	INPUT NEXT LABEL	03526

```

4889 *****
4890 **
4891 ** DECODE WAVEFORM PREAMBLE ARGUMENTS **
4892 ** **
4893 ** LEVEL 5A ROUTINE (SPECIAL - SEE GPIB NOTES ABOVE) **
4894 ** **
4895 *****
4896
4897
4898 ***** DECODE 'ENCDG' - WAVEFORM ENCODING *****
4899
4900 IENCDG MOV R11,R10 SAVE RETURN ADDRESS 03528
4901 BL INSTR INPUT WAVEFORM PREAMBLE ARGUMENT STRING 03529
4902 MOV GPIBBUF,R0 ADDRESS OF INPUT WPA VALUE 03530
4903 LI R1,ASC ADDRESS OF STRING 'ASC' 03531
4904 LI R2,3 LENGTH OF STRING 'ASC' 03532
4905 CB *R0+,*R1+ COMPARE INPUT VALUE TO STRING 03533
4906 JNE BADVAL ERROR IF STRINGS DON'T MATCH 03534
4907 DEC R2 COMPARE ONLY FIRST THREE CHARACTERS 03535
4908 JGT *-6 NOTE - (ASC... WILL MATCH AS CORRECT) 03536
4909 JMP **6 03537
4910 BADVAL CLR FATAL SET ERROR FLAG IF STRING DOESN'T MATCH 03538
4911 B *R10 03539
4912
4913 ***** DECODE 'NR.PT' - NUMBER OF POINTS/WAVEFORM *****
4914
4915 INR.PT MOV R11,R10 SAVE RETURN ADDRESS 03540
4916 BL INVAL INPUT WAVEFORM PREAMBLE ARGUMENT VALUE 03541
4917 BL FP2INT CONVERT FLOATING POINT NUMBER TO INTEGER 03542
4918 C R1,RESOLV DOES NR.PT MATCHES CURRENT POINTS/WAVEFORM? 03543
4919 JEQ **6 03544
4920 CLR WARNING SET FLAG INDICATING BAD NR.PT 03545
4921 B *R10 03546
4922
4923 ***** DECODE 'PT.FMT' - POINT FORMAT *****
4924
4925 IPT.FMT MOV R11,R10 SAVE RETURN ADDRESS 03547
4926 BL INSTR INPUT WAVEFORM PREAMBLE ARGUMENT STRING 03548
4927 MOV GPIBBUF,R0 ADDRESS OF GPIB I/O BUFFER 03549
4928 CB *R0,Y FIRST CHARACTER MUST BE A 'Y' 03550
4929 JEQ **6 03551
4930 CLR FATAL IF NOT, THEN ERROR 03552
4931 B *R10 03553
4932
4933 ***** DECODE 'XZERO' - X ZERO *****
4934
4935 IXZERO MOV R11,R10 SAVE RETURN ADDRESS 03554
4936 BL INVAL INPUT WAVEFORM PREAMBLE ARGUMENT VALUE 03555
4937 BL FP2INT CONVERT X ZERO TO INTEGER 03556
4938 JEQ **6 NON ZERO VALUE IS ERROR 03557
4939 CLR FATAL 03558
4940 B *R10 03559
4941
4942 ***** DECODE 'XINCR' - X INCREMENT *****
4943
4944 IXINCR MOV R11,R10 SAVE RETURN ADDRESS 03560
4945 BL INVAL INPUT WAVEFORM PREAMBLE ARGUMENT VALUE 03561
4946 JLT BADVAL VALUE MUST BE POSITIVE 03562
4947 JEQ BADVAL 03563
    
```

DECODE WAVEFORM PREAMBLE

03527

4948	2906	C020	DA24	MOV	W0HEAD,R0	ADDRESS OF WFM #0'S HEADER	03564
4949	290A	A020	3340	A	HEXP,R0	POINT TO HEXP IN HEADER	03565
4950	290E	C001		MOV	R1,*R0+	MOVE NEW HEXP TO HEADER	03566
4951	2910	C402		MOV	R2,*R0		03567
4952	2912	045A		B	*R10		03568
4953							
4954				*****	DECODE 'XUNIT' - X UNIT	*****	
4955							
4956	2914	C20B		IXUNIT	MOV R11,R10	SAVE RETURN ADDRESS	03569
4957	2916	C0A0	3350		MOV HSCALD,R2	HORIZONTAL SCALE UNIT OFFSET	03570
4958	291A	A0A0	DAD4	IXYJNIT	A W0HEAD,R2	ADD BASE ADDRESS OF HEADER	03571
4959	291E	06A0	3210		BL INSTR	INPUT WAVEFORM PREAMBLE ARGUMENT STRING	03572
4960	2922	0201	2020		LI R1,\$2020	DEFAULT UNIT IS ' '	03573
4961	2926	C020	33E0		MOV GPIBBUF,R0	ADDRESS OF GPIB I/O BUFFER	03574
4962	292A	9810	3321		CB *R0,COMNAB	IS FIRST CHARACTER A ','?	03575
4963	292E	1304			JEQ NOUNIT	YES, THEN NO UNIT WAS SENT	03576
4964	2930	D070			MOV3 *R0+,R1	NO, USE THIS CHARACTER AS THE UNIT	03577
4965	2932	9810	3321		CB *R0,COMNAB	ARE THERE MORE THAN ONE CHARACTERS?	03578
4966	2936	1302			JEQ *+6	YES, THIS IS A WARNING	03579
4967	2938	04E0	D94C	NOUNIT	CLR WARNING	SET WARNING FLAG IF NOT ONE ALPHA	03580
4968	293C	C401			MOV R1,*R2	MOVE NEW UNIT TO WFM 0'S HEADER	03581
4969	293E	045A			B *R10		03582
4970							
4971				*****	DECODE 'YMULT' - Y MULTIPLIER	*****	
4972							
4973	2940	C20B		IYMULT	MOV R11,R10	SAVE RETURN ADDRESS	03583
4974	2942	C260	0012		MOV 16(R13),SOFT	GET CURRENT SOFTSTACK POINTER	03584
4975	2946	06A0	3234		BL INVAL	INPUT WAVEFORM PREAMBLE ARGUMENT VALUE	03585
4976	294A	1185			JLT BADVAL	VALUE MUST BE POSITIVE	03586
4977	294C	13B5			JEQ BADVAL		03587
4978	294E	0649			DECT SOFT		03588
4979	2950	C642			MOV R2,*SOFT	PUSH NEW VERTICAL SCALE ONTO SOFTSTACK	03589
4980	2952	0649			DECT SOFT		03590
4981	2954	C641			MOV R1,*SOFT		03591
4982	2956	0649			DECT SOFT		03592
4983	2958	0409			CLR *SOFT	WFM #0 IS TO RECEIVE VERTICAL SCALE	03593
4984	295A	0420	6EE0		BLWP NEWVSC	PUT NEW VERTICAL SCALE INTO WFM 0'S HEADER	03594
4985	295E	045A			B *R10		03595
4986							
4987				*****	DECODE 'YZERO' - Y ZERO	*****	
4988							
4989	2960	C20B		IYZERO	MOV R11,R10	SAVE RETURN ADDRESS	03596
4990	2962	C260	0012		MOV 16(R13),SOFT	GET CURRENT SOFTSTACK POINTER	03597
4991	2966	06A0	3234		BL INVAL	INPUT WAVEFORM PREAMBLE ARGUMENT VALUE	03598
4992	296A	C801	DA0E		MOV R1,YZ1	SAVE Y ZERO	03599
4993	296E	C802	DA10		MOV R2,YZ2	THIS WILL BE USED IN 'ICURVE' ROUTINE	03600
4994	2972	0720	DA0C		SET0 YZFLAG	SET FLAG INDICATING Y ZERO HAS BEEN INPJT	03601
4995	2976	045A			B *R10		03602
4996							
4997				*****	DECODE 'YUNIT' - Y UNIT	*****	
4998							
4999	2978	C20B		IYUNIT	MOV R11,R10	SAVE RETURN ADDRESS	03603
5000	297A	C0A0	334C		MOV VSCALD,R2	VERTICAL SCALE UNIT OFFSET	03604
5001	297E	10C0			JMP IXYUNIT		03605

```

5003 *****
5004 **
5005 **      DECODE WAVEFORM
5006 **
5007 **      LEVEL 5A ROUTINE (SPECIAL - SEE GPIB NOTES ABOVE)
5008 **
5009 *****
5010
5011 ***** DECODE 'CURVE' - SWITCH TO CURVE INPUT *****
5012
5013 2980 C020 DA0C   ICURVE MOV  YZFLAG,R0   WAS THERE A Y ZERO INPUT?      03607
5014 2984 131A      JEQ  INCURVE   NO, THEN RETURN                03608
5015 2986 0649      DECT  SOFT
5016 2988 C650 DA10   MOV  YZ2,*SOFT   PUSH VERTICAL ZERO ONTO SOFTSTACK 03609
5017 298C 0649      DECT  SOFT
5018 298E C660 DA0E   MOV  YZ1,*SOFT
5019 2992 C0A0 DAD4   MOV  W0HEAD,R2  ADDRESS OF WFM #0'S HEADER      03613
5020 2996 A0A0 3338   A      VEXP,R2   POINT TO VEXP IN HEADER         03614
5021 299A C072      MOV  *R2+,R1    MOVE VEXP INTO R1,R2           03615
5022 299C C092      MOV  *R2,R2
5023 299E 0420 7046   BLWP FPDIVZ     COMPUTE BINARY VERTICAL ZERO     03617
5024 29A2 1903      JNO  GOODVZR
5025 29A4 04E0 D94A   CLR  FATAL      IF OVERFLOW (<-20,>+20) THEN ERROR 03619
5026 29A8 1008      JMP  INCURVE
5027 29AA C060 DAD4   GOODVZR MOV W0HEAD,R1  ADDRESS OF WFM #0'S HEADER      03621
5028 29AE A060 3348   A      VOFFAB,R1 POINT TO VOFFAB IN HEADER       03622
5029 29B2 C479      MOV  *SOFT+,*R1 MOVE NEW VERTICAL ZERO TO WFM 0'S HEADER 03623
5030 29B4 C0D1      MOV  *R1,R3     MIN IS DEFAULTED TO VERTICAL ZERO 03624
5031 29B6 0503      NEG  R3
5032 29B8 C103      MOV  R3,R4     MAX IS DEFAULTED TO VERTICAL ZERO 03626
5033 29BA C160 DAD2   INCJRVE MOV W0ADD,R5   ADDRESS OF W0'S DATA          03627
5034 29BE C1A0 D970   MOV  RESOLV,R6 CURRENT WAVEFORM RESOLUTION     03628
5035 29C2 1003      JMP  FRSTPNT
5036 29C4 06A0 3206   GPNTIN BL  CHKEOI   DID <EJI> COME WITH THIS DATA BYTE? 03630
5037 29C8 1321      JEQ  NOTRES
5038 29CA 06A0 3234   FRSTPNT BL  INVAL   INPUT NEXT GPIB WAVEFORM POINT    03632
5039 29CE 0649      DECT  SOFT
5040 29D0 C642      MOV  R2,*SOFT   PUSH POINT ONTO SOFTSTACK       03634
5041 29D2 0649      DECT  SOFT
5042 29D4 C641      MOV  R1,*SOFT
5043 29D6 0201 5000   LI   R1,$5000   LOAD A 20 INTO R1,R2           03637
5044 29DA 0202 0005   LI   R2,$0005
5045 29DE 0420 7046   BLWP FPDIVZ     CONVERT TO DATA SCREEN VALUE    03639
5046 29E2 1902      JNO  **+6
5047 29E4 04E0 D94A   CLR  FATAL      OVERFLOW IF <-20 OR >+20 DIVS     03641
5048 29E8 C1F9      MOV  *SOFT+,*R7 MOVE POINT TO WAVEFORM          03642
5049 29EA C047      MOV  R7,*R5+
5050 29EC 80C7      C      R7,R3     CHECK FOR NEW MIN              03644
5051 29EE 1501      JGT  **+4
5052 29F0 C0C7      MOV  R7,R3
5053 29F2 8107      C      R7,R4     CHECK FOR NEW MAX              03647
5054 29F4 1101      JLT  **+4
5055 29F6 C107      MOV  R7,R4
5056 29F8 05C9      INCT  SOFT
5057 29FA 0606      DEC  R6
5058 29FC 15E3      JGT  GPNTIN     INPUT RESOLUTION POINTS        03652
5059 29FE 06A0 3206   BL  CHKEJI     DID <EJI> COME WITH THIS DATA BYTE? 03653
5060 2A02 1306      JEQ  RESIN
5061          *      SETO  IGNORE   IF NOT, IGNORE THE REST        03655 DEL
    
```


5062				*				00003PATCH
5063				*	PROBLEM #2 - PATCH #3 (1 OF 1)			00003PATCH
5064				*				00003PATCH
5065				*	'READX' DOESN'T TERMINATE WAVEFORM INPUT CORRECTLY. ALL			00003PATCH
5066				*	TRAILING DELIMITERS SHOULD BE IGNORED.			00003PATCH
5067				*				00003PATCH
5068	2A04	0460	963C		B	PATCH3	BRANCH TO PATCH #3	00003PATCH
5069		2A08		BACK3	EQU	*	DEFINE REENTRY POINT	00003PATCH
5070				*				00003PATCH
5071				*	END OF PROBLEM #2			00003PATCH
5072				*				00003PATCH
5073	2A08	0050	E05C		MOVB	R7R,R1	READ DATA BYTE TO CONTINUE TRANSFER	03656
5074	2A0C	04E0	D94C	NOTRES	CLR	WARNING		03657
5075	2A10	5103		RESIN	S	R3,R4	CHECK VERTICAL DATA RANGE RESTRICTIONS	03658
5076	2A12	1902			JNO	*+6	FOR INPUT WAVEFORM	03659
5077	2A14	0460	3052		B	ZEROWFM	IF VIOLATION, THEN ERROR	03660
5078	2A18	0460	2812		B	WFMIN		03661

5080					*****		
5081					**		**
5082					**	GPIB KEY '>TEXT'	**
5083					**		**
5084					**	LEVEL 1 ROUTINE	**
5085					**		**
5086					**	INPUT: TEXT FROM GPIB TO CRT	**
5087					**	OUTPUT: NONE	**
5088					**		**
5089					*****		
5090	2A1C	C050	D992		KEY2TEXT	MOV SUSGPIB,R1	IS GPIB SUSPENDED IN COMMAND INPUT? 03663
5091	2A20	112F			JLT	TEXTERR	YES, ERROR (PREVENT GPIB FROM HANGING) 03664
5092	2A22	C050	D93A		MOV	ACPTCMD,R1	IS A COMMAND MESSAGE STILL COMING? 03665
5093	2A26	11FA			JLT	KEY2TEXT	YES, WAIT FOR IT TO COMPLETE 03666
5094	2A28	C020	3336	D944	MOV	CM1,GPIBKEY	FLAG INDICATING '>TEXT' KEY IS EXECUTING 03667
5095	2A2E	06A0	12BE		BL	CLRTXT	CLEAR CURRENT DISPLAY TEXT 03668
5096	2A32	0720	D972		SETJ	RDTFLAG	03669
5097	2A36	0420	1430		3LWP	FRDOJT	DISPLAY 4-LINE READOUT ONLY 03670
5098	2A3A	C020	3352	DAE6	MOV	C13,RQSNUM	LOAD RQS NUMBER FOR '>TEXT' KEY 03671
5099	2A40	06A0	3016		3L	STRLSTN	START GPIB LISTEN FUNCTION 03672
5100	2A44	0649			DECT	SOFT	03673
5101	2A46	C660	333E		MOV	C3,*SOFT	SET CURSOR TO LINE #3 CHARACTER #1 03674
5102	2A4A	0649			DECT	SOFT	03675
5103	2A4C	C660	333A		MOV	C1,*SOFT	03676
5104	2A50	06A0	3124		NEXTIN	3L GPIBIN	GET NEXT READOUT CHARACTER 03677
5105	2A54	9801	3320		CB	R1,SPACEB	CHECK IF THE IS A PRINTABLE CHARACTER 03678
5106	2A58	1115			JLT	CTRLIN	IF NOT, PROCESS THE CONTROL CHARACTER 03679
5107	2A5A	0649			TEXTIN	DECT	SOFT 03680
5108	2A5C	0901			SRL	R1,8	PUT CHARACTER IN LOW BYTE FOR 'TEXTS' 03681
5109	2A5E	1309			JEQ	CHKTEOI	DON'T ALLOW BYTE \$00 03682
5110	2A60	C641			MOV	R1,*SOFT	PUSH CHARACTER ONTO SOFTSTACK 03683
5111	2A62	0649			DECT	SOFT	03684
5112	2A64	C660	333A		MOV	C1,*SOFT	PUSH COUNT OF 1 ONTO SOFTSTACK 03685
5113	2A68	06A0	1362		BL	TEXTS	PUT THIS CHARACTER INTO THE READOUT DISPLAY 03686
5114	2A6C	1902			JND	CHKTEOI	DID IT FIT? 03687
5115	2A6E	04E0	D94C		WRN2TXT	CLR WARNING	NO, THIS IS A WARNING 03688
5116	2A72	04C5			CHKTEOI	CLR R5	SET FLAG INDICATING NOT <ESC> 03689
5117	2A74	06A0	3206		BL	CHKEOI	DID <EOI> COME WITH THIS DATA BYTE? 03690
5118	2A78	16EB			JNE	NEXTIN	IF NOT, GET MORE 03691
5119	2A7A	04E0	D944		RTN2TXT	CLR GPIBKEY	FLAG INDICATING GPIB IS FINISHED 03692
5120	2A7E	0300			RTWP		03693
5121							
5122	2A80	0450	2740		TEXTERR	B READERR	ABORT '>TEXT' 03694
5123							
5124	2A84	C145			CTRLIN	MOV R5,R5	HAS CONTROL CHARACTER PRECEDED BY AN <ESC>? 03695
5125	2A86	11E9			JLT	TEXTIN	YES, IT IS VALID CHARACTER 03696
5126	2A88	9801	331E		CB	R1,CRB	CHECK FOR A <CR> 03697
5127	2A8C	130B			JEQ	CRIN	03698
5128	2A8E	9801	331F		CB	R1,LFB	CHECK FOR A <LF> 03699
5129	2A92	13EF			JEQ	CHKTEOI	03700
5130	2A94	9801	3325		CB	R1,BELLB	CHECK FOR A <BEL> 03701
5131	2A98	1317			JEQ	BELLIN	03702
5132	2A9A	9801	3325		CB	R1,ESC3	CHECK FOR A <ESC> 03703
5133	2A9E	16E7			JNE	WRN2TXT	03704
5134	2AA0	0705			SETJ	R5	SET FLAG INDICATING <ESC> IN 03705
5135	2AA2	10E8			JMP	CHKTEOI+2	03706
5136	2AA4	C660	333A		CRIV	MOV C1,*SOFT+	SET CHARACTER # TO 1 03707
5137	2AA8	0599			INC	*SOFT	INCREMENT LINE # 03708
5138	2AAA	8819	3354		C	*SOFT,C14	MAXIMUM LINE # IS 14 03709

5139	2AAE	1802		JH	**6			03710
5140	2AB0	0649		DECT	SOFT			03711
5141	2AB2	10DF		JMP	CHKTEOI			03712
5142	2AB4	06A0	3206	BL	CHKEOI	DID <EOI> COME WITH THIS BYTE		03713
5143	2AB8	13E0		JEQ	RTN2TXT	YES, NO WARNING		03714
5144				SETO	IGNORE	IGNORE REST OF GPIB INPUT		03715 DEL
5145								00004PATCH
5146						PROBLEM #3 - PATCH #4 (1 OF 1)		00004PATCH
5147								00004PATCH
5148						'>TEXT' DOESN'T TERMINATE TEXT INPUT CORRECTLY. ALL		00004PATCH
5149						TRAILING DELIMITERS SHOULD BE IGNORED.		00004PATCH
5150								00004PATCH
5151	2ABA	0460	9558	B	PATCH4	BRANCH TO PATCH #4		00004PATCH
5152		2ABE		BACK4	EQU	DEFINE REENTRY POINT		00004PATCH
5153								00004PATCH
5154						END OF PROBLEM #2		00004PATCH
5155								00004PATCH
5156	2ABE	D060	E05C	MOV3	R7R,R1			03716
5157	2AC2	04E0	D94C	CLR	WARNING	SET WARNING FLAG		03717
5158	2AC6	10D9		JMP	RTN2TXT			03718
5159	2AC8	0420	6CF4	BELLIN	BLAP	BUZZIT	RING BELL	03719
5160	2ACC	10D2		JMP	CHKTEOI			03720

```

5162 *****
5163 **
5164 ** GPIB KEY 'TEXT' **
5165 ** ** **
5166 ** LEVEL 1 ROUTINE **
5167 ** ** **
5168 ** INPUT: NONE **
5169 ** OUTPUT: TEXT ON CRT TO GPIB **
5170 ** ** **
5171 *****
5172 *KEYTEXT MOV C1,GPIBKEY FLAG INDICATING 'TEXT' KEY EXECUTING 03722 DEL
5173 * ** ** 00013PATCH
5174 * PROBLEM #12 - PATCH #13 (1 OF 3) 00013PATCH
5175 * ** ** 00013PATCH
5176 * 7854 WON'T ACCEPT MULTIPLE OUTPUT COMMANDS IN ONE MESSAGE 00013PATCH
5177 * ** ** 00013PATCH
5178 2ACE 0460 96C2 KEYTEXT B PATCH13 BRANCH TO PATCH #13 00013PATCH
5179 2A02 BACK13 EQU * DEFINE REENTRY POINT 00013PATCH
5180 * ** ** 00013PATCH
5181 * END OF PATCH #13 OF PROBLEM #12 00013PATCH
5182 * ** ** 00013PATCH
5183 2AD2 1000 JMP *+002 00013PATCH
5184 2AD4 0720 D972 SETO ROTFLAG 03723
5185 2AD8 0420 1430 BLWP FRDOUT UPDATE READOUT TO CURRENT DISPLAY 03724
5186 2ADC C820 3350 DAE6 MOV C12,RQSNUM LOAD RQS NUMBER FOR 'TEXT' KEY 03725
5187 2AE2 06A0 3098 BL STRTALK START GPIB TALK FUNCTION 03726
5188 2AE6 C0A0 DADE MOV DISMEM,R2 STARTING ADDRESS OF CURRENT READOUT 03727
5189 2AEA 0206 2336 LI R6,STPTXT ADDRESS TO CONTINUE IN THIS ROUTINE 03728
5190 *****
5191 * ** **
5192 * THE FOLLOWING CODE IS SHARED BY THE GPIB QUERY ROUTINE *
5193 * 'SRQ?'. FOR THIS REASON SEVERAL RESTRICTIONS HAVE TO BE *
5194 * FOLLOWED BY THIS BLOCK OF CODE. ONLY REGISTERS R1-R5 *
5195 * MAY BE USED AS ALL OTHERS CONTAIN SAVED INFORMATION. R6 *
5196 * CONTAINS THE RETURN ADDRESS TO WHICH THIS BLOCK BRANCHES *
5197 * WHEN FINISHED. *
5198 * ** **
5199 2AEE D072 NKTTEXT MOV3 *R2+,R1 GET NEXT READOUT CHARACTER * 03729
5200 2AF0 110D JLT FRMTOUT BIT 15 = 1 IMPLIES FORMAT CHARACTER * 03730
5201 2AF2 9801 3320 CB R1,SPACEB CHECK IF THIS IS A CONTROL CHARACTER * 03731
5202 2AF6 1103 JLT CTRLOUT * 03732
5203 2AF8 06A0 3154 TEXTOUT BL GPIBOUT OUTPUT CONTROL CHARACTER * 03733
5204 2AFC 10F8 JMP NXTTEXT * 03734
5205 2AFE C0C1 CTRLOUT MOV R1,R3 SAVE CONTROL CHARACTER * 03735
5206 2B00 D060 3325 MOVB ESCB,R1 * 03736
5207 2B04 06A0 3154 BL GPIBOUT FIRST OUTPUT AN <ESC> * 03737
5208 2B08 C043 MOV R3,R1 NEXT OUTPUT CHARACTER * 03738
5209 2B0A 10F6 JMP TEXTOUT * 03739
5210 2B0C 9801 33AC FRMTOUT CB R1,NULL CHECK FOR A <NULL> * 03740
5211 2B10 1309 JEQ NULLOUT * 03741
5212 2B12 9801 33B1 CB R1,CRLF CHECK FOR A <CRLF> * 03742
5213 2B16 1309 JEQ CRLFOUT * 03743
5214 2B18 9801 33AD CB R1,ETX CHECK FOR AN <ETX> * 03744
5215 2B1C 16E8 JNE NXTTEXT * 03745
5216 2B1E 06A0 3195 ETXOUT BL TERMINATE SEND GPIB OUTPUT TERMINATION * 03746
5217 2B22 0456 B *R6 RETURN TO APPROPRIATE ROUTINE * 03747
5218 2B24 D050 3320 NJLLOJT MOV3 SPACEB,R1 CONVERT <NULL> TO A SPACE * 03748
5219 2B28 10E7 JMP TEXTOUT * 03749
5220 2B2A 9812 33AD CRLFOUT CB *R2,ETX IS THIS THE LAST <CRLF> * 03750
    
```

5221	2B2E	13F7		JEQ	ETXOUT	YES, OUTPUT GPIB TERMINATOR	*	03751
5222	2B30	D060	331E	MOV3	CRB,R1	CONVERT <CRLF> TO A <CR>	*	03752
5223	2B34	10E1		JMP	TEXTOUT		*	03753
5224				*			*	
5225				*	END OF RESTRICTED BLOCK (SEE ABOVE)		*	
5226				*			*	
5227					*****			
5228	2B36	04E0	D944	STPTXT CLR	GPIBKEY	FLAG INDICATING GPIB KEY IS FINISHED		03754
5229	2B3A	0380		RTWP				03755

5231										
5232									**	
5233						**	GPIB KEY 'SENDX'		**	
5234						**			**	
5235						**	LEVEL 1 ROUTINE		**	
5236						**			**	
5237						**	INPUT: NONE		**	
5238						**	OUTPUT: (X), WFM OR FP, TO GPIB		**	
5239						**			**	
5240						**	NOTE ---		**	
5241						**	THIS KEY SENDS THE CONTENTS OF THE X REGISTER OF THE		**	
5242						**	USER'S STACK OUT ON THE GPIB.		**	
5243						**			**	
5244						**			**	
5245						*	KEYSENDX MOV C2,GPIBKEY	FLAG INDICATING 'SENDX' KEY EXECUTING		03757 DEL
5246						*				00014PATCH
5247						*	PROBLEM #12 - PATCH #14 (2 OF 3)			00014PATCH
5248						*				00014PATCH
5249						*	7954 WON'T ACCEPT MULTIPLE OUTPUT COMMANDS IN ONE MESSAGE			00014PATCH
5250						*				00014PATCH
5251	2B3C	0460	96DC				KEYSENDX B PATCH14	BRANCH TO PATCH #14		00014PATCH
5252			2B40				BACK14 EQU *	DEFINE REENTRY POINT		00014PATCH
5253						*				00014PATCH
5254						*	END OF PATCH #14 OF PROBLEM #12			00014PATCH
5255						*				00014PATCH
5256	2B40	1000					JMP *+002			00014PATCH
5257	2B42	0720	0972				SETD RDTFLAG			03758
5258	2B46	0420	1430				BLWP FRDOUT	FORCE READOUT TO BE CURRENT		03759
5259	2B4A	C820	334C	DAE5			MOV C10,RQSNUM	LOAD RQS NUMBER FOR 'SENDX' KEY		03760
5260	2B50	06A0	3098				BL STRTALK	START GPIB TALK FUNCTION		03761
5261	2B54	06A0	690E				BL POPREG	POP X OFF USERSTACK		03762
5262	2B58	1607					JNE SENDWFM			03763
5263	2B5A	C820	332E	0920	SENDJNS		MOV NR3,GFORMAT	OUTPUT CONSTANTS IN NR3 FORMAT		03764
5264	2B60	C6A0	3270				BL OUTVAL	OUTPUT (X)		03765
5265	2B64	0460	2C7E				B STPSEND	STOP 'SENDX'		03766
5266										
5267	2B68	C820	332E	0920	SENDWFM		MOV NR3,GFORMAT	OUTPUT PREAMBLE VALUES IN NR3 FORMAT		03767
5268	2B6E	0200	2F75				LI R0,WFMIO	ADDRESS OF WFM I/O CONTROL TABLE		03768
5269	2B72	C0B0					OUTLBL MOV *R0+,R2	COUNT OF CHARACTERS IN LABEL		03769
5270	2B74	06A0	3262				BL OUTSTR	OUTPUT LABEL STRING		03770
5271	2B78	2020	333A				COC C1,R0	WAS STRING SIZE ODD?		03771
5272	2B7C	1601					JNE *+4	IF EVEN, POSITION OK		03772
5273	2B7E	0580					INC R0	IF ODD, INCREMENT POSITION		03773
5274	2B80	05C0					INCF R0	POINT TO ADDRESS OF VALUE ROUTINE		03774
5275	2B82	C070					MOV *R0+,R1	ADDRESS OF VALUE ROUTINE		03775
5276	2B84	C0C0					MOV R0,R3	SAVE CONTROL TABLE INDEX		03776
5277	2B86	0691					BL *R1	OUTPUT VALUE WHICH GOES WITH LABEL		03777
5278	2B88	C003					MOV R3,R0	RESTORE CONTROL TABLE INDEX		03778
5279	2B8A	10F3					JMP OUTLBL			03779

5281					*****		
5282					**		**
5283					**	CONSTRUCT WAVEFORM PREAMBLE ARGUMENTS	**
5284					**		**
5285					**	LEVEL 5A ROUTINE (SPECIAL - SEE GPIB NOTES ABOVE)	**
5286					**		**
5287					*****		
5288							
5289	288C	045B			OWFMPRE B	*R11	WAVEFORM PREAMBLE HEADER (NO VALUE) 03781
5290							
5291	288E	C28B			OENCDG	MOV R11,R10	SAVE RETURN ADDRESS 03782
5292	2890	0200	3329			LI R0,ASC	ADDRESS OF STRING 'ASC' 03783
5293	2894	0202	0003			LI R2,3	BYTE COUNT IN STRING 03784
5294	2898	06A0	3262		STROUT	BL OUTSTR	OUTPUT ASCII STRING 03785
5295	289C	0060	3321		COMOUT	MOV3 COMAB,R1	DELIMIT VALUES WITH A ',' 03786
5296	28A0	06A0	3154			3L GPIBOUT	03787
5297	28A4	045A				B *R10	03788
5298							
5299	28A6	C28B			ONR.PT	MOV R11,R10	SAVE RETURN ADDRESS 03789
5300	28A8	C050	0970			MOV RESOLV,R1	CURRENT WAVEFORM RESOLUTION (P/W) 03790
5301	28AC	06A0	757E			BL INT2FP	CONVERT TO FLOATING POINT 03791
5302	28B0	06A0	3270		VALOUT	BL OUTVAL	OUTPUT VALUE STRING 03792
5303	28B4	10F3				JMP COMOUT	DELIMIT WITH A ',' 03793
5304							
5305	28B6	C28B			OPT.FMT	MOV R11,R10	SAVE RETURN ADDRESS 03794
5306	28B8	0200	3328			LI R0,Y	ADDRESS OF STRING 'Y' 03795
5307	28BC	0202	0001			LI R2,1	BYTE COUNT IN STRING 03796
5308	28C0	10E3				JMP STROUT	03797
5309							
5310	28C2	C28B			OXZERO	MOV R11,R10	SAVE RETURN ADDRESS 03798
5311	28C4	0200	3327			LI R0,0	ADDRESS OF STRING 'C' 03799
5312	28C8	0202	0001			LI R2,1	BYTE COUNT IN STRING 03800
5313	28CC	10E5				JMP STROUT	03801
5314							
5315	28CE	C28B			OXINCR	MOV R11,R10	SAVE RETURN ADDRESS 03802
5316	28D0	C0A0	095E			MOV OPWF4H,R2	ADDRESS OF OPW'S HEADER 03803
5317	28D4	A0A0	3340			A HEXP,R2	ADDRESS OF HORIZONTAL INCREMENT 03804
5318	28D8	C072				MOV *R2+,R1	03805
5319	28DA	C092				MOV *R2,R2	LOAD HORIZONTAL INCREMENT 03806
5320	28DC	10E9				JMP VALOUT	03807
5321							
5322	28DE	C28B			OXUNIT	MOV R11,R10	SAVE RETURN ADDRESS 03808
5323	28E0	C0A0	095E			MOV OPWF4H,R2	ADDRESS OF OPW'S HEADER 03809
5324	28E4	A0A0	3350			A HSCALD,R2	ADDRESS OF HORIZONTAL UNITS 03810
5325	28E8	C052				MOV *R2,R1	HORIZONTAL UNIT 03811
5326	28EA	06A0	3154			BL GPIBOUT	OUTPUT HORIZONTAL UNIT 03812
5327	28EE	10D6				JMP COMOUT	03813
5328							
5329	28F0	C28B			OYZERO	MOV R11,R10	SAVE RETURN ADDRESS 03814
5330	28F2	C0A0	095E			MOV OPWF4H,R2	ADDRESS OF OPW'S HEADER 03815
5331	28F6	A0A0	3348			A VOFFA3,R2	ADDRESS OF VERTICAL OFFSET 03816
5332	28FA	C072				MOV *R2+,R1	VERTICAL OFFSET 03817
5333	28FC	C0A0	095E			MOV OPWF4H,R2	03818
5334	2C00	A0A0	3338			A VEXP,R2	ADDRESS OF VERTICAL SCALE FACTOR 03819
5335	2C04	0229	FFFC			AI SOFT,-4	03820
5336	2C08	CE72				MOV *R2+,*SOFT+	PUSH VEXP ONTO SOFTSTACK 03821
5337	2C0A	C652				MOV *R2,*SOFT	03822
5338	2C0C	0649				DECF SOFT	03823
5339	2C0E	04C2				CLR R2	03824

5340	2C10	0420	70AA	BLWP	FPMPY	ACTUAL VALUE OF VERTICAL OFFSET	03825
5341	2C14	C079		MOV	*SOFT+,R1		03826
5342	2C16	C0B9		MOV	*SOFT+,R2		03827
5343	2C18	10CB		JMP	VALOUT		03828
5344							
5345	2C1A	C28B		OVMULT	MOV R11,R10	SAVE RETURN ADDRESS	03829
5346	2C1C	0420	6814	BLWP	VRTSCL	OPW'S VERTICAL SCALE	03830
5347	2C20	10C7		JMP	VALOUT		03831
5348							
5349	2C22	C28B		OYUNIT	MOV R11,R10	SAVE RETURN ADDRESS	03832
5350	2C24	C0A0	D95E	MOV	OPWFMH,R2	ADDRESS OF OPW'S HEADER	03833
5351	2C28	A0A0	334C	A	VSCALE,R2	ADDRESS OF VERTICAL UNITS	03834
5352	2C2C	C052		MOV	*R2,R1	VERTICAL UNIT	03835
5353	2C2E	06A0	3154	BL	GPIBOUT	OUTPUT VERTICAL UNIT	03836
5354	2C32	D060	3322	MOV	SEMIB,R1	DELIMIT LAST VALUE WITH A ';'	03837
5355	2C36	06A0	3154	BL	GPIBOUT		03838
5356				* MOV	CRB,R1	ADD A <CR> AFTER THE ';'	03839 DEL
5357				*			00011PATCH
5358				*			00011PATCH
5359				*			00011PATCH
5360				*			00011PATCH
5361				*			00011PATCH
5362	2C3A	0460	96A6	B	PATCH11	BRANCH TO PATCH #11	00011PATCH
5363			2C3E	BACK11	EQU *	DEFINE REENTRY POINT	00011PATCH
5364				*			00011PATCH
5365				*			00011PATCH
5366				*			00011PATCH
5367	2C3E	06A0	3154	BL	GPIBOUT		03840
5368	2C42	045A		B	*R10		03841

```

5370 *****
5371 **
5372 **      DECODE WAVEFORM PREAMBLE ARGUMENTS      **
5373 **
5374 **      LEVEL 5A ROUTINE (SPECIAL - SEE GPIB NOTES ABOVE) **
5375 **
5376 *****
5377
    
```

5378	2C44	C820	3320	D920	OCURVE	MOV	NR2,GFORMAT	OUTPUT WAVEFORMS POINTS IN NR2 FORMAT	03843
5379	2C4A	C0E0	D95C			MOV	OPWFMD,R3	GET ADDRESS OF WAVEFORM DATA	03844
5380	2C4E	C120	D970			MOV	RESOLV,R4	GET CURRENT POINTS PER WAVEFORMS	03845
5381	2C52	0201	5000		NXTWPNT	LI	R1,\$5000	LOAD R1,R2 WITH A 20	03846
5382	2C56	0202	0005			LI	R2,\$0005		03847
5383	2C5A	0649				DECT	SOFT		03848
5384	2C5C	0409				CLR	*SOFT	PUT WAVEFORM POINT ON SOFTSTACK WITH AN	03849
5385	2C5E	0649				DECT	SOFT	EXPONENT OF 0	03850
5386	2C60	C673				MOV	*R3+,*SOFT		03851
5387	2C62	0420	70AA			BLW ^D	FPMPY	(WAVEFORM POINT) * 20 IS ACTUAL VALUE	03852
5388	2C66	C079				MOV	*SOFT+,R1	POP POINT VALUE OFF SOFTSTACK INTO R1,R2	03853
5389	2C68	C039				MOV	*SOFT+,R2		03854
5390	2C6A	06A0	3270			BL	OUTVAL	OUTPUT POINT VALUE	03855
5391	2C6E	0604				DEC	R4	OUTPUT RESOLUTION POINTS	03856
5392	2C70	1306				JEQ	STPSEND	DON'T ADD COMMA AFTER LAST POINT	03857
5393	2C72	1105				JLT	STPSEND		03858
5394	2C74	D050	3321			MOV3	COMMA,R1	SEPARATE WAVEFORMS POINTS BY COMMAS	03859
5395	2C78	06A0	3154			BL	GPIBOUT		03860
5396	2C7C	10EA				JMP	NXTWPNT		03861
5397	2C7E	06A0	3196		STPSEND	BL	TERMINATE	OUTPUT GPIB TERMINATOR	03862
5398	2C82	E820	3348	D972		SOC	CLINE16,RDTFLAG	UPDATE LINE #16 OF READOUT	03863
5399	2C88	04E0	D944			CLR	GPIBKEY	FLAG INDICATING GPIB IS FINISHED	03864
5400	2C8C	0380				RTW ^D			03865

5402					*****		
5403					**		**
5404					**	GPIB KEY 'SAVE'	**
5405					**		**
5406					**	LEVEL 1 ROUTINE	**
5407					**		**
5408					**	INPUT: NONE	**
5409					**	OUTPUT: PROGRAM TEXT OUTPUT TO GPIB	**
5410					**		**
5411					*****		
5412					*KEYSAVE MOV C3,GPIBKEY	FLAG INDICATING 'SAVE' KEY EXECUTING	03867 DEL
5413					*		00015PATCH
5414					* PROBLEM #12 - PATCH #15 (3 OF 3)		00015PATCH
5415					*		00015PATCH
5416					* 7854 WON'T ACCEPT MULTIPLE OUTPUT COMMANDS IN ONE MESSAGE		00015PATCH
5417					*		00015PATCH
5418	2C8E	0460	96F6		KEYSAVE B PATCH15	BRANCH TO PATCH #15	00015PATCH
5419			2C92		BACK15 EQU *	DEFINE REENTRY POINT	00015PATCH
5420					*		00015PATCH
5421					* END OF PATCH #15 OF PROBLEM #12		00015PATCH
5422					*		00015PATCH
5423	2C92	1000			JMP *+002		00015PATCH
5424	2C94	0720	0972		SETD ROTFLAG		03868
5425	2C98	0420	1430		BLWP FRDOUT	FORCE READOUT TO BE CURRENT	03869
5426	2C9C	C820	DAA8	DAA4	MOV PROGLN,LINENUM	IS THERE A PROGRAM TO SAVE?	03870
5427	2CA2	114C			JLT ERRSAVE	NO, THIS IS AN ERROR	03871
5428	2CA4	0420	4594		BLWP GETLINE	GET ADDRESS OF CURRENT LINE	03872
5429	2CA8	C0A0	DAA6		MOV LINEPNT,R2		03873
5430	2CAC	9812	33F2		CB *R2,ENDFLG	CURRENTLY AT END OF PROGRAM?	03874
5431	2CB0	1345			JEQ ERRSAVE	YES, ERROR (NOTHING TO SAVE)	03875
5432	2CB2	C820	334A	DAE6	MOV C9,RQSNUM	LOAD RQS NUMBER FOR 'SAVE' KEY	03876
5433	2CB8	06A0	3098		BL STRTALK	START GPIB TALK FUNCTION	03877
5434	2CBC	D0F2			NEXTMNU MOV3 *R2+,R3	GET NEXT KEY MNEMONIC	03878
5435	2CBE	C103			MOV R3,R4	SAVE KEY MNEMONIC	03879
5436	2CC0	9803	33F2		CB R3,ENDFLG	IS THIS THE END OF THE PROGRAM?	03880
5437	2CC4	1336			JEQ ENDPGRM	YES, OUTPUT TERMINATOR	03881
5438	2CC6	0983			SRL R3,8	MNEMONIC IN LOW BYTE, ZERO HIGH BYTE	03882
5439	2CC8	0A23			SLA R3,2		03883
5440	2CCA	05C3			INCF R3	MNEMONIC ADDR = KEYTAB(KEYCODE*4 + 2)	03884
5441	2CCC	C0E3	3836		MOV KEYTAB(R3),R3		03885
5442	2CD0	D073			OUTMNU MOV3 *R3+,R1	GET NEXT BYTE OF MNEMONIC	03886
5443	2CD2	1303			JEQ TRMNU	A BYTE OF \$00 TERMINATES MNEMONIC	03887
5444	2CD4	06A0	3154		BL GPIBOUT	OUTPUT THIS DATA BYTE TO GPIB	03888
5445	2CD8	10FB			JMP OUTMNU	YES, OUTPUT ALL BYTES	03889
5446	2CDA	D050	3320		TRMNU MOV3 SPACEB,R1	ADD A SPACE ' ' AFTER EACH MNEMONIC	03890
5447	2CDE	06A0	3154		BL GPIBOUT		03891
5448	2CE2	9804	37EF		CB R4,NXTKEYB	WAS THIS A 'NEXT' KEY?	03892
5449	2CE6	1304			JEQ ADDCR	YES, ADD <CR>	03893
5450	2CE8	9804	3805		CB R4,LNNKEYB	WAS THIS A 'LNN' KEY?	03894
5451	2CEC	1309			JEQ ADDLBL	YES, ADD LINE #	03895
5452	2CEE	10E6			JMP NEXTMNU		03896
5453	2CF0	9812	33F2		ADDCR CB *R2,ENDFLG	IS THIS THE END OF THE PROGRAM?	03897
5454	2CF4	131E			JEQ ENDPGRM	YES, OUTPUT TERMINATOR	03898
5455	2CF6	D060	331E		MOV3 CRB,R1	NO, OUTPUT A <CR>	03899
5456	2CFA	06A0	3154		BL GPIBOUT		03900
5457	2CFE	10DE			JMP NEXTMNU		03901
5458	2D00	D0F2			ADDLBL MOV3 *R2+,R3	GET LINE # (BCD IN HIGH BYTE OF R3)	03902
5459	2D02	C043			MOV R3,R1		03903
5460	2D04	0241	F000		ANDI R1,\$F000	KEEP TEN'S DIGIT	03904

5461	2008	0941		SRL	R1,4		03905
5462	200A	0221	3000	AI	R1,\$3000	CONVERT DIGIT TO ASCII	03906
5463	200E	06A0	3154	BL	GPIBOUT		03907
5464	2012	0060	3320	MOV3	SPACEB,R1	SEND A SPACE AFTER EACH DIGIT	03908
5465	2016	06A0	3154	BL	GPIBOUT		03909
5466	201A	C043		MOV	R3,R1		03910
5467	201C	0241	0F00	ANDI	R1,\$0F00	KEEP ONE'S DIGIT	03911
5468	2020	0221	3000	AI	R1,\$3000	CONVERT DIGIT TO ASCII	03912
5469	2024	06A0	3154	BL	GPIBOUT		03913
5470	2028	0060	3320	MOV3	SPACEB,R1	SEND A SPACE AFTER EACH DIGIT	03914
5471	202C	06A0	3154	BL	GPIBOUT		03915
5472	2030	10C5		JMP	NEXTMNU		03916
5473	2032	06A0	3196	ENDPRGM	BL	TERMINATE	03917
5474	2036	04E0	0944	STPSAVE	CLR	GPIBKEY	03918
5475	203A	0380			RTW ^D		03919
5476	203C	04E0	094A	ERRSAVE	CLR	FATAL	03920
5477	2040	10FA		JMP	STPSAVE	SET ERROR FLAG	03921

```

5479 *****
5480 **
5481 ** GPIB KEY 'ID' **
5482 ** **
5483 ** LEVEL 1 ROUTINE **
5484 ** **
5485 ** PUTS UP ID MESSAGE ON SCREEN **
5486 ** **
5487 ** - INSTRUMENT ID **
5488 ** - ROM AND PROM PATCH VERSION NUMBERS **
5489 ** - AMOUNT OF RAM IN THE SYSTEM **
5490 ** - IF THE GPIB BOARD IS IN THE SYSTEM THE FOLLOWING **
5491 ** INFORMATION IS ALSO DISPLAYED: **
5492 ** GPIB ADDRESS **
5493 ** MODE **
5494 ** TERMINATOR OPTION **
5495 ** SRQ OPTIONS ENABLED **
5496 ** **
5497 ** INPUT - NONE **
5498 ** OUTPUT - NONE **
5499 **
5500 *****
5501 KEYID EQU * 03923
5502 2042 0720 2042 D972 SETJ RDTFLAG 03924
5503 2046 0720 DACC SETJ OPROGLN REDISP_LAY PL# IF NECESSARY 03925
5504 204A 4820 336E D9CA SZC MW8KHZ,DSPRO 03926
5505 2050 06A0 129E BL CLRXT CLEAR THE SCREEN 03927
5506 2054 0649 DECT SOFT PUSH LINE NUMBER 03928
5507 2056 C660 334A MOV C9,*SOFT 03929
5508 205A 0649 DECT SOFT PUSH CHARACTER POSITION 03930
5509 205C C660 333A MOV C1,*SOFT 03931
5510 2060 0201 2EFE LI R1,IDMSG GET MESSAGE ADDRESS 03932
5511 2064 0221 0003 AI R1,3 SPACE PAST 'ID' 03933
5512 2068 C0A0 3356 MOV C15,R2 GET CHARACTER COUNT OF MSG 03934
5513 206C 06A0 2ECC BL SETUP 03935
5514 2070 0211 DA04 LI R1,VRSNUM GET MESSAGE ADDRESS 03936
5515 2074 C0A0 3348 MOV C8,R2 GET CHARACTER COUNT OF MSG 03937
5516 2078 06A0 2ECC BL SETUP 03938
5517 207C C020 D942 MOV GPIBOPT,R0 03939
5518 2080 1301 JEQ *+4 IF NO GPIB IN SYSTEM ... 03940
5519 2082 0380 RTWP ... PUT OUT ONE LINE ONLY 03941
5520 2084 05C9 INCT SOFT 03942
5521 2086 0599 INC *SOFT MOVE DOWN TO NEXT DISPLAY LINE 03943
5522 2088 0649 DECT SOFT 03944
5523 208A C660 333A MOV C1,*SOFT RESET CHAR POINTER TO FRONT OF LINE 03945
5524 208E 0201 2E68 LI R1,ASCADR GET MESSAGE ADDRESS 03946
5525 2092 C0A0 3348 MOV C8,R2 GET CHARACTER COUNT OF MSG 03947
5526 2096 06A0 2ECC BL SETUP 03948
5527 209A D0A0 D91A MOV3 GPIBADR,R2 GET GPIB SWITCH SETTINGS 03949
5528 209E 0982 SRL R2,8 MOVE TO LSB OF R2 03950
5529 20A0 0542 INV R2 03951
5530 20A2 0242 001F ANDI R2,$001F MASK OFF NON-ADDRESS SETTINGS 03952
5531 20A6 04C1 CLR R1 03953
5532 20A8 3C60 334C JIV C10,R1 03954
5533 20AC 0221 0030 AI R1,$0030 CONVERT TENS DIGIT TO ASCII 03955
5534 20B0 0222 0030 AI R2,$0030 CONVERT ONES DIGIT TO ASCII 03956
5535 20B4 0649 DECT SOFT STAY ON CURRENT LINE AND CHAR POSITION 03957
5536 20B6 C642 MOV R2,*SOFT PUSH ONES DIGIT 03958
5537 20B8 0649 DECT SOFT 03959
    
```

5538	20BA	C641		MOV	R1,*SOFT	PUSH TENS DIGIT	03960
5539	20BC	0649		DECT	SOFT		03961
5540	20BE	C660	333C	MOV	C2,*SOFT	PUSH CHARACTER COUNT	03962
5541	20C2	06A0	1362	BL	TEXTS		03963
5542	20C6	C020	0922	MOV	ONOFF,R0	SEE IF SPIB IS ONLINE	03964
5543	20CA	111A		JLT	OFFLIN		03965
5544	20CC	C020	0926	MOV	TOTLLO,R0		03966
5545	20D0	1104		JLT	DSPTKO		03967
5546	20D2	1586		JGT	DSPLNO		03968
5547	20D4	0201	2E90	LI	R1,ASCFL	TALK LISTEN OPTION	03969
5548	20D8	1002		JMP	*+6		03970
5549	20DA	0201	2E70	DSPTKO	LI R1,ASCFO	TALK ONLY OPTION	03971
5550	20DE	1002		JMP	*+6		03972
5551	20E0	0201	2E80	DSP_NO	LI R1,ASCLO	LISTEN ONLY OPTION	03973
5552	20E4	C0A0	3358	MOV	C16,R2	GET CHARACTER COUNT OF MSG	03974
5553	20E8	06A0	2ECC	BL	SETUP		03975
5554	20EC	C020	0924	MOV	TRMYPE,R0	IF TERMINATOR IS EDI ONLY ...	03976
5555	20F0	1300		JEQ	NXTLIN	... DONE WITH CURRENT LINE	03977
5556	20F2	0201	2EA8	LI	R1,ASCFL	... OTHERWISE ADD '/LF' TO END OF LINE	03978
5557	20F6	C0A0	333E	MOV	C3,R2	GET CHARACTER COUNT OF MSG	03979
5558	20FA	06A0	2ECC	BL	SETUP		03980
5559	20FE	1006		JMP	NXTLIN		03981
5560	2E00	0201	2EA0	OFFLIN	LI R1,ASCOFL	GET MESSAGE ADDRESS	03982
5561	2E04	C0A0	3348	MOV	C8,R2	GET CHARACTER COUNT OF MSG	03983
5562	2E08	06A0	2ECC	BL	SETUP		03984
5563			2E0C	NXTLIN	EQU *		03985
5564	2E0C	05C9		INCT	SOFT		03986
5565	2E0E	05D9		INCT	*SOFT	INSERT BLANK LINE	03987
5566	2E10	0649		DECT	SOFT		03988
5567	2E12	C660	333A	MOV	C1,*SOFT	RESET CHAR POSITION TO FRONT OF LINE	03989
5568	2E16	0201	2EAC	LI	R1,ASCON	GET MESSAGE ADDRESS	03990
5569	2E1A	C0A0	3340	MOV	C4,R2	GET CHARACTER COUNT OF MSG	03991
5570	2E1E	06A0	2ECC	BL	SETUP		03992
5571	2E22	05C9		INCT	SOFT		03993
5572	2E24	0599		INC	*SOFT	MOVE TO NEXT LINE	03994
5573	2E26	0649		DECT	SOFT		03995
5574	2E28	C660	333A	MOV	C1,*SOFT	RESET CHAR POINTER TO FRONT OF LINE	03996
5575	2E2C	0201	2EB0	LI	R1,ASCOFF	GET MESSAGE ADDRESS	03997
5576	2E30	C0A0	3340	MOV	C4,R2	GET CHARACTER COUNT OF MSG	03998
5577	2E34	06A0	2ECC	BL	SETUP		03999
5578				*			
5579				*	SET UP 3RD AND 4TH DISPLAY LINES		
5580				*			
5581	2E38	0204	32A2	LI	R4,RQSENBL		04000
5582	2E3C	0205	2EB4	LI	R5,ASCRQS		04001
5583	2E40	C0E0	091E	MOV	RSVENBL,R3		04002
5584			2E44	CJNSP	EQU *		04003
5585	2E44	05C9		INCT	SOFT		04004
5586	2E46	C660	3350	MOV	C12,*SOFT	PUSH LINE NUMBER	04005
5587	2E4A	20D4		CDC	*R4,R3	CHECK WHETHER FLAG IS ON OR OFF	04006
5588	2E4C	1301		JEQ	*+4		04007
5589	2E4E	0599		INC	*SOFT	IF FLAG OFF ENTER STATUS ON NEXT LINE	04008
5590	2E50	0649		DECT	SOFT		04009
5591	2E52	C0A0	3340	MOV	C4,R2	GET LENGTH OF MESSAGE	04010
5592	2E56	C045		MOV	R5,R1	GET MESSAGE ADDRESS	04011
5593	2E58	05C4		INCT	R4	GET NEXT FLAG	04012
5594	2E5A	A142		A	R2,R5	GET NEXT MESSAGE	04013
5595	2E5C	06A0	2ECC	BL	SETUP		04014
5596	2E60	0285	2ECC	CI	R5,SRQTEND	GOT ALL THE FLAGS?	04015

5597	2E64	11EF	JLT	CONDSP	...	ND, THEN KEEP GOING	04016
5598	2E66	0380	RTWP				04017
5599			*				
5600	2E68	41	ASCAJR	FCC	'ADDRESS:'		04018
	2E69	44					
	2E6A	44					
	2E6B	52					
	2E6C	45					
	2E6D	53					
	2E6E	53					
	2E6F	3A					
5601	2E70	20	ASCTO	FCC	' TALK ONLY EOI'		04019
	2E71	54					
	2E72	41					
	2E73	4C					
	2E74	4B					
	2E75	20					
	2E76	4F					
	2E77	4E					
	2E78	4C					
	2E79	59					
	2E7A	20					
	2E7B	20					
	2E7C	20					
	2E7D	45					
	2E7E	4F					
	2E7F	49					
5602	2E80	20	ASCLO	FCC	' LISTEN ONLY EOI'		04020
	2E81	4C					
	2E82	49					
	2E83	53					
	2E84	54					
	2E85	45					
	2E86	4E					
	2E87	20					
	2E88	4F					
	2E89	4E					
	2E8A	4C					
	2E8B	59					
	2E8C	20					
	2E8D	45					
	2E8E	4F					
	2E8F	49					
5603	2E90	20	ASCTL	FCC	' TALK LISTEN EOI'		04021
	2E91	54					
	2E92	41					
	2E93	4C					
	2E94	4B					
	2E95	20					
	2E96	4C					
	2E97	49					
	2E98	53					
	2E99	54					
	2E9A	45					
	2E9B	4E					
	2E9C	20					
	2E9D	45					
	2E9E	4F					
	2E9F	49					

5604	2EA0	20	ASCOFL	FCC	' OFFLINE'	04022
	2EA1	4F				
	2EA2	46				
	2EA3	46				
	2EA4	4C				
	2EA5	49				
	2EA6	4E				
	2EA7	45				
5605	2EA8	2F	ASCLF	FCC	' /LF '	04023
	2EA9	4C				
	2EAA	46				
	2EAB	20				
5606	2EAC	20	ASCON	FCC	' ONI'	04024
	2EAD	4F				
	2EAE	4E				
	2EAF	3A				
5607	2EB0	4F	ASCOFF	FCC	' OFFI'	04025
	2EB1	46				
	2EB2	46				
	2EB3	3A				
5608			*			
5609			*	THE FOLLOWING 6 FCC STATEMENTS MUST REMAIN		
5610			*	IN A CONTIGUOUS BLOCK		
5611			*			
5612	2EB4	20	ASCRQS	FCC	' RQS'	04026
	2EB5	52				
	2EB6	51				
	2EB7	53				
5613	2EB8	20		FCC	' IOC'	04027
	2EB9	49				
	2EBA	4F				
	2EBB	43				
5614	2EBC	20		FCC	' REM'	04028
	2EBD	52				
	2EBE	45				
	2EBF	4D				
5615	2EC0	20		FCC	' EXR'	04029
	2EC1	45				
	2EC2	58				
	2EC3	52				
5616	2EC4	20		FCC	' CER'	04030
	2EC5	43				
	2EC6	45				
	2EC7	52				
5617	2EC8	20		FCC	' OPC'	04031
	2EC9	4F				
	2ECA	50				
	2ECB	43				
5618	2ECC		SRQTEND	EQU	*	04032

SETJP

04033

5620				*****			
5621				**			**
5622				**	SETJP - PUSHES ADDRESS OF MESSAGE TO BE OUTPUT		**
5623				**	AND THE NUMBER OF CHARACTERS IT CONTAINS		**
5624				**	ALSO CALLS TEXT, WHICH PLACES THE		**
5625				**	MESSAGE IN THE DISPLAY BUFFER		**
5626				**			**
5627				**	INPUT - R1, CONTAINS THE ADDRESS OF THE MESSAGE		**
5628				**	R2, THE NUMBER OF CHARACTERS IN THE MESSAGE		**
5629				**			**
5630				**	DESTROYS - R6		**
5631				**			**
5632				*****			
5633		2ECC		SETJP	EQU *		04034
5634	2ECC	C18B		MOV	R11,R6	SAVE RETURN ADDRESS	04035
5635	2ECE	0649		DECF	SOFT	LINE # AND CHAR POINTER ALREADY ON STACK	04036
5636	2ED0	C641		MOV	R1,*SOFT	PUSH MESSAGE ADDRESS	04037
5637	2ED2	0649		DECF	SOFT		04038
5638	2ED4	C642		MOV	R2,*SOFT	PUSH CHARACTER COUNT	04039
5639	2ED6	06A0	136C	BL	TEXT		04040
5640	2EDA	C2C6		MOV	R6,R11	RESTORE RETURN ADDRESS	04041
5641	2EDC	045B		B	*R11		04042

QUERY 'ID?'

04043

```

5643 *****
5644 **
5645 ** 'ID?' QUERY HANDLER **
5646 ** ** **
5647 ** GPIB INTERRUPT ROUTINE **
5648 ** ** **
5649 ** NOTE --- **
5650 ** THIS ROUTINE RESPONDS TO THE GPIB QUERY 'ID?' BY **
5651 ** SENDING THE IDENTIFICATION MESSAGE: **
5652 ** ** **
5653 ** ID TEK/7854V79.1,XX.YY **
5654 ** ** **
5655 ** WHERE XX IS THE ROM VERSION # & **
5656 ** YY IS THE PATCH VERSION # **
5657 ** ** **
5658 *****
    
```

Address	OpCode	Op1	Op2	Op3	Op4	Description	Address
5659	2EDE	C20B				QJEID MOV R11,R10	04044
5660	2EE0	0200	2EFE			LI R0,IDMSG	04045
5661	2EE4	0202	0012			LI R2,10	04046
5662	2EE8	06A0	3252			BL OUTSTR	04047
5663	2EEC	0200	0A04			LI R0,VRSNUM	04048
5664	2EF0	0202	0008			LI R2,8	04049
5665	2EF4	06A0	3252			BL OUTSTR	04050
5666	2EF8	06A0	3196			BL TERMINATE	04051
5667	2EFC	045A				B *R10	04052
5668							

5669	2EFE		49			ICMSG FCG 'ID TEK/7854,V79.1,'	04053
	2EFF		44				
	2F00		20				
	2F01		54				
	2F02		45				
	2F03		48				
	2F04		2F				
	2F05		37				
	2F06		38				
	2F07		35				
	2F08		34				
	2F09		2C				
	2F0A		56				
	2F0B		37				
	2F0C		39				
	2F0D		2E				
	2F0E		31				
	2F0F		2C				

QUERY 'SRQ?'

04054

5671				*****			
5672				**			**
5673				**	'SRQ?'	QUERY HANDLER	**
5674				**			**
5675				**	GPIB	INTERRUPT ROUTINE	**
5676				**			**
5677				**	NOTE ---		**
5678				**	THIS ROUTINE	RESPONDS TO THE GPIB QUERY 'SRQ?'	**
5679				**	BY SENDING	LINE 16 OF THE DISPLAY READOUT TO THE	**
5680				**	GPIB.		**
5681				**			**
5682				*****			
5683	2F10	C28B		QJESRQ	MOV	R11,R10	SAVE RETURN ADDRESS 04055
5684	2F12	0200	2F2C		LI	R0,SRQMSG	ADDRESS OF START OF 'SRQ?' MESSAGE 04056
5685	2F16	0202	0004		LI	R2,4	BYTE COUNT OF STRING 04057
5686	2F1A	06A0	3262		BL	OUTSTR	OUTPUT START OF SRQ STRING 04058
5687	2F1E	C0A0	D9F2		MOV	LINE15,R2	GET ADDRESS OF LINE 16 OF DISPLAY READOUT 04059
5688	2F22	0206	2F2A		LI	R6,ENDSRQ	LOAD RETURN ADDRESS 04060
5689	2F26	0450	2AEE		B	NXTTEXT	BRANCH TO SHARED CODE 04061
5690	2F2A	045A		ENDSRQ	B	*R10	RETURN TO CALLER 04062
5691							
5692	2F2C		53	SRQMSG	FCC	'SRQ '	04063
	2F2D		52				
	2F2E		51				
	2F2F		20				

QUERY 'ERR?'

04064

```

5694 *****
5695 **
5696 ** 'ERR?' QUERY HANDLER **
5697 ** **
5698 ** GPIB INTERRUPT ROUTINE **
5699 ** **
5700 ** NOTE --- **
5701 ** THIS ROUTINE RESPONDS TO THE GPIB QUERY 'ERR?' BY **
5702 ** SENDING 'ERR 00' IF AN ERROR OR WARNING CONDITION **
5703 ** EXISTS AND SENDX 'ERR NONE' IF NO ERROR OR WARNING. **
5704 ** **
5705 *****
    
```

5706	2F30	C288		QJEERR	MOV	R11,R10	SAVE RETURN ADDRESS	04065
5707	2F32	0200	2F4E		LI	R0,ERR00	ADDRESS OF 'ERR 00' MESSAGE	04066
5708	2F36	C060	D94A		MOV	FATAL,R1	IS THERE AN ERROR?	04067
5709	2F3A	1102			JLT	*+6	NO, OUTPUT 'ERR 00'	04068
5710	2F3C	0200	2F54		LI	R0,ERR01	YES, OUTPUT 'ERR 01'	04069
5711	2F40	0202	0006		LI	R2,6	BYTE COUNT OF STRING	04070
5712	2F44	06A0	3262		BL	OUTSTR	OUTPUT ERROR MESSAGE	04071
5713	2F48	06A0	3196		BL	TERMINATE	ADD GPIB TERMINATOR	04072
5714	2F4C	045A			B	*R10		04073
5715								
5716	2F4E		45	ERR00	FCC	'ERR 00'		04074
	2F4F		52					
	2F50		52					
	2F51		20					
	2F52		30					
	2F53		30					
5717								
5718	2F54		45	ERR01	FCC	'ERR 01'		04075
	2F55		52					
	2F56		52					
	2F57		20					
	2F58		30					
	2F59		31					

KEYS 'SWL' & 'SWH'

04076

5720	2F5A	C320	33CE	KEYSWL	MOV	TTLOUT,R12	SET TTL CRU LINE LOW TO SET	04077
5721	2F5E	1000			SBZ	0	TTL OUTPUT LOW	04078
5722	2F60	0380			RTW ²			04079
5723								
5724	2F62	C320	33CE	KEYSWH	MOV	TTLOUT,R12	SET TTL CRU LINE HIGH TO SET	04080
5725	2F66	1E00			SBZ	0	TTL OUTPUT HIGH	04081
5726	2F68	0380			RTW ²			04082

5728	2F6A	C820	3348	DAE6	KEYRQS	MOV	C8,RQSNUM	ISSUE 'RQS' SERVICE REQUEST	04084
5729	2F70	06A0	32AE			BL	RQS		04085
5730	2F74	0380				RTWP			04086

I/O CONTROL TABLE FOR 'READX' & 'SENDX'

04087

5732			*****			
5733			**			**
5734			**	TABLE TO CONTROL WAVEFORM I/O VIA 'READX' & 'SENDX'		**
5735			**			**
5736			**	NOTE ---		**
5737			**	THE ORDER OF ENTRY OF THE CONTROL UNITS IS DEFINED BY		**
5738			**	THE ORDER THAT IS REQUIRED BY THE 'SENDX'		**
5739			**			**
5740			**	EACH CONTROL UNIT CONSISTS OF AN ASCII STRING PLUS		**
5741			**	ITS TERMINATOR AND THE ADDRESSES OF THE APPROPRIATE		**
5742			**	INPUT AND OUTPUT ROUTINES FOR THAT UNIT.		**
5743			**			**
5744			*****			
5745		2F76	WFNID	EQU	*	04088
5746	2F76	0007	WFMPRE	WORD	7	04089
5747	2F79	57	FCC	'WFMPRE'	WAVEFORM PREAMBLE HEADER	04090
	2F79	46				
	2F7A	40				
	2F7B	50				
	2F7C	52				
	2F7D	45				
	2F7E	20				
	2F7F	20				
5748	2F80	282A	WORD	WFMPRE		04091
5749	2F82	288C	WORD	OWFMPRE		04092
5750	2F84	0006	WORD	6		04093
5751	2F86	45	FCC	'ENCDS'	CURVE DATA ENCODING	04094
	2F87	4E				
	2F88	43				
	2F89	44				
	2F8A	47				
	2F8B	3A				
5752	2F8C	289C	WORD	IENCDS		04095
5753	2F8E	288E	WORD	OENCDS		04096
5754	2F90	0006	WORD	6		04097
5755	2F92	4E	FCC	'NR.PT'	NUMBER OF POINTS	04098
	2F93	52				
	2F94	2E				
	2F95	50				
	2F96	54				
	2F97	3A				
5756	2F98	28BE	WORD	INR.PT		04099
5757	2F9A	28A6	WORD	ONR.PT		04100
5758	2F9C	0007	WORD	7		04101
5759	2F9E	50	FCC	'PT.FMT'	POINT FORMAT	04102
	2F9F	54				
	2FA0	2E				
	2FA1	46				
	2FA2	40				
	2FA3	54				
	2FA4	3A				
	2FA5	20				
5760	2FA6	28D4	WORD	IPT.FMT		04103
5761	2FA8	2886	WORD	OPT.FMT		04104
5762	2FAA	0006	WORD	6		04105
5763	2FAC	58	FCC	'XZERO'	HORIZONTAL ZERO OFFSET	04106
	2FAD	5A				
	2FAE	45				
	2FAF	52				

	2FB0	4F				
	2FB1	3A				
5764	2FB2	28EA	WORD	IXZERO		04107
5765	2FB4	28C2	WORD	OXZERO		04108
5766	2FB6	0006	WORD	6		04109
5767	2FB8	58	FCC	'XINCR:'	HORIZONTAL INCREMENT BETWEEN POINTS	04110
	2FB9	49				
	2FBA	4E				
	2FBB	43				
	2FBC	52				
	2FBD	3A				
5768	2FBE	28FC	WORD	IXINCR		04111
5769	2FC0	28CE	WORD	OXINCR		04112
5770	2FC2	0006	WORD	6		04113
5771	2FC4	58	FCC	'XUNIT:'	HORIZONTAL SCALE FACTOR UNIT	04114
	2FC5	55				
	2FC6	4E				
	2FC7	49				
	2FC8	54				
	2FC9	3A				
5772	2FCA	2914	WORD	IXUNIT		04115
5773	2FCB	28DE	WORD	OXUNIT		04116
5774	2FCE	0006	WORD	6		04117
5775	2FD0	59	FCC	'YZERO:'	VERTICAL ZERO OFFSET	04118
	2FD1	5A				
	2FD2	45				
	2FD3	52				
	2FD4	4F				
	2FD5	3A				
5776	2FD6	2960	WORD	IYZERO		04119
5777	2FD8	28F0	WORD	OYZERO		04120
5778	2FDA	0006	WORD	6		04121
5779	2FDC	59	FCC	'YMULT:'	VERTICAL SCALE FACTOR	04122
	2FDD	40				
	2FDE	55				
	2FDF	4C				
	2FE0	54				
	2FE1	3A				
5780	2FE2	2940	WORD	IYMULT		04123
5781	2FE4	2C1A	WORD	OYMULT		04124
5782	2FE6	0006	WORD	6		04125
5783	2FE8	59	FCC	'YUNIT:'	VERTICAL SCALE FACTOR UNIT	04126
	2FE9	55				
	2FEA	4E				
	2FEB	49				
	2FEC	54				
	2FED	3A				
5784	2FEE	2978	WORD	IYUNIT		04127
5785	2FF0	2C22	WORD	OYUNIT		04128
5786	2FF2	0006	WORD	6		04129
5787	2FF4	43	FCC	'CURVE'	CURVE HEADER	04130
	2FF5	55				
	2FF6	52				
	2FF7	56				
	2FF8	45				
	2FF9	20				
5788	2FFA	2980	WORD	ICURVE		04131
5789	2FFC	2C44	WORD	OCURVE		04132
5790	2FFE	0000	WORD	0		04133

5792 *
 5793 * TABLE OF VALID 7854 QUERIES.
 5794 *
 5795 * TABLE CONSISTS OF AN ASCII STRING REPRESENTING THE QUERY
 5796 * WHICH IS TERMINATED BY 1 OR MORE BYTES OF \$00. THEN, ON AN
 5797 * EVEN WORD BOUNDARY, THE ADDRESS OF ITS CORRESPONDING ROUTINE
 5798 * IS GIVEN.
 5799 *

5800	3000	49	QUETABL FCC	'ID?'	IDENTIFY YOURSELF	04135
	3001	44				
	3002	3F				
5801	3003	00	BYTE	0		04136
5802	3004	2EDE	WORD	QUEID		04137
5803	3006	53	FCC	'SRQ?'	WHY DID YOU REQUEST SERVICE?	04138
	3007	52				
	3008	51				
	3009	3F				
5804	300A	0000	WORD	0		04139
5805	300C	2F10	WORD	QUESRQ		04140
5806	300E	45	FCC	'ERR?'	SPECIFY YOUR ERROR CONDITIONS	04141
	300F	52				
	3010	52				
	3011	3F				
5807	3012	0000	WORD	0		04142
5808	3014	2F30	WORD	QUEERR		04143
5809	3016		ENDQT3L EQU	*		04144

```

5811 *****
5812 **
5813 ** 'STRLSTN' - START LISTEN FUNCTION FOR GPIB INPUT KEYS **
5814 ** **
5815 ** LEVEL 5A ROUTINE **
5816 ** **
5817 ** INPUT: RQSNUM - SERVICE REQUEST NUMBER FOR LISTEN KEY **
5818 ** OUTPUT: GPIB INTERFACE SETUP TO LISTEN **
5819 ** DESTROYS: R7,R8,R10,R12 **
5820 ** **
5821 ** NOTE --- **
5822 ** THIS ROUTINE INITIALIZES ALL VARIABLES NEEDED BY GPIB **
5823 ** INPUT ROUTINES AND ISSUES THE APPROPRIATE SERVICE **
5824 ** REQUEST TO THE GPIB CONTROLLER TO SETUP THE 7854 AS **
5825 ** A GPIB LISTENER. IF AN ERROR CONDITION IS DETECTED **
5826 ** THE ERROR FLAG IS SET AND THE KEY IS ABORTED. **
5827 ** **
5828 *****
5829 3016 C28B STRLSTN MOV R11,R10 SAVE RETURN ADDRESS 04146
5830 3018 C1E0 D966 MOV KEY,R7 CURRENTLY EXECUTING KEYCODE WITH FLAGS 04147
5831 301C 21E0 3382 COC STOPKEY,R7 SHOULD THIS KEY BE STOPPED? 04148
5832 3020 130D JEQ ERRLSTN YES, ABORT KEY 04149
5833 3022 E020 3382 D966 SOC STOPKEY,KEY ALLOW KEY TO BE STOPPED 04150
5834
5835 3028 C1E0 D942 MOV GPIBOPT,R7 IS THE GPIB OPTION INSTALLED? 04151
5836 302C 1148 JLT ERRTALK NO, ABORT KEY 04152
5837 302E C1E0 D922 MOV ONOFF,R7 IS 7854 ONLINE TO GPIB BUS? 04153
5838 3032 1145 JLT ERRTALK NO, ABORT KEY 04154
5839
5840 3034 C1E0 D926 MOV TOTLLO,R7 HOW IS 7854'S GPIB MODE SELECTED? 04155
5841 3038 1522 JGT LOLSTN LISTEN-ONLY, WITHOUT SRQ 04156
5842 303A 131F JEQ TLLSTN TALK-LISTEN, WITH SRQ IF ENABLED 04157
5843 303C 8820 D944 3336 ERRLSTN C GPIBKEY,CN1 WHICH INPUT KEY IS EXECUTING? 04158
5844 3042 133D JEQ ERRTALK '>TEXT' 04159
5845 3044 C1E0 D91C MOV CNSWFM,R7 'READX', CHECK WHETHER CONSTANT OR WAVEFORM 04160
5846 3048 1102 JLT ZEROCNS CONSTANT 04161
5847 304A 1339 JEQ ERRTALK NEITHER 04162
5848 304C 1502 JGT ZEROWFM WAVEFORM 04163
5849 304E 04C0 ZEROCNS CLR R0 FLAG INDICATING FLOATING POINT NUMBER 04164
5850 3050 100D JMP ZCNSWFM 04165
5851 3052 0649 ZEROWFM DECT SOFT 04166
5852 3054 0439 CLR *SOFT 04167
5853 3056 06A0 6F25 BL NULLWFM ZERO WAVEFORM #0 04168
5854 305A 04C1 CLR R1 04169
5855 305C 06A0 6984 BL ADWFM GET ADDRESS OF W0'S HEADER 04170
5856 3060 C079 MOV *SOFT+,R1 04171
5857 3062 A060 3354 A DISPLA,R1 ADDRESS OF W0'S DISPLAY FLAG 04172
5858 3066 E460 333C SOC C2,*R1 ENSURE W0 IS DISPLAYED AS THE OPW 04173
5859 306A 0700 SETO R0 FLAG INDICATING WAVEFORM NUMBER 04174
5860 306C 04C1 ZCNSWFM CLR R1 04175
5861 306E 04C2 CLR R2 PUSH RESULT ONTO USERSTACK 04176
5862 3070 06A0 6962 BL PSHREG 04177
5863 3074 04E0 D91C CLR CNSWFM FLAG INDICATING 'READX' IDLE 04178
5864 3078 1022 JMP ERRTALK 04179
5865
5866 307A 06A0 32AE TLLSTN BL RQS ISSUE LOCAL RSV (SRQ IF ENABLED) 04180
5867 307E C1E0 D940 LOLSTN MOV INOUT,R7 CHECK GPIB I/O STATUS 04181
5868 3082 13FD JEQ LOLSTN IF IDLE, WAIT 04182
5869 3084 15DB JGT ERRLSTN IF OUTPUT, ERROR 04183
  
```


STRLSTN - START LISTEN FUNCTION FOR GPIB INPUT KEYS

04145

5870	3086	C1E0	D926	MOV	TOTLLO,R7	CHECK IF 7054 IS IN TALK-LISTEN MODE	04184
5871	308A	1603		JNE	*+8	IF NOT, ENABLE STOP KEY DURING TRANSMISSION	04185
5872	308C	4820	3382	SZC	STOPKEY,KEY	IF INPUT, KEY CAN CONTINUE	04186
5873	3092	06A0	3304	BL	RQSOFF	TURN SERVICE REQUEST OFF	04187
5874	3096	045A		B	*R10		04188

5876					*****		
5877					**		**
5878					**	'STRTALK' - START TALK FUNCTION FOR GPIB OUTPUT KEYS	**
5879					**		**
5880					**	LEVEL 5A ROUTINE	**
5881					**		**
5882					**	INPUT: RQSNUM - SERVICE REQUEST NUMBER FOR TALK KEY	**
5883					**	OUTPJT: GPIB INTERFACE SETUP TO TALK	**
5884					**	DESTROYS: R7,R8,R10,R12	**
5885					**		**
5886					**	NOTE ---	**
5887					**	THIS ROUTINE INITIALIZES ALL VARIABLES NEEDED BY GPIB	**
5888					**	OUTPUT ROUTINES AND ISSUES THE APPROPRIATE SERVICE	**
5889					**	REQUEST TO THE GPIB CONTROLLER TO SETUP THE 7854 AS	**
5890					**	A GPIB TALKER. IF AN ERROR CONDITION IS DETECTED	**
5891					**	THE ERROR FLAG IS SET AND THE KEY IS ABORTED.	**
5892					**		**
5893					*****		
5894	3098	C288			STRTALK	MOV R11,R10	SAVE RETURN ADDRESS. 04190
5895	309A	C1E0	D966			MOV KEY,R7	CURRENTLY EXECUTING KEYCODE WITH FLAGS 04191
5896	309E	21E0	3382			CDC STOPKEY,R7	SHOULD THIS KEY BE STOPPED? 04192
5897	30A2	130D				JEQ ERRTALK	YES, ABORT KEY 04193
5898	30A4	E820	3382	D966		SOC STOPKEY,KEY	ALLOW KEY TO BE STOPPED 04194
5899							
5900	30AA	C1E0	D942			MOV GPIBOPT,R7	IS THE GPIB OPTION INSTALLED? 04195
5901	30AE	1107				JLT ERRTALK	NO, ABORT KEY 04196
5902	30B0	C1E0	D922			MOV ONOFF,R7	IS 7854 ONLINE TO GPIB BUS? 04197
5903	30B4	1104				JLT ERRTALK	NO, ABORT KEY 04198
5904							
5905	30B6	C1E0	D925			MOV TOTLLO,R7	HOW IS 7854'S GPIB MODE SELECTED? 04199
5906	30BA	1113				JLT TOTALK	TALK-ONLY, WITHOUT SRQ 04200
5907	30BC	1309				JEQ TLTALK	TALK-LISTEN, WITH SRQ IF ENABLED 04201
5908	30BE	04E0	D94A		ERRTALK	CLR FATAL	LISTEN-ONLY, IS AN ERROR 04202
5909	30C2	04E0	D944			CLR GPIBKEY	FLAG INDICATING GPIB KEY IS FINISHED 04203
5910	30C6	0720	D972			SETO RDTFLAG	UPDATE ENTIRE CALCULATOR READOUT 04204
5911	30CA	06A0	3304			BL RQSOFF	TURN SERVICE REQUEST OFF 04205
5912	30CE	0380				RTWP	04206
5913							
5914	30D0	06A0	32AE		TLTALK	BL RQS	ISSUE LOCAL RSV (SRQ IF ENABLED) 04207
5915	30D4	C1E0	D940			MOV INOUT,R7	CHECK GPIB I/O STATUS 04208
5916	30D8	13FD				JEQ *-4	IF IDLE, WAIT 04209
5917	30DA	1503				JGT TOTALK	IF OUTPUT, EVERYTHING IS ALRIGHT 04210
5918	30DC	04E0	D940			CLR INOUT	FLAG INDICATING GPIB I/O IDLE 04211
5919	30E0	10EE				JMP ERRTALK	04212
5920	30E2	D320	E048		TOTALK	MOV R2R,R12	READ ADDRESS STATUS REGISTER 04213
5921	30E6	2320	337C			CDC LACS,R12	IS 7854 LISTEN ADDRESSED? 04214
5922	30EA	13E9				JEQ ERRTALK	YES, STOP WAITING 04215
5923	30EC	D320	E04C			MOV R3R,R12	HANDSHAKE DATA IN HIGH BYTE OF R1 04216
5924	30F0	D24C	7800			ANDI R12,\$7000	KEEP HANDSHAKE LINES ONLY 04217
5925	30F4	D28C	3000			CI R12,\$3000	IS THERE A VALID GPIB LISTENER? 04218
5926	30F8	16F4				JNE TOTALK	NO, WAIT FOR ONE 04219
5927	30FA	C820	333A	D940		MOV C1,INOUT	YES, SET TALK FLAG 04220
5928	3100	C1E0	D925			MOV TOTLLO,R7	CHECK IF 7854 IS IN TALK-LISTEN MODE 04221
5929	3104	1603				JNE *-8	IF NOT, ENABLE STOP KEY DURING TRANSMISSION 04222
5930	3106	4820	3382	D966		SZC STOPKEY,KEY	IF OUTPUT, KEY CAN CONTINUE 04223
5931	310C	06A0	3304			BL RQSOFF	TURN SERVICE REQUEST OFF 04224
5932	3110	045A				B *-R10	04225

```

5934 *****
5935 **
5936 ** CHECK IF DATA BYTE IS GPIB DELIMITER **
5937 ** **
5938 ** LEVEL 5 ROUTINE **
5939 ** **
5940 *****
5941 *CHKOLM LI R7,DLMTBL COMMAND DELIMITERS' TABLE 04227 DEL
5942 * 00023PATCH
5943 * PROBLEM #18 - PATCH #23 (2 OF 2) 00023PATCH
5944 * 00023PATCH
5945 * CORRECT INCOMPATIBILITIES WITH TEK 4924 GPIB TAPE DRIVE 00023PATCH
5946 * 00023PATCH
5947 3112 0207 97E4 CHKOLM LI R7,DLMTBL1 LOAD ALTERNATE DELIMITER TABLE ADDRESS 00023PATCH
5948 * 00023PATCH
5949 * END OF PATCH #23 OF PROBLEM #18 00023PATCH
5950 * 00023PATCH
5951 3116 0237 MOV *R7+,R8 COUNT OF DELIMITERS IN TABLE 04228
5952 3118 90C1 CB R1,*R7+ IS THIS CHARACTER IS A DELIMITER? 04229
5953 311A 1303 JEQ ISOLM YES, RETURN WITH = STATUS 04230
5954 311C 0608 DEC R8 04231
5955 311E 15FC JGT *-6 04232
5956 3120 0608 DEC R8 NO, RETURN WITH <> STATUS 04233
5957 3122 045B ISOLM B *R11 04234
    
```

GPIBIN - GPIB INPUT

04235

```

5959 *****
5960 **
5961 ** INPUT DATA BYTE FROM GPIB (KEY ROUTINES ONLY) **
5962 **
5963 ** LEVEL 5C ROUTINE (SPECIAL - SEE NOTE) **
5964 **
5965 ** INPUT: DATA BYTE FROM GPIB **
5966 ** OUTPUT: R1 GETS DATA BYTE IN HIGH BYTE (LOW BYTE $00) **
5967 ** INTGPIB1 GETS GPIB INTERRUPT STATUS **
5968 ** DESTROYS: R1 **
5969 **
5970 ** STACK OPERATIONS: **
5971 ** NONE **
5972 **
5973 ** NOTE --- **
5974 ** THIS ROUTINE CAN ONLY USE REGISTER R1. **
5975 **
5976 *****
5977 *GPIBIN MOV INOUT,R1 CHECK GPIB I/O STATUS 04236 DEL
5978 * 00021PATCH
5979 * PROBLEM #17 - PATCH #21 (1 OF 1) 00021PATCH
5980 * 00021PATCH
5981 * SWITCH FROM DATA INPUT TO COMMAND INPUT DOESN'T OCCUR CORRECTLY 00021PATCH
5982 * 00021PATCH
5983 3124 0460 978A GPIBIN B PATCH21 BRANCH TO PATCH #21 (REPLACE ENTIRE ROUTINE) 00021PATCH
5984 * 00021PATCH
5985 * END OF PROBLEM #17 00021PATCH
5986 * 00021PATCH
5987 3128 13FD JEQ GPIBIN IF IOLE, WAIT 04237
5988 312A 1588 JGT ERRLSTN IF OUTPUT, ERROR 04238
5989 312C C820 D930 D932 MOV INTGPIB,INTGPIB1 SAVE THIS DATA BYTES INTERRUPT STATUS 04239
5990 3132 04E0 D940 CLR INOUT FLAG INDICATING GPIB I/O IOLE 04240
5991 3136 D060 E05C MOVB R7R,R1 READ DATA BYTE 04241
5992 313A 0241 7F00 ANDI R1,$7F00 KEEP ONLY GOOD INFO 04242
5993 313E C820 D924 D924 MOV TRMTPY,TRMTPY CHECK TERMINATOR TYPE 04243
5994 3144 1306 JEQ EOIONLY IF <EOI> ONLY THEN DONE 04244
5995 3146 9801 331F CB R1,LFB CHECK FOR <EOI> OR <LF> TERMINATOR 04245
5996 314A 1603 JNE EOIONLY 04246
5997 314C E820 337A D932 SOC END,INTGPIB1 IF <LF> SET <EOI> FLAG 04247
5998 3152 045B EOIONLY B *R11 04248
    
```

6000					*****			
6001					**			**
6002					**	OUTPUT DATA BYTE TO GPIB (KEY ROUTINES & QUERIES)		**
6003					**			**
6004					**	LEVEL 50 ROUTINE (SPECIAL - SEE NOTE)		**
6005					**			**
6006					**	INPUT: DATA BYTE IN HIGH BYTE OF R1		**
6007					**	OUTPUT: DATA BYTE TO GPIB		**
6008					**	DESTROYS: R1		**
6009					**			**
6010					**	STACK OPERATIONS:		**
6011					**	NONE		**
6012					**			**
6013					**	NOTE ---		**
6014					**	THIS ROUTINE CAN USE ONLY REGISTER R1.		**
6015					**			**
6016					*****			
6017	3154	C820	D940	D940	GPIBOUT	MOV INOUT, INOUT	CHECK GPIB I/O STATUS	04250
6018	315A	13FC				JEQ GPIBOUT	IF IDLE, WAIT	04251
6019	315C	1113				JLT ERRORT	IF INPUT, ERROR	04252
6020	315E	04E0	D940			CLR INOUT	SET IDLE STATUS	04253
6021	3162	06C1				SWP3 R1	PUT DATA BYTE IN LOW BYTE	04254
6022	3164	0860	E040		NOLSTNR	MOVB R2R, R1	READ ADDRESS STATUS REGISTER	04255
6023	3168	2060	337C			COC LACS, R1	IS 7854 LISTEN ADDRESSED?	04256
6024	316C	130B				JEQ ERRORT	YES, STOP WAITING	04257
6025	316E	0850	E04C			MOV3 R3R, R1	HANDSHAKE DATA IN HIGH BYTE OF R1	04258
6026	3172	0241	70FF			ANDI R1, 870FF	KEEP HANDSHAKE LINES ONLY	04259
6027	3176	9801	3384			CB R1, CH3000	IS THERE A VALID GPIB LISTENER?	04260
6028	317A	16F4				JNE NOLSTNR	NO, WAIT FOR ONE	04261
6029	317C	06C1				SWP3 R1	PUT DATA BYTE BACK INTO HIGH BYTE	04262
6030	317E	0801	E07C			MOV3 R1, R7H	SEND DATA BYTE TO GPIB	04263
6031	3182	045B			NOTGPIB	B *R11		04264
6032	3184	04E0	D940		ERRORT	CLR INOUT	FLAG INDICATING GPIB I/O IDLE	04265
6033	3188	C060	D944			MOV GPIBKEY, R1	WHO IS CALLER?	04266
6034	318C	13FA				JEQ NOTGPIB	IF NOT GPIB ROUTINE, JUST RETURN	04267
6035	318E	0281	0004			CI R1, 4	IS THIS A QUERY OUTPUT?	04268
6036	3192	1695				JNE ERRORTALK	NO, ABORT KEY	04269
6037	3194	045A				B *R10	YES, ABORT QUERY	04270

TERMINATE - GPIB OUTPUT TERMINATION

04271

```

6039 *****
6040 **
6041 **      OUTPUT TERMINATOR TO GPIB (KEY ROUTINES & QUERIES)      **
6042 **
6043 **      LEVEL 5C ROUTINE (SPECIAL - SEE NOTE)                      **
6044 **
6045 **      INPUT:      NONE
6046 **      OUTPUT:     TERMINATOR TO GPIB
6047 **      DESTROYS:   R1
6048 **
6049 **      STACK OPERATIONS:
6050 **      NONE
6051 **
6052 **      NOTE ---
6053 **      THIS ROUTINE CAN USE ONLY REGISTER R1.
6054 **
6055 *****
6056 3196 C820 D940 D940 TERMINATE MOV INOUT,INOUT CHECK GPIB I/O STATUS      04272
6057 319C 13FC JEQ TERMINATE IF IDLE, WAIT 04273
6058 319E 11F2 JLT ERRORT IF INPUT, ERROR 04274
6059 31A0 D060 E048 NOTRMLS MOV3 R2R,R1 READ ADDRESS STATUS REGISTER 04275
6060 31A4 2060 337C CDC LACS,R1 IS 7854 LISTEN ADDRESSED? 04276
6061 31A8 13ED JEQ ERRORT YES, STOP WAITING 04277
6062 31AA D060 E04C MOV3 R3R,R1 HANDSHAKE DATA IN HIGH BYTE OF R1 04278
6063 31AE 0241 7000 ANDI R1,$7000 KEEP HANDSHAKE LINES ONLY 04279
6064 31B2 0281 3000 CI R1,$3000 IS THERE A VALID GPIB LISTENER? 04280
6065 31B6 16F4 JNE NOTRMLS NO, WAIT FOR ONE 04281
6066 31B8 D060 331E MOV3 CRB,R1 FIRST OUTPUT <CR> 04282
6067 31BC C820 D924 D924 MOV TRMTYPE,TRMTYPE CHECK TERMINATOR TYPE 04283
6068 31C2 1317 JEQ EOIOJT IF <EOI> ONLY SEND <CR> WITH <EOI> 04284
6069 31C4 04E0 D940 CLR INOUT FLAG INDICATING GPIB I/O IDLE 04285
6070 31C8 D801 E07C MOV3 R1,R7W OUTPUT <CR> 04286
6071 31CC C820 D940 D940 MOV INOUT,INOUT CHECK GPIB I/O STATUS 04287
6072 31D2 13FC JEQ *-6 IF IDLE, WAIT 04288
6073 31D4 11D7 JLT ERRORT IF INPUT, ERROR 04289
6074 31D6 D060 E048 NOTRML MOV3 R2R,R1 READ ADDRESS STATUS REGISTER 04290
6075 31DA 2060 337C CDC LACS,R1 IS 7854 LISTEN ADDRESSED? 04291
6076 31DE 13D2 JEQ ERRORT YES, STOP WAITING 04292
6077 31E0 D060 E04C MOV3 R3R,R1 HANDSHAKE DATA IN HIGH BYTE OF R1 04293
6078 31E4 0241 7000 ANDI R1,$7000 KEEP HANDSHAKE LINES ONLY 04294
6079 31E8 0281 3000 CI R1,$3000 IS THERE A VALID GPIB LISTENER? 04295
6080 31EC 16F4 JNE NOTRML NO, WAIT FOR ONE 04296
6081 31EE D060 331F MOV3 LFB,R1 OUTPUT <LF> WITH <EOI> 04297
6082 31F2 0720 D92A EOIOJT SET3 SEOI FLAG INDICATING LAST BYTE OUTPUT 04298
6083 31F6 04E0 D940 CLR INOUT FLAG INDICATING GPIB I/O IDLE 04299
6084 31FA C320 33BA MOV GPIBEOI,R12 CRU ADDRESS FOR <EOI> LINE 04300
6085 31FE 1000 SBO 0 TURN <EOI> LINE ON 04301
6086 3200 D801 E07C MOV3 R1,R7W OUTPUT TERMINATOR DATA BYTE 04302
6087 3204 045B B *R11 04303

```

CHKEDI - CHECK IF <EOI> CAME WITH LAST DATA BYTE

04304

```

6089 *****
6090 **
6091 ** CHECK IF <EOI> CAME WITH LAST DATA BYTE **
6092 ** **
6093 ** LEVEL 5 ROUTINE **
6094 ** **
6095 *****
6096 3206 C1E0 D932 CHKEDI MOV INTGPIB1,R7 INTERRUPT STATUS OF LAST BYTE INPUT 04305
6097 320A 21E0 337A CDC END,R7 DID <EOI> COME WITH LAST BYTE? 04306
6098 320E 045B B *R11 = STATJS BIT HAS THE ANSWER (= IS YES) 04307

```

INSTR - INPUT STRING FROM GPIB

04308

6100					*****	
6101					**	**
6102				**	INPUT STRING FROM GPIB	**
6103				**		**
6104				**	LEVEL 5B ROUTINE (SPECIAL - SEE GPIB NOTES ABOVE)	**
6105				**		**
6106					*****	
6107	3210	C30B		INSTR	MOV R11,R12	SAVE RETURN ADDRESS 04309
6108	3212	C020	33E0		MOV GPIBBUF,R0	ADDRESS FOR GPIB INPUT 04310
6109	3216	06A0	3124	NXTSTR	BL GPIBIN	GET NEXT GPIB DATA BYTE 04311
6110	321A	9801	3320		CB R1,SPACEB	DON'T USE A SPACE AS A DELIMITER 04312
6111	321E	1303			JEQ SPCSTR	04313
6112	3220	06A0	3112		BL CHKDLH	CHECK FOR DELIMITER 04314
6113	3224	1304			JEQ STRIN	04315
6114	3226	0C01		SPCSTR	MOV3 R1,*R0+	MOVE THIS DIGIT TO BUFFER 04316
6115	3228	06A0	3206		BL CHKEOI	DID <EOI> COME WITH THIS DATA BYTE? 04317
6116	322C	16F4			JNE NXTSTR	IF NOT, GET NEXT DATA BYTE 04318
6117	322E	0420	3321	STRIN	MOV3 CONHAB,*R0	DELIMITER FOR STRING 04319
6118	3232	045C			B *R12	04320

INVAL - INPJT VALUE FROM GPIB

04321

```

6120 *****
6121 **
6122 ** INPUT VALUE FROM GPIB **
6123 ** **
6124 ** LEVEL 5B ROUTINE (SPECIAL - SEE GPIB NOTES ABOVE) **
6125 ** **
6126 *****
6127 3234 C30B INVAL MOV R11,R12 SAVE RETURN ADDRESS 04322
6128 3236 C020 33E0 MOV GPIBBUF,R0 ADDRESS FOR GPIB INPUT 04323
6129 323A 06A0 3124 NXTVAL BL GPIBIN GET NEXT GPIB DATA BYTE 04324
6130 323E 06A0 3112 BL CHKOLM CHECK FOR DELIMITER 04325
6131 * JEQ VALIN 04326 DEL
6132 * MOVB R1,*R0+ MOVE THIS DIGIT TO BUFFER 04327 DEL
6133 * DLMVAL BL CHKEOI DID <EOI> COME WITH THIS DATA BYTE? 04328 DEL
6134 * JNE NXTVAL IF NOT, GET NEXT DATA BYTE 04329 DEL
6135 * VALIN C GPIBBJF,R0 IF DELIMITER ONLY GET MORE 04330 DEL
6136 * JEQ DLMVAL 04331 DEL
6137 * 00020PATCH
6138 * PROBLEM #15 - PATCH #20 (1 OF 1) 00020PATCH
6139 * 00020PATCH
6140 * BUFFER OVERFLOW IS NOT CHECKED FOR IN VALUE INPUT 00020PATCH
6141 * 00020PATCH
6142 3242 0460 975E B PATCH20 BRANCH TO PATCH #20 00020PATCH
6143 3246 BACK2J EQU * DEFINE REENTRY POINT 00020PATCH
6144 * 00020PATCH
6145 * END OF PROBLEM #15 00020PATCH
6146 * 00020PATCH
6147 3246 1005 JMP *+012 00020PATCH
6148 3248 1004 JMP *+010 00020PATCH
6149 324A 1003 JMP *+008 00020PATCH
6150 324C 1002 JMP *+006 00020PATCH
6151 324E 1001 JMP *+004 00020PATCH
6152 3250 1000 JMP *+002 00020PATCH
6153 3252 0420 3321 MOV3 COMMB,*R0 DELIMITER FOR 'SCANIN' 04332
6154 3256 C020 33E0 MOV GPIBBUF,R0 ADDRESS OF INPUT ASCII VALUE 04333
6155 325A 0420 6618 BLWP SCANIN CONVERT VALUE TO FLOATING POINT 04334
6156 325E C041 MOV R1,R1 SET STATUS BITS =,> 04335
6157 3260 045C B *R12 04336

```

OUTSTR - OUTPUT STRING TO GPIB

04337

```

6159 *****
6160 **
6161 **      OUTPUT AN ASCII DATA STRING TO GPIB      **
6162 **
6163 **      LEVEL 5B ROUTINE (SPECIAL - SEE NOTE)      **
6164 **
6165 **      INPUT:   R0 - ADDRESS OF ASCII STRING      **
6166 **              R2 - BYTE COUNT OF STRING          **
6167 **      OUTPUT:  ASCII STRING TO GPIB              **
6168 **      DESTROYS: R0,R1,R2,R12                     **
6169 **
6170 **      STACK OPERATIONS:                          **
6171 **              NONE                                **
6172 **
6173 **      NOTE ---                                    **
6174 **      THIS ROUTINE CANNOT USE R10 AS THIS IS USED BY OTHER **
6175 **      LEVEL 5 ROUTINES WHICH CALL THIS ROUTINE.  **
6176 **

```

6177 *****

6178	3262	C30B	OUTSTR	MOV	R11,R12	SAVE RETURN ADDRESS	04338
6179	3264	D070		MOVB	*R0+,R1	GET NEXT DATA BYTE TO OUTPUT	04339
6180	3266	06A0	3154	BL	GPIBOUT	OUTPUT DATA BYTE TO GPIB	04340
6181	326A	0602		DEC	R2	DECREMENT BYTE COUNT	04341
6182	326C	15FB		JGT	OUTSTR+2		04342
6183	326E	045C		B	*R12		04343

OUTVAL - OUTPUT VALUE TO GPIB

04344

6185				*****			
6186				**			**
6187				**	OUTPUT FLOATING POINT VALUE IN ASCII TO GPIB		**
6188				**			**
6189				**	LEVEL 5B ROUTINE (SPECIAL - SEE NOTE)		**
6190				**			**
6191				**	INPUT: R1,R2 - FLOATING POINT NUMBER TO OUTPUT		**
6192				**	GFORMAT - ASCII CONVERSION FORMAT		**
6193				**	OUTPJT: ASCII VALUE TO GPIB		**
6194				**	DESTROYS: R1,R2,R12		**
6195				**			**
6196				**	STACK OPERATIONS:		**
6197				**	SOFTSTACK: USED BUT NO EFFECT TO CALLER		**
6198				**			**
6199				**	NOTE ---		**
6200				**	THIS ROUTINE CANNOT USE R10 AS IT IS USED BY OTHER		**
6201				**	LEVEL 5 ROUTINE WHICH CALL THIS ROUTINE.		**
6202				**			**
6203				**	THIS ROUTINE CAN ONLY BE CALLED FROM ROUTINES USING		**
6204				**	THE LEVEL 1 WORKSPACE. THIS REQUIREMENT IS NEEDED		**
6205				**	BECAUSE THE LEVEL 2 ROUTINE 'FP2ASC' IS USED TO		**
6206				**	CONVERT THE FLOATING POINT NUMBER TO ASCII.		**
6207				**			**
6208				*****			*****
6209	3270	C30B		OJTVAL	MOV R11,R12	SAVE RETURN ADDRESS	04345
6210	3272	0649			DECT SOFT		04346
6211	3274	C642			MOV R2,*SOFT	PUSH FLOATING POINT NUMBER ONTO SOFTSTACK	04347
6212	3276	0649			DECT SOFT		04348
6213	3278	C641			MOV R1,*SOFT		04349
6214	327A	0649			DECT SOFT		04350
6215	327C	C660	0920		MOV GFORMAT,*SOFT	PUSH CONVERSION FORMAT ONTO SOFTSTACK	04351
6216	3280	0420	75E0		BLWP FP2ASC	CONVERT NUMBER TO ASCII	04352
6217	3284	C0B9			MOV *SOFT+,R2	POP BYTE COUNT OF ASCII NUMBER	04353
6218	3286	C079		OJTFP	MOV *SOFT+,R1	GET NEXT DATA BYTE	04354
6219	3288	0A81			SLA R1,8	PUT DATA BYTE IN HIGH BYTE	04355
6220	328A	06A0	3154		BL GPIBOUT	OUTPUT DATA BYTE TO GPIB	04356
6221	328E	0602			DEC R2	DECREMENT BYTE COUNT	04357
6222	3290	15FA			JST OUTFP		04358
6223	3292	045C			B *R12		04359

```

6225 *****
6226 **
6227 ** GPIB SERVICE REQUEST GENERATOR **
6228 ** **
6229 ** LEVEL 5 ROUTINE **
6230 ** **
6231 ** INPUT: RQSNUM (REQUEST SERVICE NUMBER) **
6232 ** OUTPUT: GPIB SRQ (IF ENABLED) **
6233 ** DESTROYS: R7,R8,R12 **
6234 ** **
6235 ** NOTE --- **
6236 ** GPIB SERVICE REQUESTS ARE GENERATED WHENEVER THE GPIB **
6237 ** INTERFACE IS IN A REMOTE CONDITION (REMS & RWLS) AND **
6238 ** THE 7854 NEEDS ASSISTANCE TO CONTINUE OPERATION. **
6239 ** **
6240 ** MOST SERVICE REQUESTS CAN BE DISABLED BY THE INTERFACE. **
6241 ** THE TABLE CONTAINING THE APPROPRIATE STATUS BYTE FOR **
6242 ** EACH SERVICE REQUEST NUMBER LISTS WHICH RSV COMMAND **
6243 ** CONTROLS THAT SERVICE REQUEST. A 16 BIT WORD IN RAM **
6244 ** IS USED TO STORED THE CURRENT STATE (ON, OFF) FOR EACH **
6245 ** SERVICE REQUEST NUMBER. EACH NUMBER USES THE SAME BIT **
6246 ** NUMBER IN THE WORD FOR ITS CONTROL (0=OFF, 1=ON). **
6247 ** BIT 15 IS USED FOR OVERALL CONTROL VIA THE 'RQSON' & **
6248 ** 'RQSOFF' COMMANDS. **
6249 ** **
6250 ** THE TWO POWER ON SERVICE REQUESTS CANNOT BE TURNED OFF. **
6251 ** THIS IS ACCOMPLISHED BY SETTING THE RSV BIT IN THEIR **
6252 ** STATUS BYTES IN THE TABLE. **
6253 ** **
6254 *****
  
```

				NO.	7854 CONDITION	RSV CMD	
6257							
6258	3294	00	RSVT3L BYTE \$00	0	- NO RSV PENDING		04361
6259	3295	02	BYTE \$02	1	- COMMAND GROUP COMPLETED	OPC	04362
6260	3296	02	BYTE \$02	2	- END OF PROGRAM	OPC	04363
6261	3297	21	BYTE \$21	3	- EXTERNAL COMMAND ERROR	CER	04364
6262	3298	00	BYTE \$00				04365
6263	3299	22	BYTE \$22	5	- EXECUTION ERROR	EXR	04366
6264	329A	41	BYTE \$41	6	- POWER ON		04367
6265	329B	63	BYTE \$63	7	- POWER ON CHECK FAILURE		04368
6266	329C	10	BYTE \$10	8	- RQS COMMAND	REM	04369
6267	329D	90	BYTE \$90	9	- SAVE COMMAND	IOC	04370
6268	329E	92	BYTE \$92	10	- SENDX COMMAND	IOC	04371
6269	329F	93	BYTE \$93	11	- READX COMMAND	IOC	04372
6270	32A0	94	BYTE \$94	12	- TEXT COMMAND	IOC	04373
6271	32A1	95	BYTE \$95	13	- >TEXT COMMAND	IOC	04374
6272							
6273	32A2	0000	RQSEN3L WORD \$0000		ENABLE FLAG FOR ALL SERVICE REQUESTS		04375
6274	32A4	3E00	IOCMENBL WORD \$3E00		" " " IOC " "		04376
6275	32A6	0100	REMEN3L WORD \$0100		" " " REM " "		04377
6276	32A8	0020	EXRENBL WORD \$0020		" " " EXR " "		04378
6277	32AA	0018	CEREN3L WORD \$0018		" " " CER " "		04379
6278	32AC	0006	OPCMENBL WORD \$0006		" " " OPC " "		04380
6279							
6280	32AE	C320 D942	RQS MOV GPIBOPT,R12		IS GPIB OPTION INSTALLED IN 7854?		04381
6281	32B2	112E	JLT NORQS		NO, THEN DON'T ISSUE A SERVICE REQUEST		04382
6282	32B4	C320 D922	MOV ONOFF,R12		IS 7854 ONLINE TO THE GPIB BUS?		04383
6283	32B8	112B	JLT NORQS		NO, THEN DON'T ISSUE A SERVICE REQUEST		04384

RQS - REQUEST SERVICE

04350

6284	32BA	C320	D925		MOV	TOTLLO,R12	IS 7854 SELECTED AS A TALKER/LISTENER?	04385	
6285	32BE	1628			JNE	NORQS	NO, DON'T SEND SERVICE REQUEST	04386	
6286	32C0	C320	DAE6		MOV	RQSNUM,R12	REQUEST SERVICE NUMBER	04387	
6287	32C4	C200			MOV	R0,R0	SAVE R0	04388	
6288	32C6	D207	0001		LI	R7,1	SET LSB IN R7	04389	
6289	32CA	C00C			MOV	R12,R0	REQUEST SERVICE NUMBER	04390	
6290	32CC	1301			JEQ	*+4	DON'T SHIFT IF RQSNUM=0	04391	
6291	32CE	0A07			SLA	R7,R0	SHIFT BIT TO RSV ENABLE BIT LOCATION	04392	
6292	32D0	C008			MOV	R0,R0	RESTORE R0	04393	
6293	32D2	D32C	3294	NORMAL	MOV3	RSVTBL(R12),R12	SERVICE REQUEST STATUS BYTE	04394	
6294	32D6	C228	D975		MOV	INUSE,R0	IS 7854 CURRENTLY BUSY EXECUTING COMMANDS?	04395	
6295	32DA	1302			JEQ	*+6		04396	
6296	32DC	E320	3380		SOC	RSVBUSY,R12	YES, SET BUSY STATUS BIT	04397	
6297	32E0	C220	D91E		MOV	RSVENBL,R0	RSV ENABLE FLAGS	04398	
6298	32E4	1505			JGT	RSVOFF	B15 = 0 IMPLIES RSV'S OFF (EXCEPT POWER ON)	04399	
6299	32E6	1304			JEQ	RSVOFF		04400	
6300	32E8	2207			COC	R7,R0	IS SELECTED RSV ON?	04401	
6301	32EA	1602			JNE	RSVOFF	NO, ISSUE STATUS BYTE WITH NO RSV	04402	
6302	32EC	E320	3385		SOC	RSV,R12	YES, ISSUE STATUS BYTE WITH RSV (SRQ)	04403	
6303	32F0	2320	3386	RSVOFF	COC	RSV,R12	IS RSV BIT SET?	04404	
6304	32F4	1609			JNE	NORSV	NO, THEN NO SRQ GENERATED	04405	
6305	32F6	C20C			MOV	R12,R0	SAVE STATUS BYTE	04406	
6306	32F8	C320	33CA		MOV	SRQLED,R12		04407	
6307	32FC	1000			S30	0	TURN SRQ LED ON	04408	
6308	32FE	D808	E074		MOV3	R0,R5W	SEND RSV STATUS BYTE TO GPIA	04409	
6309	3302	045B			B	*R11		04410	
6310					*				
6311					*	ONLY R12 CAN BE USED AFTER ENTRY POINT 'RQSOFF'			
6312					*				
6313	3304	D320	E054		RQSOFF	MOV3	R5R,R12	READ CURRENT STATUS BYTE FROM GPIA	04411
6314	3308	4320	3386		NORSV	SZC	RSV,R12	RESET RSV BIT	04412
6315	330C	D80C	E074			MOV3	R12,R5W	SEND -RSV STATUS BYTE TO GPIA	04413
6316	3310	D320	33CA		NORQS	MOV	SRQLED,R12		04414
6317	3314	1E00			SBZ	0	TURN SRQ LED OFF	04415	
6318	3316	04E0	DAE5		CLR	RQSNUM	CLEAR SERVICE REQUEST NUMBER	04416	
6319	331A	045B			B	*R11		04417	

GPIB CONSTANTS

04418

6321

6322

***** TABLE OF GPIB DELIMITERS

6323

6324 331C

0007

D_MTB3L WORD 7

BYTE COUNT OF DELIMITERS IN TABLE

04419

6325 331E

00

CRB BYTE \$0D

04420

6326 331F

0A

LFB BYTE \$0A

04421

6327 3320

20

SPACEB FCC ' '

04422

6328 3321

2C

COMMA3 FCC ','

04423

6329 3322

3B

SEMI3 FCC ';'

04424

6330 3323

3A

COLON3 FCC ':'

04425

6331 3324

40

AT FCC '_A'

04426

6332

6333 3325

1B

ESC3 BYTE \$1B

04427

6334 3326

07

BELL3 BYTE \$07

04428

6335 3327

30

D FCC '0'

04429

6336 3328

59

Y FCC 'Y'

04430

6337 3329

41

ASC FCC 'ASC'

04431

332A

53

332B

43

6338

EVEN

04432

6339 332C

6650

NR2 WORD \$6650

NR2 CONVERSION FORMAT

04433

6340 332E

E640

NR3 WORD \$E640

NR3 CONVERSION FORMAT

04434

***** HARDWARE CONFIGURATION/INTERFACE *****

04435

6342		*								
6343		*	ED DOCUMENTATION OF THE ADDRESSES AND THE FLAG							
6344		*	INPUTS AND OUTPUTS OF THE SYSTEM.							
6345		*			READ	WRITE				
6346	E000	MWRD	EQU	\$E000		X		SYSTEM MODE WORD	04436	
6347	E002	AWRD	EQU	\$E002		X	X	DIGITIZER SET-UP WORDS.	04437	
6348	E004	BWRD	EQU	\$E004			X		04438	
6349	E00A	SAPCNT	EQU	\$E00A			X	SHEEPS COUNTER PRESET	04439	
6350		*	EQU	\$E00C				SPARE		
6351	E00E	K3	EQU	\$E00E		X		KEYBOARD CODE OF THE KEY DOWN	04440	
6352	E010	FP	EQU	\$E010		X	X	FRONT PANEL	04441	
6353	E012	ADXD	EQU	\$E012			X	DISPLAY HORIZONTAL PRELOAD	04442	
6354	E014	ADYD	EQU	\$E014			X	DISPLAY VERTICAL PRELOAD	04443	
6355	E040	R0R	EQU	\$E040		X		G INTERRUPT STATUS REGISTER	04444	
6356	E060	R0W	EQU	\$E060			X	P INTERRUPT "MASK" REGISTER	04445	
6357	E044	R1R	EQU	\$E044		X		I COMMAND STATUS REGISTER	04446	
6358	E064	R1W	EQU	\$E064			X	B UNUSED	04447	
6359	E048	R2R	EQU	\$E048		X		ADDRESS STATUS REGISTER	04448	
6360	E068	R2W	EQU	\$E068			X	ADDRESS MODE REGISTER	04449	
6361	E04C	R3R	EQU	\$E04C		X		A AUXILLARY COMMAND REGISTER	04450	
6362	E06C	R3W	EQU	\$E06C			X	D AUXILLARY COMMAND REGISTER	04451	
6363	E050	R4R	EQU	\$E050		X		D ADDRESS SWITCH REGISTER	04452	
6364	E070	R4W	EQU	\$E070			X	R ADDRESS REGISTER	04453	
6365	E054	R5R	EQU	\$E054		X		E SERIAL POLL REGISTER	04454	
6366	E074	R5W	EQU	\$E074			X	S SERIAL POLL REGISTER	04455	
6367	E058	R5R	EQU	\$E058		X		S COMMAND PASS-THRU REGISTER	04456	
6368	E078	R6W	EQU	\$E078			X	E PARALLEL POLL REGISTER	04457	
6369	E05C	R7R	EQU	\$E05C		X		S DATA IN REGISTER	04458	
6370	E07C	R7W	EQU	\$E07C			X	S DATA OUT REGISTER	04459	
6371		*								
6372		*	THESE ARE THE TEST ADDRESSES							
6373		*								
6374	E210	BP	EQU	\$E210					04460	
6375	E210	MATCH	EQU	BP				SOME PROGRAMS CALL IT MATCH	04461	
6376	E212	TUART	EQU	\$E212					04462	
6377	E312	TUART	EQU	TUART+\$100					04463	
6378	E214	KUART	EQU	\$E214					04464	
6379	E314	KUART	EQU	KUART+\$100					04465	
6380	E216	TIMER	EQU	\$E216					04466	
6381	E218	RTWPI	EQU	\$E218					04467	
6382	E21A	WDSY	EQU	\$E21A					04468	
6383	E21C	PCDSY	EQU	\$E21C					04469	
6384	E21E	STDSY	EQU	\$E21E					04470	
6386		*	CRU OUTPUT FLAGS							
6387		*								
6388		*	THE HARDWARE INITIALIZES THESE TO ALL 0 ON							
6389		*	1. POWER UP							
6390		*	2. WHEN RESET IS PUSHED TO SIMULATE POWER UP TO THE MPU SYSTEM							
6391		*	3. WHEN THE LOAD BUTTON IS PUSHED							
6392		*								
6393		*	FOR THE INTERRUPT RESETS ALL ARE RESET BY SETTING THE FLAG TO							
6394		*	TO '0' AND ENABLED BY SETTING THE FLAG TO '1'. SINCE THE HARDWARE							
6395		*	SETS THEM TO 0 UNDER THE CONDITIONS STATED ABOVE, THEY MUST BE							
6396		*	PUT TO '1' BY THE SOFTWARE BEFORE INTERRUPTS CAN COME THROUGH.							
6397		*								
6398	0000	RJSY	EQU	0				DISPLAY DONE INTERRUPT RESET	04472	

***** HARDWARE CONFIGURATION/INTERFACE *****

04435

6399	0001	STADSY EQU	1	START DISPLAY	04473
6400	0002	STPJSY EQU	2	STOP DISPLAY	04474
6401	0003	RSPIB EQU	3	GPIB RESET	04475
6402	0004	OUTPUT EQU	4	TTL SIGNAL TO REAR PANEL	04476
6403	0005	REMOTE EQU	5	REMOTE/LOCAL LED IN KEYBOARD	04477
6404	0006	W=MLT EQU	6	WAVEFORM LIGHT IN FP INDICATING STORED WAVEFORM	04478
6405	0007	RRTC EQU	7	20 MSEC CLOCK RESET	04479
6406	*	*	8	SPARE	
6407	*	*	9	SPARE	
6408	*	*	10	SPARE	
6409	0003	RTIM EQU	11	TIMER INTERRUPT RESET	04480
6410	000C	RSTBP EQU	12	BREAK POINT LATCH RESET	04481
6411	000D	TIMSPD EQU	13	TIMER SPEED 0=50US/PT 1=5US/PT	04482
6412	000E	TIMEN EQU	14	TIMER ENABLE 0=STOP 1=GO	04483
6413	000F	TIMCLR EQU	15	CLEAR TIMER	04484
6414	0010	K3ID EQU	16	KEYBOARD IDENTIFICATION CODE	04485
6415	0011	BJSY EQU	17	BUSY LED IN KEYBOARD	04486
6416	0012	ERROR EQU	18	ERROR LED IN KEYBOARD	04487
6417	0013	AJDID EQU	19	SPEAKER	04488
6418	0014	ID EQU	20	I/O LED IN KEYBOARD	04489
6419	0015	SRQ EQU	21	SRQ LED IN KEYBOARD	04490
6420	0016	K3RST EQU	22	KEYBOARD INTERRUPT RESET	04491
6421	0017	RJACQEN EQU	23	READOUT ACQUISITION DMA ENABLE	04492

6423 * CRJ INPUTS
6424 * IF THE CRJ INPUT IS A '1' THEN THERE IS 8K OF RAM INSTALLED
6425 * IF IT IS A '0' THEN THERE IS ONLY 4K OF RAM INSTALLED.

6427	*	*			
6428	*	INTERRUPTS AND POSITIONS			
6429	*	*			
6430	0000	PJD EQU	0	POWER UP AND DOWN	04495
6431	0001	I1 EQU	1	TEST ADAPTER FOR EACH INSTRUCTION	04496
6432	0002	IBP EQU	2	BREAKPOINT	04497
6433	0003	ITERM EQU	3	TERMINAL	04498
6434	0004	IKRON EQU	4	KRONOS INTERRUPT	04499
6435	0005	ITIMER EQU	5	TIMER INTERRUPT	04500
6436	*	*	6	IS A SPARE	
6437	*	*	7	IS A SPARE	
6438	*	*			
6439	*	THESE ARE THE SYSTEM INTERRUPTS			
6440	*	THESE ARE JUMPERABLE SO THAT THEY MAY BE SET TO ANY ORDER 8-F			
6441	*	PRESENTLY THEY AR WIRED AS FOLLOWS			
6442	*	*	8	IS A SPARE	
6443	0009	IRTC EQU	9	REAL TIME CLOCK (20MS) INTERRUPT	04501
6444	000A	IJSY EQU	10	DISPLAY IS DONE	04502
6445	000B	IGPIB EQU	11	GPIB INTERRUPT	04503
6446	000C	IKB EQU	12	KEYBOARD INTERRUPT	04504
6447	000D	IAC1 EQU	13	SWEEPS COUNTER HAS 256 SWEEPS MAX	04505
6448	*	*	14	IS A SPARE	
6449	*	*	15	IS A SPARE	
6450	E000	MODE EQU MDWRD			04506
6451	E002	AWRD EQU AWRD			04507
6452	E004	BWRD EQU BWRD			04508
6453	E012	DXAR EQU ADXO			04509

***** HARDWARE CONFIGURATION/INTERFACE *****

04435

6454	E00A	SWCNT	EQU SWPCNT	04510
6455	E014	DYAJR	EQU ADYD	04511
6456	E00E	KBCDJE	EQU KB	04512
6457	E010	FRONT	EQU FP	04513
6458	0016	RSTKB	EQU KBRST	04514
6459		*UNIT	WORD SCE	
6460	0008	TERMINAL	EQU 8	04515
6461	E018	HEXDSY	EQU \$E018	04516
6462	E000	DMDWRD	EQU MDWRD	04517
6463	E00A	AMDWRD	EQU SWPCNT	04518

5465							
6466							
5467							
6468							
5469							
6470							
5471							
6472							
5473							04520
6474	3330	FFF0	CN16	EVEN	WORD -16		04521
6475	3332	FFFC	CN4	WORD -4			04522
6476	3334	FFFE	CN2	WORD -2			04523
6477	3336	FFFF	CN1	WORD -1			04524
6478	3338	0000	C0	WORD 0			04525
6479	333A	0001	C1	WORD 1			04526
6480	333C	0002	C2	WORD 2			04527
6481	333E	0003	C3	WORD 3			04528
6482	3340	0004	C4	WORD 4			04529
6483	3342	0005	C5	WORD 5			04530
6484	3344	0006	C6	WORD 6			04531
6485	3346	0007	C7	WORD 7			04532
6486	3348	0008	C8	WORD 8			04533
6487	334A	0009	C9	WORD 9			04534
6488	334C	000A	C10	WORD 10			04535
6489	334E	000B	C11	WORD 11			04536
6490	3350	000C	C12	WORD 12			04537
6491	3352	000D	C13	WORD 13			04538
6492	3354	000E	C14	WORD 14			04539
6493	3356	000F	C15	WORD 15			04540
6494	3358	0010	C16	WORD 16			04541
6495	335A	0011	C17	WORD 17			04542
6496	335C	0012	C18	WORD 18			04543
6497	335E	0014	C20	WORD 20			04544
6498	3360	001F	C31	WORD 31			04545
6499	3362	0020	C32	WORD 32			04546
6500	3364	0021	C33	WORD 33			04547
6501	3366	0028	C40	WORD 40			04548
6502	3368	0029	C41	WORD 41			04549
6503	336A	0030	C48	WORD 48			04550
6504	336C	0032	C50	WORD 50			04551
6505	336E	0040	C64	WORD 64			04552
6506	3370	0052	C98	WORD 98			04553
6507	3372	0053	C99	WORD 99			04554
6508	3374	0054	C100	WORD 100			04555
6509	3376	0080	C128	WORD 128			04556
6510	3378	0100	C256	WORD 256			04557
6511	337A	0200	C512	WORD 512			04558
6512	337C	0400	C1024	WORD 1024			04559
6513		3348	CH8	EQU C8			04560
6514		3358	C410	EQU C16			04561
6515		3362	C420	EQU C32			04562
6516		336A	CH30	EQU C48			04563
6517		336E	C443	EQU C64			04564
6518		3376	C480	EQU C128			04565
6519		3378	C4100	EQU C256			04566
6520		337A	CH200	EQU C512			04567
6521		337C	C4400	EQU C1024			04568
6522	337E	0800	CH800	WORD \$0800			04569
6523	3380	1000	C41000	WORD \$1000			04570

ROM 04519

6524	3382	2000	C42000	WORD	\$2000	04571	
6525	3384	3000	CH3000	WORD	\$3000	04572	
6526	3386	4000	C44000	WORD	\$4000	04573	
6527	3388	5000	C45000	WORD	\$5000	04574	
6528	338A	6000	C46000	WORD	\$6000	04575	
6529	338C	3FFF	C43FFF	WORD	\$3FFF	04576	
6530	338E	7FFF	C47FFF	WORD	\$7FFF	04577	
6531	3390	8000	C48000	WORD	\$8000	04578	
6532	3392	E001	C4E001	WORD	\$E001	04579	
6533	3394	E002	C4E002	WORD	\$E002	04580	
6534	3396	E004	C4E004	WORD	\$E004	04581	
6535	3398	E008	C4E008	WORD	\$E008	04582	
6536	339A	E011	C4E011	WORD	\$E011	04583	
6537	339C	E022	C4E022	WORD	\$E022	04584	
6538	339E	E044	C4E044	WORD	\$E044	04585	
6539	33A0	E088	C4E088	WORD	\$E088	04586	
6540		3336	C4FF	EQU	CN1	04587	
6541		3336	CHFFFF	EQU	CN1	04588	
6542			*				
6543			* VARIABLES FOR VXP0 - VERTICAL EXPAND				
6544			*				
6545		338C	VXP4X	EQU	CH3FFF	04589	
6546	33A2	C000	VXPMIN	WORD	\$C000	04590	
6547			*				
6548			* FRONT PANEL INPUT CODES				
6549			*				
6550		3350	VERTL	EQU	C12	04591	
6551		334C	VERTALT	EQU	C10	04592	
6552		334E	VERTADD	EQU	C11	04593	
6553		3346	VERTCHP	EQU	C7	04594	
6554		333C	VERTR	EQU	C2	04595	
6555		333E	HORZA	EQU	C3	04596	
6556		3344	HORZALT	EQU	C6	04597	
6557		3354	HORZCHP	EQU	C14	04598	
6558		3352	HORZB	EQU	C13	04599	
6559			*				
6560			* FOLLOWING VARIABLES SEGMENT WFM HEADER SPACE				
6561			*				
6562		3338	VEXP	EQU	C8	04600	
6563		3340	HEXP	EQU	C4	04601	
6564		3348	VOFFAB	EQU	C8	04602	
6565		334C	VSCALD	EQU	C10	04603	
6566		3350	HSCALD	EQU	C12	04604	
6567		3354	DISPLA	EQU	C14	04605	
6568			*				
6569			* INITIALIZED HEADER				
6570			*				
6571			*				
6572			* INITIAL VALUES FOR ACQUIRE/DISPLAY MODE CONTROL WORDS				
6573			*				
6574	33A4	1000	DSPRLTI	WORD	\$1000	04606	
6575	33A6	0800	DSPWFMI	WORD	\$0800	04607	
6576	33A8	0A20	DSPCRSI	WORD	\$0A20	04608	
6577	33AA	1940	DSPROI	WORD	\$1940	04609	
6578			*				
6579			* SYSTEM INTERRUPT FLAGS				
6580			*				
6581		3386	ACQINT	EQU	CH4000	04610	
6582		3382	DSPINT	EQU	CH2000	04611	

6583		337E	GPIBINT	EQU	CH000	GPIB INTERRUPT	04612
6584		337C	KBINT	EQU	CH400	KEYBOARD INTERRUPT	04613
6585		337A	C_KINT	EQU	CH200	20 MILLISECOND CLOCK INTERRUPT	04614
6586			*				
6587			*		READOUT CONTROL CHARACTERS		
6588			*				
6589	33AC	80	NULL	BYTE	\$80	<NULL>	04615
6590	33AD	83	ETX	BYTE	\$83	<END-OF-TEXT>	04616
6591	33AE	8A	LF	BYTE	\$8A	<LINEFEED>	04617
6592	33AF	8D	CR	BYTE	\$8D	<CARRIAGE RETURN>	04618
6593	33B0	9E	RS	BYTE	\$9E	<RESET>	04619
6594	33B1	9F	CRLF	BYTE	\$9F	<CARRIAGE RETURN><LINEFEED>	04620
6595			*				
6596			*		CRU INTERRUPT RESET ADDRESSES		
6597			*				
6598	33B2	0000	DSPRST	WORD	RDSY*2	DISPLAY INTERRUPT RESET	04621
6599	33B4	0002	DSPSTR	WORD	STADSY*2	DISPLAY START	04622
6600	33B6	0004	DSPSTP	WORD	STPDSY*2	DISPLAY STOP	04623
6601	33B8	002C	KYBRST	WORD	KBRST*2	KEYBOARD INTERRUPT RESET	04624
6602	33BA	0006	GPIBE0I	WORD	RGPIB*2	GPIB RESET	04625
6603	33BC	000E	C_KRST	WORD	RRTC*2	20 MILLISECOND REALTIME CLOCK INTERRUPT RESET	04626
6604	33BE	0025	SPEAKER	WORD	AUDIO*2	SPEAKER CONTROL (TOGGLE RATE CONTROLS TONE)	04627
6605	33C0	002E	RJDMA	WORD	ROACQEN*2	REALTIME READOUT DMA CONTROL (0-NO DMA,1-DMA)	04628
6606	33C2	0024	ERRLED	WORD	ERROR*2	ERROR LED (1-ON,0-OFF)	04629
6607	33C4	0022	BJSYLED	WORD	BUSY*2	BUSY LED (1-ON,0-OFF)	04630
6608	33C6	0028	I0LED	WORD	I0*2	I/O LED (1-ON,0-OFF)	04631
6609	33C8	000C	WFMLED	WORD	WFMLT*2	STORED WFM DISPLAY LED	04632
6610	33CA	002A	SRQLED	WORD	SRQ*2	SRQ LED (1-ON,0-OFF)	04633
6611	33CC	000A	REMLED	WORD	REMOTE*2	REMOTE-ONLY LED (1-ON,0-OFF)	04634
6612	33CE	0008	TTL0UT	WORD	OUTPUT*2	TTL OUTPUT (1-+5V,0-0V)	04635
6613			*				
6614			*		GPIB DATA MASKS		
6615			*				
6616	33D0	E000	INTMASK	WORD	\$E000	ENABLE GPIB INTERRUPTS: B0,GET,APT,CMD,3I	04636
6617		3338	ADDRMDE	EQU	C0	HANDSHAKE PROCEDURES	04637
6618		3380	RSVBUSY	EQU	CH1000	BUSY BIT IN SERIAL POLL STATUS BYTE	04638
6619		337A	DCAS	EQU	CH200	DEVICE CLEAR ACTIVE STATE (R1R)	04639
6620		337E	RLC	EQU	CH800	REMOTE/LOCAL STATE CHANGED	04640
6621		3386	RSV	EQU	CH4000	SERVICE REQUEST BIT (R5W)	04641
6622		3380	SS	EQU	CH1000	BUSY STATUS BIT (7854 EIS)	04642
6623		3386	RFD	EQU	CH4000	HOLDOFF RFD ON ALL DATA (R3W)	04643
6624		337C	H0LA	EQU	CH400	COMPLETE HANDSHAKE STOPPED BY RFD HOLDOFF (R20)	04644
6625		3380	DACR	EQU	CH1000	MPU HAS EXAMINED SECONDARY ADDRESS (R3W)	04645
6626		3382	EJILF	EQU	CH2000	<EOI> OR <LF> TERMINATORS (R4R)	04646
6627		3386	TALKER	EQU	CH4000	TALK-ONLY CONFIGURATION (R4R)	04647
6628		338A	LISTENER	EQU	CH6000	LISTEN-ONLY CONFIGURATION (R4R)	04648
6629		3390	RESET	EQU	CH8000	GPIB SOFTWARE RESET (R3W)	04649
6630		3390	MA	EQU	CH8000	MY ADDRESS HAS OCCURRED (R2R)	04650
6631		338A	RLOK	EQU	CH6000	LOCAL LOCKOUT ENABLED (R1R)	04651
6632		3386	TJ	EQU	CH4000	SET TO TALK-ONLY MODE (R2W)	04652
6633		3382	LO	EQU	CH2000	SET TO LISTEN-ONLY MODE (R2W)	04653
6634		3390	ONLINE	EQU	CH8000	ONLINE-OFFLINE SWITCH (R4R)	04654
6635		3386	DAL	EQU	CH4000	DISABLE THE LISTENER (R4W)	04655
6636		3382	DAT	EQU	CH2000	DISABLE THE TALKER (R4W)	04656
6637		3378	BI	EQU	CH100	BYTE HAS BEEN RECEIVED (R0R)	04657
6638		3386	BO	EQU	CH4000	BYTE HAS BEEN OUTPUT (R0R)	04658
6639		3378	APTE	EQU	CH100	ENABLE ADDRESS PASS-THROUGH	04659
6640		337E	APT	EQU	CH800	ADDRESS PASS-THROUGH HAS OCCURRED (R0R)	04660
6641		337C	CHJ	EQU	CH400	SPAS+RLC+-DSEL(DCAS+UUCG+UACG) (R0R)	04661

6642		3382	GET	EQU	CH2000	GROUP EXECUTE TRIGGER HAS OCCURRED (R0R)	04662
6643		337C	LACS	EQU	CH400	LISTENER ACTIVE STATE (R2R)	04663
6644		337E	MSA	EQU	CH800	MY SECONDARY ADDRESS (R3W)	04664
6645		337A	END	EQU	CH200	AN EOI HAS OCCURRED WITH -ATN (R0R)	04665
6646		3382	FEOI	EQU	CH2000	SET EOI TRUE (R3W)	04666
6647		3386	SRQS	EQU	CH4000	BUS IN SERVICE REQUEST STATE (R5R)	04667
6648		3386	REM	EQU	CH4000	REMOTE/LOCAL STATUS BIT (R1R)	04668
6649		337E	TACS	EQU	CH800	TALKER ACTIVE STATE (R2R)	04669
6650		337C	RTL	EQU	CH400	RETURN TO LOCAL MESSAGE (R3R/W)	04670
6651		337C	S2AS	EQU	CH400	SERIAL POLL ACTIVE STATE IS IN EFFECT	04671
6652			*			THE FOLLOWING VARIABLES ARE MASKS TO SET AND RESET	
6653			*			THE ACQUIRE/DISPLAY MODE WORD'S STATUS	
6654			*				
6655			*				
6656		3382	MAACQR	EQU	CH2000	SOC: ACQUISITION ----- SZC: NO ACQUISITION	04672
6657		3380	MMDRLT	EQU	CH1000	SOC: REALTIME DISPLAY --- SZC: NO REALTIME DISPLAY	04673
6658		337E	MADCLC	EQU	CH800	SOC: CALCULATOR DISPLAY - SZC: NO CALC DISPLAY	04674
6659		3380	MWEXCLK	EQU	CH1000	SOC: XTERNAL CLOCK ----- SZC: INTERNAL CLOCK	04675
6660		336E	MA8KHZ	EQU	CH40	SOC: 8KHZ READOUT ----- SZC: BURST READOUT	04676
6661		3362	MAXVSY	EQU	CH20	SOC: X-Y DISPLAY ----- SZC: Y-T DISPLAY	04677
6662		337E	MW2WFM	EQU	CH800	SOC: ACQUIRE 2 WFMS ----- SZC: ACQUIRE 1 WFM	04678
6663		3348	MWVCTR	EQU	CH8	SOC: VECTOR WFMS ----- SZC: DOT WFMS	04679
6664		3340	DLYMODE	EQU	C4	DELAY MODE SWITCH	04680
6665			*				
6666			*			VARIABLES THAT FOLLOW PARTITION MEMORY	
6667			*				
6668			*				
6669			*				
6670		DFA0	STARTUSER	EQU	\$DFA0	START OF USER STACK	04681
6671	33D2	DFG2	TOPJSE	WORD	STARTUSER+34	TOP OF USERS STACK	04682
6672	33D4	DFA0	WSTK	WORD	STARTUSER+0	LAST WORD POPPED FROM USER	04683
6673	33D6	DFA6	XSTK	WORD	STARTUSER+6	NORMAL BOTTOM OF USER	04684
6674	33D8	DFAC	YSTK	WORD	STARTUSER+12	ELEMENT Y ADDRESS OF USER	04685
6675	33DA	DF98	VSTK	WORD	STARTUSER+24	ELEMENT 3 ADDRESS OF USER	04686
6676	33DC	DF8E	TSTK	WORD	STARTUSER+30	ELEMENT T ADDRESS OF USER	04687
6677	33DE	EQ00	SOFTST	WORD	\$EQ00	TOP OF SOFTWARE STACK + 2	04688
6678	33E0	DD00	GND0BAS	WORD	\$DD00	GROUND 0 ACQUISITION ADDRESS	04689
6679	33E2	DE00	GND1BAS	WORD	\$DE00	GROUND 1 ACQUISITION ADDRESS	04690
6680		33E0	GPIB3JF	EQU	GND0BAS		04691
6681	33E4	03E7	ENDLINE	WORD	999	LINE NUMBER FOR END OF USER PROGRAM	04692
6682	33E6	000A	BJFLEN	WORD	10	LENGTH OF INPUT KEY BUFFER	04693
6683	33E8	000A	GJSUBLEN	WORD	10	LENGTH OF GOSUB RETURN POINTER BUFFER	04694
6684			*				
6685			*			FOLLOWING VARIABLES GIVE UNIVERSAL CONSTANTS	
6686			*				
6687	33EA	45	EEXMNU	BYTE	'E		04695
6688	33EB	2E	DECMNJ	BYTE	'.		04696
6689		3382	STOPKEY	EQU	CH2000	ALLOWS 'STOP' DURING KEY EXECUTION	04697
6690	33EC	2D	NEG4NJ	FCC	'-'		04698
6691	33ED	00		BYTE	0		04699
6692	33EE	2B	POS4NJ	FCC	'+'		04700
6693	33EF	00		BYTE	0		04701
6694	33F0	FF01	LEFT3	WORD	\$FF01	FRONT PANEL SELECTION FOR LEFT-8	04702
6695	33F2	FF	ENDFLG	BYTE	\$FF,00	USER PROGRAM END FLAG	04703
	33F3	00					
6696	33F4	0000	CHARSTART	WORD	13	START ON X STACK ON LINE 4 READOUT	04704
6697	33F6	6000	FP12M	WORD	\$6000		04705
6698		3340	FP12E	EQU	C4	FP 12 EXPONENT	04706
6699	33F8	5000	FP20M	WORD	\$5000	FP 20 MANTISSA	04707

ROM

04519

6700	33FA	0005	FP20E	WORD \$0005	FP 20 EXPONENT	04708
6701	33FC	6000	FP64	WORD \$6000	FP 6 MANTISSA	04709
6702		333E	FP6E	EQU C3	FP 5 EXPONENT	04710
6703	33FE	5000	FP10M	WORD \$5000	FP 10 MANTISSA	04711
6704		3340	FP10E	EQU C4	FP10 EXPONENT	04712
6705	3400	7000	FP1000M	WORD \$7000	FP 1000 MANTISSA	04713
6706	3402	000A	FP1000E	WORD \$000A	FP 1000 EXPONENT	04714
6707	3404	4001	FP1M	WORD \$4000	FP 1 MANTISSA	04715
6708		333A	FP1E	EQU C1	FP 1 EXPONENT	04716
6709	3406	0800	OV	WORD \$0800	OVERFLOW STATUS BIT	04717
6710	3408	1800	COV	WORD \$1800	CARRY & OVERFLOW STATUS BITS	04718
6711	340A	5A82	INVSQR2	WORD \$5A82	1/(2**0.5) - 0.707107	04719
6712	340C	2B6C	DCNST	WORD \$2B6C	CONSTANT D - 0.3392334	04720
6713	340E	6F2E	LOGEM	WORD \$6F2E	COMMON LOG OF E	04721
6714		3336	LOGEE	EQU C01		04722
6715	3410	49AF	LN10M	WORD \$49AF	NATURAL LOG OF 10	04723
6716		333C	LN10E	EQU C2		04724
6717	3412	6B60	MAXRATH	WORD \$6B60	MAXIMUM SWEEP RATE / 10 FOR DIGITIZER (.2US)	04725
6718	3414	FFEA	MAXRATE	WORD \$FFEA		04726
6719	3416	1666	S4MAX	WORD \$1666	SCREEN MAXIMUM FOR AUTOSCALE	04727
6720			*ZMAX	WORD \$7FFF	RANGE MAXIMUM FOR AUTOSCALE	04728 DEL
6721			*			00010PATCH
6722			*	PROBLEM #9 - PATCH #10 (1 OF 1)		00010PATCH
6723			*			00010PATCH
6724			*	AUTOSCALE DOESN'T WORK IF THINGS WORK OUT EXACTLY		00010PATCH
6725			*			00010PATCH
6726	3418	7FF0	Z4MAX	WORD \$7FF0	RANGE MAXIMUM FOR AUTOSCALE (<20 DIVS)	00010PATCH
6727			*			00010PATCH
6728			*	END OF PROBLEM #9		00010PATCH
6729			*			00010PATCH
6730	341A	8E60	SQRTA	WORD \$8E60	CONSTANT A FOR SQUARE ROOT ALGORITHM	04729
6731	341C	3300	SQRTB	WORD \$3300	CONSTANT B FOR SQUARE ROOT ALGORITHM	04730
6732	341E	6660	SQRTC	WORD \$6660	CONSTANT C FOR SQUARE ROOT ALGORITHM	04731
6733	3420	B1F0	LOG0.5M	WORD \$B1F0	COMMON LOG OF 0.5	04732
6734		3336	LOG0.5E	EQU C01		04733
6735	3422	A688	LOG0.2M	WORD \$A688	COMMON LOG OF 0.2	04734
6736		3338	LOG0.2E	EQU C0		04735
6737	3424	58B9	LN2MANT	WORD \$58B9	FLOATING APPROX. FOR LN(2)	04736
6738		3338	LN2EXP	EQU C0		04737
6739	3426	B172	LN2	WORD \$B172	FRACTIONAL APPROX. FOR LN(2)	04738
6740		3426	LN01	EQU LN2	LN(2) DOUBLE PRECISION	04739
6741	3428	17F8	LN02	WORD \$17F8		04740
6742	342A	0028	LINELEN	WORD 40	MAXIMUM NUMBER OF CHARACTERS PER READOUT LINE	04741
6743	342C	0030	NUMMNU	WORD \$30	\$30 -- SUBTRACT FROM NUMBER IN ASCII TO GET NUMBER	04742
6744	342E	00	NJLWRD	BYTE 0	TEK-CODE NULL	04743
6745	342F	20	NJLWRDB	BYTE \$20		04744
6746	3430	2020	SPACES	WORD \$2020	ASCII FOR SPACES	04745
6747	3432	00	SPACE	BYTE \$00	ASCII FOR SPACE WITH NEXT BYTE	04746
6748	3433	20	SPACBYT	BYTE \$20	ASCII FOR SPACE	04747
6749	3434	0063	PERCENT	WORD 99	PERCENTAGE FILL FOR ACQUISITION	04748
6750	3436	005F	WARNPER	WORD 95	PERCENTAGE FILL NEEDED FOR NO WARNING	04749
6751		3390	BADPNT	EQU CH0000	BAD ACQUISITION POINT TO FILL BUFFER BEFORE ACQUIRE	04750
6752	3438	9FFF	GTLTHSK	WORD \$9FFF	SZC STATUS MASK TO LOOK AT GT,LT,EQ ONLY	04751
6753		3348	DSPMAX	EQU C8	MAXIMUM NUMBER ARE WAVEFORMS DISPLAYED	04752
6754		337A	FRSTRES	EQU C512	STARTING RESOLUTION OF 512	04753
6755			*			
6756			*	DEFAULT HEADER VALUES		
6757			*			
6758	343A	5000	NJLLHEAD	WORD \$5000	VEXP	04754

6759	343C	0005	WORD	\$0005		04755
6760	343E	5000	WORD	\$5000	HEXP	04756
6761	3440	FFFF	WORD	\$FFFF		04757
6762	3442	0000	WORD	0	VERTICAL ZERO (DIVISIONS)	04758
6763	3444	56	FCC	'V'	ASCII VERTICAL UNITS	04759
	3445	20				
6764	3446	53	FCC	'S'	ASCII HORIZONTAL UNITS	04760
	3447	20				
6765	3449	0000	WORD	0	DISPLAY FLAG	04761
6766	344A	0000	WORD	810	SPARE	04762
6767						
6768						
6769						
6770	345A	0080	WFM2K	WORD 128,2	2K OPTION	04763
	345C	0002				
6771	345E	0100	WORD	256,2		04764
	3460	0002				
6772	3462	0200	WORD	512,2		04765
	3464	0002				
6773	3466	0400	WORD	1024,0		04766
	3468	0000				
6774	346A	0080	WFM4K	WORD 128,16	4K OPTION	04767
	346C	0010				
6775	346E	0100	WORD	256,8		04768
	3470	0008				
6776	3472	0200	WORD	512,4		04769
	3474	0004				
6777	3476	0400	WORD	1024,2		04770
	3478	0002				
6778	347A	0080	WFM8K	WORD 128,40	8K OPTION	04771
	347C	0028				
6779	347E	0100	WORD	256,20		04772
	3480	0014				
6780	3482	0200	WORD	512,10		04773
	3484	000A				
6781	3486	0400	WORD	1024,5		04774
	3488	0005				
6782						
6783						
6784						
6785	348A	0002	MAP2	WORD 2	K RAM	04775
6786	348C	A000	WORD	\$A000	WOADD 2K OPTION	04776
6787	348E	D8E0	WORD	\$D8E0	WOHEAD	04777
6788	3490	A000	WORD	\$A000	WFBAS	04778
6789	3492	D900	WORD	\$D900	TOPHE0	04779
6790	3494	0000	WORD	0	CONSTR	04780
6791	3496	0000	WORD	0	MAXCNS	04781
6792	3498	D800	WORD	\$D800	DISMEM	04782
6793	349A	D8B0	WORD	\$D8B0	PROGHE0	04783
6794	349C	D8B8	WORD	\$D8B8	MAXPROG	04784
6795	349E	345A	WORD	WFM2K	RAMOPT	04785
6796	34A0	0004	MAP4	WORD 4	K RAM	04786
6797	34A2	A000	WORD	\$A000	WOADD 4K OPTION	04787
6798	34A4	D1E0	WORD	\$D1E0	WOHEAD	04788
6799	34A6	A000	WORD	\$A000	WFBAS	04789
6800	34A8	D200	WORD	\$D200	TOPHE0	04790
6801	34AA	D4A0	WORD	\$D4A0	CONSTR	04791
6802	34AC	0032	WORD	50	MAXCNS	04792
6803	34AE	D200	WORD	\$D200	DISMEM	04793

6804	3480	0568	WORD	\$0568	PROGMEM	04794	
6805	3482	08FC	WORD	\$08FC	MAXPROG	04795	
6806	3484	346A	WORD	WFM4K	RAMOPT	04796	
6807	3486	0008	MAP8	WORD	8	K RAM	04797
6808	3488	A000	WORD	\$A000	WOADD	5K OPTION	04798
6809	348A	CCE0	WORD	\$CCE0	WOHEAD		04799
6810	348C	A000	WORD	\$A000	WFM8K		04800
6811	348E	C000	WORD	\$C000	TOPHE0		04801
6812	34C0	CFA0	WORD	\$CFA0	CONSTR		04802
6813	34C2	0064	WORD	100	MAXCNS		04803
6814	34C4	C000	WORD	\$C000	DISMEM		04804
6815	34C6	D130	WORD	\$D130	PROGMEM		04805
6816	34C8	08FC	WORD	\$08FC	MAXPROG		04806
6817	34CA	347A	WORD	WFM8K	RAMOPT		04807
6818			*				
6819	34CC	8100	SYS2K	WORD	\$8100	VALID RAM MAP FOR 2 K SYSTEM	04808
6820	34CE	C300	SYS4K	WORD	\$C300	VALID RAM MAP FOR 4 K SYSTEM	04809
6821	34D0	FF00	SYS8K	WORD	\$FF00	VALID RAM MAP FOR 8 K SYSTEM	04810
6822	34D2	FFF1	PWRERR1	WORD	\$FFF1	POWER UP ERROR # 1	04811
6823	34D4	FFF2	PWRERR2	WORD	\$FFF2	POWER UP ERROR # 2	04812
6824	34D6	FFF3	PWRERR3	WORD	\$FFF3	POWER UP ERROR # 3	04813
6825	34D8	FFF4	PWRERR4	WORD	\$FFF4	POWER UP ERROR # 4	04814
6826			*				
6827	34DA	7800	NOFPKEY	WORD	\$7800	MASK FOR NO FRONT PANEL KEY DEPRESSED	04815
6828			*				
6829			*	THE FOLLOWING ARE DISPLAY FLAGS FOR READOUT SETUP			
6830			*				
6831			*				
6832		333A	CLINE1	EQU	C1	CHANGE IN LINE #1 (OPWFM DATA)	04816
6833		333C	CLINE2	EQU	C2	CHANGE IN LINE #2 (DISPLAYED WFM #S)	04817
6834		3340	CLINE15	EQU	C4	CHANGE IN LINE #15 (CURSOR INFO)	04818
6835		3348	CLINE16	EQU	C8	CHANGE IN LINE #16 (STACK X & Y, KEY, ERROR, PL#)	04819
6836	34DC	00		BYTE	0		04820
6837	34DD	3E	CARRDT	BYTE	'>		04821

6839	34DE	002A	ASTERISK WORD	**	** CHARACTER	04823
6840	34E0	0080	BLANK WORD	\$0080	<NULL>=<BLANK>	04824
6841	34E2	0080	MSIGNS WORD	\$0080	MANTISSA SIGNS <POSITIVE>	04825
6842	34E4	0020	WORD	'-	<NEGATIVE>	04826
6843	34E6	0028	ESIGNS WORD	'+	EXPONENT SIGNS <POSITIVE>	04827
6844	34E8	0020	WORD	'-	<NEGATIVE>	04828
6845	34EA	0048	POSEXP WORD	'K	KILO	04829
6846	34EC	0040	WORD	'M	MEGA	04830
6847	34EE	0047	WORD	'G	GIGGA	04831
6848	34F0	0054	WORD	'T	TERRA	04832
6849	34F2	0060	NEGEXP WORD	\$0060	MILLI	04833
6850	34F4	0013	WORD	\$0013	MICRO	04834
6851	34F6	006E	WORD	\$006E	NANO	04835
6852	34F8	0070	WORD	\$0070	PICO	04836
6853	34FA	002E	PERIOD1 WORD	'.		04837
6854	34FC	0045	E WORD	'E		04838

THE FOLLOWING ARE TABLES USED FOR CONVERTING NUMBERS BETWEEN
 FLOATING POINT BASE 2 AND FLOATING POINT BASE 10.

				POWER	DOUBLE-WORD MANTISSA	EXPONENT	
6859							
6860							
6861							
6862							
6863	34FE 3500	44B8 FFE3	WORD \$44B8,\$FFE3	10** -9	\$44B8 \$2FA1	\$FFE3	04839
6864	3502 3504	55E6 FFE6	WORD \$55E6,\$FFE6	10** -8	\$55E6 \$3B89	\$FFE6	04840
6865	3506 3508	6858 FFE9	WORD \$6858,\$FFE9	10** -7	\$685F \$CA68	\$FFE9	04841
6866	350A 350C	431C FFED	WORD \$431C,\$FFED	10** -6	\$431B \$DE83	\$FFED	04842
6867	350E 3510	53E3 FFFJ	WORD \$53E3,\$FFF0	10** -5	\$53E2 \$D624	\$FFF0	04843
6868	3512 3514	68DC FFF3	WORD \$68DC,\$FFF3	10** -4	\$68DB \$8BAC	\$FFF3	04844
6869	3516 3518	4189 FFF7	WORD \$4189,\$FFF7	10** -3	\$4189 \$374C	\$FFF7	04845
6870	351A 351C	51EC FFFA	WORD \$51EC,\$FFFA	10** -2	\$51EB \$851F	\$FFFA	04846
6871	351E 3520	6665 FFFJ	WORD \$6665,\$FFF0	10** -1	\$6666 \$6666	\$FFF0	04847
6872	3522 3524	4000 0001	ONETBL WORD \$4000,\$0001	10** 0	\$4000 \$0000	\$0001	04848
6873	3526 3528	5000 0004	WORD \$5000,\$0004	10** 1	\$5000 \$0000	\$0004	04849
6874	352A 352C	6400 0007	WORD \$6400,\$0007	10** 2	\$6400 \$0000	\$0007	04850
6875	352E 3530	7008 000A	WORD \$7000,\$000A	10** 3	\$7000 \$0000	\$000A	04851
6876	3532 3534	4E20 000E	WORD \$4E20,\$000E	10** 4	\$4E20 \$0000	\$000E	04852
6877	3536 3538	61A8 0011	WORD \$61A8,\$0011	10** 5	\$61A8 \$0000	\$0011	04853
6878	353A 353C	7A12 0014	WORD \$7A12,\$0014	10** 6	\$7A12 \$0000	\$0014	04854
6879	353E 3540	4C43 0018	WORD \$4C4B,\$0018	10** 7	\$4C4B \$4000	\$0018	04855
6880	3542	5F5E	WORD \$5F5E,\$001B	10** 8	\$5F5E \$1000	\$001B	04856

6881	3544 3546 3548	001B 7736 001E 0028		WORD \$7736,\$001E	10**9	\$7735 \$9400	\$001E	04857
6882			ONESIZE EQU *-ONETBL					04858
6883			TENT3L					
6884	354A 354C	6FFD FEB4		WORD \$6FFD,\$FEB4	10**-100	\$6FFC \$8B92	\$FE34	04859
6885	354E 3550	412F FED6		WORD \$412F,\$FED6	10**-90	\$412F \$6612	\$FED6	04860
6886	3552 3554	4BE3 FEF7		WORD \$4BE3,\$FEF7	10**-80	\$4BE2 \$8050	\$FEF7	04861
6887	3556 3558	5858 FF18		WORD \$5858,\$FF18	10**-70	\$5857 \$A476	\$FF18	04862
6888	355A 355C	66D8 FF39		WORD \$66D8,\$FF39	10**-60	\$66D8 \$12AB	\$FF39	04863
6889	355E 3560	77BA FF5A		WORD \$77BA,\$FF5A	10**-50	\$77B9 \$E928	\$FF5A	04864
6890	3562 3564	45B1 FF7C		WORD \$45B1,\$FF7C	10**-40	\$4530 \$989E	\$FF7C	04865
6891	3566 3568	5121 FF9D		WORD \$5121,\$FF9D	10**-30	\$5121 \$2FFC	\$FF9D	04866
6892	356A 356C	5E73 FF9E		WORD \$5E73,\$FF9E	10**-20	\$5E72 \$8432	\$FF9E	04867
6893	356E 3570	6DF3 FF3F		WORD \$6DF3,\$FFDF	10**-10	\$6DF3 \$7F67	\$FFDF	04868
6894	3572 3574	4000 0031	TENCTR	WORD \$4000,\$0001	10**0	\$4000 \$0000	\$0001	04869
6895	3576 3578	4A81 0022		WORD \$4A81,\$0022	10**10	\$4A81 \$7C80	\$0022	04870
6896	357A 357C	568C 0043		WORD \$568C,\$0043	10**20	\$568C \$75E3	\$0043	04871
6897	357E 3580	64F9 0064		WORD \$64F9,\$0064	10**30	\$64F9 \$64E7	\$0064	04872
6898	3582 3584	758D 0085		WORD \$758D,\$0085	10**40	\$758C \$A7C7	\$0085	04873
6899	3586 3588	446C 00A7		WORD \$446C,\$00A7	10**50	\$446C \$3B16	\$00A7	04874
6900	358A 358C	4FAB 00C8		WORD \$4FAB,\$00C8	10**60	\$4FA7 \$9393	\$00C8	04875
6901	358E 3590	5CB3 00E9		WORD \$5CB8,\$00E9	10**70	\$5CBA \$E85B	\$00E9	04876
6902	3592 3594	6BF4 010A		WORD \$6BF4,\$010A	10**80	\$6BF3 \$8D48	\$010A	04877
6903	3596 3598	7DAC 012B		WORD \$7DAC,\$012B	10**90	\$7DAC \$3C25	\$012B	04878
6904	359A 359C	4927 014D		WORD \$4927,\$014D	10**100	\$4926 \$8496	\$014D	04879
6905		0054	TENSIZE EQU *-TENTBL					04880

6907			*				
6908			*	FORMAT TABLE FOR 4-LINE READOUT			
6909			*				
6910			*	ENTRY ORDER:			
6911			*	1- ADDRESS OF BEGINNING OF FIELD			
6912			*	2- NUMBER OF CHARACTERS IN STRING			
6913			*	3- ASCII STRING TO OUTPUT			
6914			*	4- FORMAT TO BE USED IN CALLING 'FP2ASC'			
6915			*				
6916			*	ENTRIES 2 & 3 ARE OMITTED IF NO STRING OUTPUT			
6917			*	ENTRY 4 IS OMITTED IF NO VALUE OUTPUT			
6918			*	LISTING 4 PRECEDES LISTING 2 IF VALUE IS OUTPUT FIRST			
6919			*				
6920	359E	0001		OPW	WORD 1,1	STARTING POSITION FOR STRING 'OPW NN '	04882
	35A1	0001					
6921	35A2	0004			WORD 4	FOUR CHARACTERS	04883
6922	35A4	4F		FCC	'OPW '	CHARACTER STRING 'OPW '	04884
	35A5	50					
	35A6	57					
	35A7	20					
6923	35A8	2424			WORD \$2424	INTEGER, LEFT JUSTIFIED, 2 DIGITS, 4 SPACES	04885
6924	35AA	0001	VZR		WORD 1,9	STARTING POSITION FOR STRING 'VZR NNNNNN'	04886
	35AC	0009					
6925	35AE	0004			WORD 4	FOUR CHARACTERS	04887
6926	35B0	56		FCC	'VZR '	CHARACTER STRING 'VZR '	04888
	35B1	5A					
	35B2	52					
	35B3	20					
6927	35B4	6446			WORD \$6446	FIXED, LEFT JUSTIFIED, 4 DIGITS, 5 SPACES	04889
6928	35B6	0001	VERT		WORD 1,19	STARTING POSITION FOR VERTICAL SCALE	04890
	35B8	0013					
6929	35BA	E84A			WORD \$E84A	ENGINEERING, 4 DIGITS, 10 SPACES	04891
6930	35BC	0001	HCRZ		WORD 1,30	STARTING POSITION FOR HORIZONTAL SCALE	04892
	35BE	001E					
6931	35C0	E84A			WORD \$E84A	ENGINEERING, 4 DIGITS, 10 SPACES	04893
6932	35C2	0002	DSW		WORD 2,1	STARTING POSITION FOR 'DSW NN NN NN ...'	04894
	35C4	0001					
6933	35C6	0004			WORD 4	FOUR CHARACTERS	04895
6934	35C8	44		FCC	'DSW '	CHARACTER STRING 'DSW '	04896
	35C9	53					
	35CA	57					
	35CB	20					
6935	35CC	2022			WORD \$2022	INTEGER, 2 DIGITS, 2 SPACES	04897
6936	35CE	0002	VS		WORD 2,35	STARTING POSITION FOR STRING 'VS NN'	04898
	35D0	0023					
6937	35D2	0003			WORD 3	THREE CHARACTERS	04899
6938	35D4	56		FCC	'VS '	CHARACTER STRING 'VS '	04900
	35D5	53					
	35D6	20					
	35D7	20					
6939	35D8	2422			WORD \$2422	INTEGER, LEFT JUSTIFIED, 2 DIGITS, 2 SPACES	04901
6940	35DA	000F	VCR		WORD 15,2	STARTING POSITION FOR STRING 'VCRD ='	04902
	35DC	0002					
6941	35DE	0008			WORD 8	EIGHT CHARACTERS	04903
6942	35E0	20		FCC	'VCRD ='	CHARACTER STRING 'VCRD ='	04904
	35E1	56					
	35E2	43					
	35E3	52					
	35E4	44					

	35E5	20					
	35E6	3D					
	35E7	20					
6943	35E8	EC40		WORD	\$EC40	ENG, LEFT JST, LETTER EXP, 4 DGTS, N SPACES	04905
6944	35EA	000F	HCR	WORD	15,22	STARTING POSITION FOR STRING 'HCRD ='	04906
	35EC	0015					
6945	35EE	0008		WORD	8	EIGHT CHARACTERS	04907
6946	35F0	20		FCC	'HCRD = '	CHARACTER STRING 'HCRD ='	04908
	35F1	48					
	35F2	43					
	35F3	52					
	35F4	44					
	35F5	20					
	35F6	3D					
	35F7	20					
6947	35F8	EC40		WORD	\$EC40	ENG, LEFT JST, LETTER EXP, 4 DGTS, N SPACES	04909
6948	35FA	000F	DVCR	WORD	15,2	STARTING POSITION FOR STRING '^VCRD ='	04910
	35FC	0002					
6949	35FE	0008		WORD	8	EIGHT CHARACTERS	04911
6950	3600	0E		BYTE	\$0E	DELTA	04912
6951	3601	56		FCC	'VCRD = '	CHARACTER STRING '^VCRD ='	04913
	3602	43					
	3603	52					
	3604	44					
	3605	20					
	3606	3D					
	3607	20					
6952	3608	EC40		WORD	\$EC40	ENG, LEFT JST, LETTER EXP, 4 DGTS, N SPACES	04914
6953	360A	000F	D-HCR	WORD	15,22	STARTING POSITION FOR STRING '^HCRD ='	04915
	360C	0016					
6954	360E	0008		WORD	8	EIGHT CHARACTERS	04916
6955	3610	0E		BYTE	\$0E	DELTA	04917
6956	3611	48		FCC	'HCRD = '	CHARACTER STRING '^HCRD ='	04918
	3612	43					
	3613	52					
	3614	44					
	3615	20					
	3616	3D					
	3617	20					
6957	3618	EC40		WORD	\$EC40	ENG, LEFT JST, LETTER EXP, 4 DGTS, N SPACES	04919
6958	361A	001D	YSTCK	WORD	16,1	STARTING POSITION FOR Y STACK ELEMENT	04920
	361C	0001					
6959	361E	E84A		WORD	\$E84A	ENGR, LETTER EXP, 4 DIGITS, 10 SPACES	04921
6960	3620	0010	XSTCK	WORD	16,12	STARTING POSITION FOR X STACK ELEMENT	04922
	3622	000C					
6961	3624	E84A		WORD	\$E84A	ENGR, LETTER EXP, 4 DIGITS, 10 SPACES	04923
6962	3626	0010	WFM	WORD	16,1	STARTING POSITION FOR STRING 'WFM'	04924
	3628	0001					
6963	362A	2026		WORD	\$2026	INTEGER, 2 DIGITS, 6 SPACES	04925
6964	362C	0004		WORD	4	FOUR CHARACTERS	04926
6965	362E	20		FCC	'WFM'	CHARACTER STRING 'WFM'	04927
	362F	57					
	3630	46					
	3631	4D					
6966	3632	0010	XWFM	WORD	16,11	STARTING POSITION FOR STRING 'WFM'	04928
	3634	0003					
6967	3636	2027		WORD	\$2027	INTEGER, 2 DIGITS, 6 SPACES	04929
6968	3638	0004		WORD	4	FOUR CHARACTERS	04930
6969	363A	20		FCC	'WFM'	CHARACTER STRING 'WFM'	04931

04881

	363B	57					
	363C	46					
	363D	4D					
6970	363E	0010	KEYMNJ	WORD	16,22	STARTING POSITION FOR KEY MNUEMONIC	04932
	3640	0016					
6971	3642	0005		WORD	5	FIVE CHARACTERS	04933
6972	3644	20		FCC	'		04934
	3645	20					
	3646	20					
	3647	20					
	3648	20					
	3649	20					
6973	364A	0010	ENDSTAT	WORD	16,27	STARTING POSITION FOR STRING 'END'	04935
	364C	0013					
6974	364E	0007		WORD	7	SEVEN CHARACTERS	04936
6975	3650	45		FCC	'END'	CHARACTER STRING 'END'	04937
	3651	4E					
	3652	44					
	3653	20					
	3654	20					
	3655	20					
	3656	20					
	3657	20					
6976	3658	0010	PAUSE	WORD	16,27	STARTING POSITION FOR STRING 'PAUSE'	04938
	365A	0013					
6977	365C	0007		WORD	7	SEVEN CHARACTERS	04939
6978	365E	50		FCC	'PAUSE'	CHARACTER STRING 'PAUSE'	04940
	365F	41					
	3660	55					
	3661	53					
	3662	45					
	3663	20					
	3664	20					
	3665	20					
6979	3666	0010	BUSY1	WORD	16,27	STARTING POSITION FOR STRING 'BUSY'	04941
	3668	0013					
6980	366A	0007		WORD	7	SEVEN CHARACTERS	04942
6981	366C	42		FCC	'BUSY'	CHARACTER STRING 'BUSY'	04943
	366D	55					
	366E	53					
	366F	59					
	3670	20					
	3671	20					
	3672	20					
	3673	20					
6982	3674	0010	ERR	WORD	16,27	STARTING POSITION FOR STRING 'ERROR'	04944
	3676	0013					
6983	3678	0007		WORD	7	SEVEN CHARACTERS	04945
6984	367A	45		FCC	'ERROR'	CHARACTER STRING 'ERROR'	04946
	367B	52					
	367C	52					
	367D	4F					
	367E	52					
	367F	20					
	3680	20					
	3681	20					
6985	3682	0010	WARN	WORD	16,27	STARTING POSITION FOR STRING 'WARNING'	04947
	3684	0013					
6986	3686	0007		WORD	7	SEVEN CHARACTERS	04948

6987	3688	57	FCC	'WARNING'	CHARACTER STRING 'WARNING'	04949
	3689	41				
	368A	52				
	368B	4E				
	368C	49				
	368D	4E				
	368E	47				
	368F	20				
6988	3690	0010	STOPIN	WORD 16,27	STARTING POSITION FOR STRING 'STOP IN'	04950
	3692	001B				
6989	3694	0007		WORD 7	SEVEN CHARACTERS	04951
6990	3696	53	FCC	'STOP IN'	CHARACTER STRING 'STOP IN'	04952
	3697	54				
	3698	4F				
	3699	50				
	369A	20				
	369B	49				
	369C	4E				
	369D	20				
6991	369E	0010	TRUES	WORD 16,27	STARTING POSITION FOR STRING 'TRJE'	04953
	36A0	0013				
6992	36A2	0007		WORD 7		04954
6993	36A4	54	FCC	'TRUE'		04955
	36A5	52				
	36A6	55				
	36A7	45				
	36A8	20				
	36A9	20				
	36AA	20				
	36AB	20				
6994	36AC	0010	FALSES	WORD 16,27	STARTING POSITION FOR STRING 'FALSE'	04956
	36AE	001B				
6995	36B0	0007		WORD 7		04957
6996	36B2	46	FCC	'FALSE'		04958
	36B3	41				
	36B4	4C				
	36B5	53				
	36B6	45				
	36B7	20				
	36B8	20				
	36B9	20				
6997	36BA	0010	BLANKST	WORD 16,27	STARTING POSITION FOR STRING ' '	04959
	36BC	0013				
6998	36BE	0007		WORD 7	SEVEN CHARACTERS	04960
6999	36C0	20	FCC	' '	CHARACTER STRING ' '	04961
	36C1	20				
	36C2	20				
	36C3	20				
	36C4	20				
	36C5	20				
	36C6	20				
	36C7	20				
7000	36C8	0010	BLNKOVR	WORD 16,29	STARTING POSITION FOR STRING ' '	04962
	36CA	0010				
7001	36CC	0005		WORD 5	FIVE CHARACTERS	04963
7002	36CE	20	FCC	' '	CHARACTER STRING ' '	04964
	36CF	20				
	36D0	20				
	36D1	20				

	36D2	20				
	36D3	20				
7003	36D4	0010	PLBLNK	WORD 16,35	STARTING POSITION FOR STRING ' ' ' '	04965
	36D6	0023				
7004	36D8	0005		WORD 6		04966
7005	36DA	20		FCC ' ' ' '		04967
	36DB	20				
	36DC	20				
	36DD	20				
	36DE	20				
	36DF	20				
7006	36E0	0010	PL	WORD 16,34	STARTING POSITION FOR STRING 'PL ' ' '	04968
	36E2	0022				
7007	36E4	0004		WORD 4	FOUR CHARACTERS	04969
7008	36E6	20		FCC ' PL ' ' '	CHARACTER STRING 'PL ' ' '	04970
	36E7	50				
	36E8	4C				
	36E9	20				
7009	36EA	2433		WORD \$2433	INTEGER, LEFT JUST, 3 DIGITS, 3 SPACES	04971
7010						
7011			*			
7012			*	THE FOLLOWING TABLE IS USED TO FORMAT THE 2-LINE READOUT		
7013			*			
7014			*	LISTING ORDER		
7015			*	1- POSITION OF NEXT TEKCODE WORD		
7016			*	LINE #, CHARACTER #		
7017			*			
7018	36EC	0001	ASCII	WORD 1,1		04972
	36EE	0001				
7019	36F0	0010		WORD 16,1		04973
	36F2	0001				
7020	36F4	0001		WORD 1,11		04974
	36F6	0003				
7021	36F8	0010		WORD 16,11		04975
	36FA	0003				
7022	36FC	0001		WORD 1,21		04976
	36FE	0015				
7023	3700	0010		WORD 16,21		04977
	3702	0015				
7024	3704	0001		WORD 1,31		04978
	3706	001F				
7025	3708	0010		WORD 16,31		04979
	370A	001F				
7026		370C	ENDASC EQU *			04980
7027		DF00	TEKCODE EQU \$DF00			04981
7028		DF88	TS10.7887 EQU TEKCODE+139	ADDRESS OF 7887'S READOUT TIMESLOT 10		04982
7029	370C	8D	B)	BYTE \$8D	CODE FOR EXTERNAL 7887 CLOCK	04983
7030	370D	CD	C)	BYTE \$CD	CODE FOR / 1000 7887 CLOCK	04984
7031	370E	DD	D)	BYTE \$DD	CODE FOR NORMAL 7887 CLOCK	04985
7032	370F	00		BYTE 0		04986

TEKCODE TO ASCII CONVERSION TABLE

04987

7034 *
 7035 * THIS TABLE IS USED TO CONVERT FROM TEKCODE TO ASCII
 7036 *
 7037 * THE INDEX INTO THIS TABLE IS THE TEKCODE CHARACTER
 7038 * MINUS A HEXADECIMAL 60.
 7039 *
 7040 * INDEX = TEKCODE - \$0060
 7041 *
 7042 * SPECIAL CONTROL CHARACTERS HAVE BB SET TO 1
 7043 * WHILE NORMAL CHARACTERS HAVE BB SET TO 0.
 7044 *
 7045 * UNDEFINED TEKCODE CHARACTERS ARE FILLED WITH \$80
 7046 *
 7047 * CHARACTER TEKCODE NOTES

TEKCODE	CHARACTER	TEKCODE	NOTES	INDEX
7048				
7049	3710	80	TEKASC BYTE 13:\$80	SPACE-NULL 60-6C 04988
7050	3710	20	BYTE 32	SPACE 6D 04989
7051	371E	80	BYTE 2:\$80	SPACE-NULL 6E-6F 04990
7052	3720	80	BYTE 13:\$80	70-7C 04991
7053	3720	A0	BYTE \$A0	SKIP-NULL 7D 04992
7054	372E	80	BYTE \$80	7E 04993
7055	372F	2E	DECPNT BYTE 46	7F 04994
7056	3730	80	BYTE \$80	80 04995
7057	3731	D7	BYTE \$D7	CTRL 81 DECIMAL PNT LOC 7 04996
7058	3732	D6	BYTE \$D6	CTRL 82 DECIMAL PNT LOC 6 04997
7059	3733	D5	BYTE \$D5	CTRL 83 DECIMAL PNT LOC 5 04998
7060	3734	D4	BYTE \$D4	CTRL 84 DECIMAL PNT LOC 4 04999
7061	3735	D3	BYTE \$D3	CTRL 85 DECIMAL PNT LOC 3 05000
7062	3736	80	BYTE 7:\$80	86-8C 05001
7063	373D	A0	BYTE \$A0	SKIP-NULL 8D 05002
7064	373E	80	BYTE 2:\$80	8E-8F 05003
7065	3740	4A	BYTE 'J	J 90 05004
7066	3741	46	BYTE 'F	F 91 05005
7067	3742	50	BYTE 'P	P 92 05006
7068	3743	59	BYTE 'Y	Y 93 05007
7069	3744	5A	BYTE 'Z	Z 94 05008
7070	3745	4C	BYTE 'L	L 95 05009
7071	3746	4E	BYTE 'N	N 96 05010
7072	3747	55	BYTE 'U	U 97 05011
7073	3748	80	BYTE 5:\$80	98-9C 05012
7074	374D	A0	BYTE \$A0	SKIP-NULL 9D 05013
7075	374E	44	BYTE 'D	D 9E 05014
7076	374F	51	BYTE 'Q	Q 9F 05015
7077	3750	63	BYTE 99	CENTI A0 05016
7078	3751	42	BYTE 'B	B A1 05017
7079	3752	64	BYTE 100	DECI A2 05018
7080	3753	48	BYTE 'H	H A3 05019
7081	3754	57	BYTE 'W	W A4 05020
7082	3755	41	BYTE 'A	A A5 05021
7083	3756	56	BYTE 'V	V A6 05022
7084	3757	53	BYTE 'S	S A7 05023
7085	3758	80	BYTE 5:\$80	A8-AC 05024
7086	375D	A0	BYTE \$A0	SKIP-NULL AD 05025
7087	375E	45	BYTE 'E	E AE 05026
7088	375F	16	BYTE 22	OMEGA AF 05027
7089	3760	47	BYTE 'G	G B0 05028
7090	3761	4D	BYTE 'M	M B1 05029
7091	3762	4B	BYTE 'K	K B2 05030
7092	3763	58	BYTE 'X	X B3 05031

TEKCODE TO ASCII CONVERSION TABLE

04987

7093	3764	70		BYTE 112	PICO	34		05032
7094	3765	6E		BYTE 110	NANO	35		05033
7095	3766	13		BYTE 19	MICRO	36		05034
7096	3767	60		BYTE 109	MILLI	37		05035
7097	3768	80		BYTE 51\$80		38-BC		05036
7098	376D	A0		BYTE \$A0		3D		05037
7099	376E	52		BYTE 'R	R	BE		05038
7100	376F	54		BYTE 'T	T	BF		05039
7101	3770	80		BYTE 41\$80		C0-C3		05040
7102	3774	C5		BYTE \$C5	CTRL	34	REDUCE PREFIX, ADD ONE ZERO	05041
7103	3775	C4		BYTE \$C4	CTRL	C5	REDUCE PREFIX	05042
7104	3776	C2		BYTE \$C2	CTRL	C6	ADD TWO ZEROS	05043
7105	3777	C1		BYTE \$C1	CTRL	C7	ADD ONE ZERO	05044
7106	3778	80		BYTE 51\$80		C8-CC		05045
7107	377D	A0		BYTE \$A0	SKIP-NULL	CD		05046
7108	377E	E0		BYTE \$E0	CTRL	CE	IDENTIFY	05047
7109	377F	80		BYTE \$80		CF		05048
7110	3780	43		BYTE 'C	C	DD		05049
7111	3781	2B	PLUS	BYTE '+	+	D1		05050
7112	3782	2D	MINJS	BYTE '-	-	D2		05051
7113	3783	2B		BYTE '+	+	D3		05052
7114	3784	2F		BYTE '/'	/	D4		05053
7115	3785	49		BYTE 'I	I	D5		05054
7116	3786	3C		BYTE '<	<	D6		05055
7117	3787	1C		BYTE 28	DOWN-ARROW	D7		05056
7118	3788	80		BYTE 51\$80		D8-DC		05057
7119	378D	A0		BYTE \$A0	SKIP-NULL	DD		05058
7120	378E	3E		BYTE '>	>	DE		05059
7121	378F	0E		BYTE 14	DELTA	DF		05060
7122	3790	37	ASCII7	BYTE '7	7	E0		05061
7123	3791	36	ASCII6	BYTE '6	6	E1		05062
7124	3792	35	ASCII5	BYTE '5	5	E2		05063
7125	3793	34	ASCII4	BYTE '4	4	E3		05064
7126	3794	33	ASCII3	BYTE '3	3	E4		05065
7127	3795	32	ASCII2	BYTE '2	2	E5		05066
7128	3796	31	ASCII1	BYTE '1	1	E6		05067
7129	3797	30	ASCII0	BYTE '0	0	E7		05068
7130	3798	80		BYTE 51\$80		E8-EC		05069
7131	379D	A0		BYTE \$A0	SKIP-NULL	ED		05070
7132	379E	39	ASCII9	BYTE '9	9	EE		05071
7133	379F	38	ASCII8	BYTE '8	8	EF		05072
7134	37A0	F0	JJMP	BYTE 161\$F0	JUMP	F0-FF	JUMP	05073
7135								
7136	3780	20	IDENTIFY FCC	' IDENTIFY '				05074
	3781	49						
	3782	44						
	3783	45						
	3784	4E						
	3785	54						
	3786	49						
	3787	46						
	3788	59						
	3789	20						

7138	37BA	0000	MODESW	WORD	0		05076
7139	37BC	0000		WORD	0		05077
7140	37BE	006B		WORD	\$6B	VERTICAL RIGHT	05078
7141	37C0	006E		WORD	\$6E	HORIZONTAL A	05079
7142	37C2	0000		WORD	0		05080
7143	37C4	0000		WORD	0		05081
7144	37C6	006F		WORD	\$6F	HORIZONTAL ALT	05082
7145	37C8	006A		WORD	\$6A	VERTICAL CHOP	05083
7146	37CA	0000		WORD	0		05084
7147	37CC	0000		WORD	0		05085
7148	37CE	0068		WORD	\$68	VERTICAL ALT	05086
7149	37D0	0069		WORD	\$69	VERTICAL ADD	05087
7150	37D2	0060		WORD	\$60	VERTICAL LEFT	05088
7151	37D4	0071		WORD	\$71	HORIZONTAL B	05089
7152	37D6	0072		WORD	\$72	HORIZONTAL CHOP	05090
7153	37D8	0000		WORD	0		05091

Address	Hex	Hex	Description	Address
7155			*	
7156			* TABLE OF EDIT KEYS REQUIRING SPECIAL HANDLING DURING PROGRAM ENTRY	
7157			*	
7158		37DA	SPEC.PROG EQU *	05093
7159	37DA	007D	WORD \$7D	05094
7160	37DC	2C8E	WORD KEYSAVE	05095
7161	37DE	0061	CLLKEY WORD \$61	05096
7162	37E0	4885	WORD KEYCLL	05097
7163	37E2	00E1	CLPKEY WORD \$E1,KEYCLP	05098
	37E4	48DC		
7164	37E6	000D	PROGKEY WORD \$0D,KEYPROG	05099
	37E8	4054		
7165	37EA	0002	GEXEKEY WORD \$02,KEYGEXE	05100
	37EC	4000		
7166	37EE	00	NXTKEY BYTE 0	05101
7167	37EF	63	NXTKEYB BYTE \$63	05102
7168	37F0	4786	WORD KEYNEXT	05103
7169	37F2	00	PREVKEY BYTE 00	05104
7170	37F3	62	PREVKEYB BYTE \$62	05105
7171	37F4	478A	WORD KEYPREV	05106
7172	37F6	00F3	WORD \$F3,KEYF SHIFT OF 'F' IS SAME AS 'F'	05107
	37F8	49C2		
7173	37FA	08	SHIFTKEY BYTE 0	05108
7174	37FB	73	SHIFTKEYB BYTE \$73	05109
7175	37FC	49C2	WORD KEYF	05110
7176	37FE	0009	GPRGKEY WORD \$09,KEYGPROG GPIB 'PROGRAM' COMMAND	05111
	3800	404C		
7177	3802	FFFF	ENDPNT WORD \$FFFF	05112
7178	3804	00	LNNKEY BYTE 0	05113
7179	3805	E2	LNNKEYB BYTE \$E2	05114
7180	3806	0008	NXTKEYG WORD \$0008	05115
7181	3808	000A	STEPKEYG WORD \$000A	05116
7182	380A	0003	NJPKEY WORD \$0003	05117
7183	380C	0010	RJSKEY WORD \$10	05118
7184	380E	0067	RJNKEY WORD \$67	05119
7185	3810	00	PAJSKEY BYTE \$00	05120
7186	3811	81	PAUSKEYB BYTE \$81	05121
7187			*	
7188			*	
7189			* NUMBER TABLE FOLLOWS	
7190			*	
7191			*	
7192		3812	NJMTAB EQU *	05122
7193	3812	001A	WORD \$1A,\$2A,\$2B,\$2C,\$3A,\$33,\$3C,\$4A,\$4B,\$4C	05123
	3814	002A		
	3816	0023		
	3818	002C		
	381A	003A		
	381C	0033		
	381E	003C		
	3820	004A		
	3822	0043		
	3824	004C		
7194	3826	006C	EEXKEY WORD \$6C	05124
7195	3828	001B	DECKEY WORD \$1B	05125
7196	382A	001C	CHSKEY WORD \$1C	05126
7197	382C	FF	ENDKEY BYTE \$FF	05127
7198	382D	FF	ENDKEYB BYTE \$FF	05128
7199	382E	003D	POSKEY WORD \$3D	05129

7200	3830	0040	NEGKEY WORD \$40	05130
7201	3832	00	PANIC BYTE \$00	05131
7202	3833	01	PANICB BYTE \$01	05132
7203	3834	0011	NTRKEY WORD \$11	05133
7204			EVEN	05134

KEY HANDLERS

05135

7206		*			
7207		*	COMPUTED TABLE FOR KEY HANDLERS AND MNEMONIC POINTERS		
7208		*			
7209		*	KEY TRANSFER ADDRESS OBTAINED BY (KEY * 4) + KEYTAB + GOLD		
7210		*	WHERE GOLD = 0 OR 512		
7211		*			
7212		*	MNEMONIC POINTER IN WORD FOLLOWING TRANSFER ADDRESS		
7213		*			
7214		*	NON-EXECUTABLE		
7215		*			
7216			KEYTAB	EQU *	05136
7217	3836		4904	WORD KEYNOP	00
7218	3838		3C56	WORD MNUSTOP	
7219	383A		4904	WORD KEYNOP	01
7220	383C		3C56	WORD MNUSTOP	
7221	383E		4000	WORD KEYGEKE	02
7222	3840		3C7E	WORD MNUEXECUTE	
7223	3842		196C	WORD KEYGND	03
7224	3844		3C64	WORD MNUGND	
7225	3846		1922	WORD KEYAVG	04
7226	3848		3C68	WORD MNUAVG	
7227	384A		1904	WORD KEYAQR	05
7228	384C		3C6C	WORD MNUAQR	
7229	384E		08C8	WORD KEYSTORED	06
7230	3850		3C70	WORD MNUSTORED	
7231	3852		0BA2	WORD KEYSOPE	07
7232	3854		3C78	WORD MNUSCOPE	
7233	3856		4786	WORD KEYGNEXT	08
7234	3858		3DA8	WORD MNUNEXT	
7235	385A		404C	WORD KEYGPRG	09
7236	385C		3C86	WORD MNUPRGS	
7237	385E		476A	WORD KEYSTEP	0A
7238	3860		3C5C	WORD MNUSTEP	
7239	3862		4904	WORD KEYNOP	0B
7240	3864		3C62	WORD MNUNOP	
7241	3866		4904	WORD KEYNOP	0C
7242	3868		3C62	WORD MNUNOP	
7243	386A		4054	WORD KEYPRGS	0D
7244	386C		3C86	WORD MNUPRGS	
7245	386E		4904	WORD KEYNOP	0E
7246	3870		3C62	WORD MNUNOP	
7247	3872		2ACE	WORD KEYTEXT	0F
7248	3874		3C92	WORD MNUTEXT	
7249	3876		2F6A	WORD KEYRQS	10
7250	3878		3C4C	WORD MNURQS	
7251	387A		648A	WORD KEYENTER	11
7252	387C		3C98	WORD MNUENTER	
7253	387E		4F10	WORD KEYLN	12
7254	3880		3C9E	WORD MNULN	
7255	3882		596A	WORD KEYRISE	13
7256	3884		3CA2	WORD MNURISE	
7257	3886		5A3A	WORD KEYDELAY	14
7258	3888		3CA8	WORD MNUDELAY	
7259	388A		5A1A	WORD KEYWIDTH	15
7260	388C		3CAE	WORD MNUWIDTH	
7261	388E		5F86	WORD KEYCRS1LFT	16
7262	3890		3CB4	WORD MNUCRS1LFT	
7263	3892		5F8C	WORD KEYCRS1RGT	17
7264	3894		3CB8	WORD MNUCRS1RGT	

KEY HANDLERS

05135

7265	3896	4934	WORD KEYNOP	18	05185
7266	3898	3C52	WORD MNUNOP		05186
7267	389A	49D4	WORD KEYNOP	19	05187
7268	389C	3C52	WORD MNUNOP		05188
7269	389E	649E	WORD KEY0	1A	05189
7270	38A0	3CC0	WORD MNU0		05190
7271	38A2	649E	WORD KEY.	1B	05191
7272	38A4	3CC2	WORD MNU.		05192
7273	38A6	649E	WORD KEYCHS	1C	05193
7274	38A8	3CC4	WORD MNUCHS		05194
7275	38AA	4A4A	WORD KEYDIV	1D	05195
7276	38AC	3CC8	WORD MNUDIV		05196
7277	38AE	49D4	WORD KEYNOP	1E	05197
7278	38B0	3C62	WORD MNUNOP		05198
7279	38B2	49D4	WORD KEYNOP	1F	05199
7280	38B4	3C62	WORD MNUNOP		05200
7281	38B6	191C	WORD KEYAVG1000	20	05201
7282	38B8	3CCA	WORD MNUAVG1000		05202
7283	38BA	646E	WORD KEYX2Y	21	05203
7284	38BC	3C02	WORD MNUX2Y		05204
7285	38BE	4E5C	WORD KEYABS	22	05205
7286	38C0	3E5E	WORD MNUABS		05206
7287	38C2	5ADA	WORD KEYENERGY	23	05207
7288	38C4	3CDC	WORD MNUENERGY		05208
7289	38C6	5AD4	WORD KEYAREA	24	05209
7290	38C8	3CE4	WORD MNUAREA		05210
7291	38CA	5B12	WORD KEYPER	25	05211
7292	38CC	3CEA	WORD MNUPER		05212
7293	38CE	5DF8	WORD KEYHCRD	26	05213
7294	38D0	3CEE	WORD MNUHCRD		05214
7295	38D2	5E16	WORD KEYVCRD	27	05215
7296	38D4	3CF4	WORD MNUVCRD		05216
7297	38D6	49D4	WORD KEYNOP	28	05217
7298	38D8	3C62	WORD MNUNOP		05218
7299	38DA	49D4	WORD KEYNOP	29	05219
7300	38DC	3C62	WORD MNUNOP		05220
7301	38DE	649E	WORD KEY1	2A	05221
7302	38E0	3CFA	WORD MNU1		05222
7303	38E2	649E	WORD KEY2	2B	05223
7304	38E4	3CFC	WORD MNU2		05224
7305	38E6	649E	WORD KEY3	2C	05225
7306	38E8	3CFE	WORD MNU3		05226
7307	38EA	4A44	WORD KEYMULT	2D	05227
7308	38EC	3D00	WORD MNUMULT		05228
7309	38EE	49D4	WORD KEYNOP	2E	05229
7310	38F0	3C62	WORD MNUNOP		05230
7311	38F2	49D4	WORD KEYNOP	2F	05231
7312	38F4	3C62	WORD MNUNOP		05232
7313	38F6	1916	WORD KEYAVG100	30	05233
7314	38F8	3C3E	WORD MNUAVG100		05234
7315	38FA	644C	WORD KEYCLX	31	05235
7316	38FC	3D02	WORD MNUCLX		05236
7317	38FE	49D4	WORD KEYNOP	32	05237
7318	3900	3C62	WORD MNUNOP		05238
7319	3902	58C4	WORD KEYRMS	33	05239
7320	3904	3D06	WORD MNURMS		05240
7321	3906	58BC	WORD KEYMEAN	34	05241
7322	3908	3D0A	WORD MNUMEAN		05242
7323	390A	5830	WORD KEYMID	35	05243

7324	390C	3D10	WORD MNUMID		05244
7325	390E	641C	WORD KEYP.W	36	05245
7326	3910	3D14	WORD MNUP.W		05246
7327	3912	623C	WORD KEYPNT	37	05247
7328	3914	3D18	WORD MNUPNT		05248
7329	3916	61E0	WORD KEYCNS	38	05249
7330	3918	3D1C	WORD MNUCNS		05250
7331	391A	6162	WORD KEYWFM	39	05251
7332	391C	3D20	WORD MNUWFM		05252
7333	391E	649E	WORD KEY4	3A	05253
7334	3920	3D24	WORD MNU4		05254
7335	3922	649E	WORD KEY5	3B	05255
7336	3924	3D26	WORD MNU5		05256
7337	3926	649E	WORD KEY6	3C	05257
7338	3928	3D28	WORD MNU6		05258
7339	392A	4A50	WORD KEYPLJS	3D	05259
7340	392C	3D2A	WORD MNUPLUS		05260
7341	392E	49D4	WORD KEYNOP	3E	05261
7342	3930	3C62	WORD MNUNOP		05262
7343	3932	49D4	WORD KEYNOP	3F	05263
7344	3934	3C62	WORD MNUNOP		05264
7345	3936	1910	WORD KEYAVG10	40	05265
7346	3938	3C38	WORD MNUAVG10		05266
7347	393A	4F15	WORD KEYSQRT	41	05267
7348	393C	3D2C	WORD MNUSQRT		05268
7349	393E	49D4	WORD KEYNOP	42	05269
7350	3940	3C52	WORD MNUNOP		05270
7351	3942	5826	WORD KEYMAX	43	05271
7352	3944	3D32	WORD MNUMAX		05272
7353	3946	582A	WORD KEYMIN	44	05273
7354	3948	3D36	WORD MNUMIN		05274
7355	394A	5836	WORD KEYP2P	45	05275
7356	394C	3D3A	WORD MNUP2P		05276
7357	394E	0080	WORD KEYVECT	46	05277
7358	3950	3D3E	WORD MNUVECT		05278
7359	3952	0062	WORD KEYTIME	47	05279
7360	3954	3D44	WORD MNU TIME		05280
7361	3956	0CA8	WORD KEYCLW	48	05281
7362	3958	3D4A	WORD MNUCLW		05282
7363	395A	0C3C	WORD KEYDSW	49	05283
7364	395C	3D4E	WORD MNUDSW		05284
7365	395E	649E	WORD KEY7	4A	05285
7366	3960	3D52	WORD MNU7		05286
7367	3962	649E	WORD KEY8	4B	05287
7368	3964	3D54	WORD MNU8		05288
7369	3966	649E	WORD KEY9	4C	05289
7370	3968	3D56	WORD MNU9		05290
7371	396A	4A58	WORD KEYMINUS	4D	05291
7372	396C	3D58	WORD MNUMINUS		05292
7373	396E	2638	WORD KEYOPCOFF	4E	05293
7374	3970	3EEA	WORD MNUOPCOFF		05294
7375	3972	26C8	WORD KEYGEROFF	4F	05295
7376	3974	3EC8	WORD MNUGEROFF		05296
7377	3976	26D8	WORD KEYEXROFF	50	05297
7378	3978	3ECE	WORD MNUEXROFF		05298
7379	397A	49D4	WORD KEYNOP	51	05299
7380	397C	3C52	WORD MNUNOP		05300
7381	397E	54AD	WORD KEYINTG	52	05301
7382	3980	3D5A	WORD MNUINTG		05302

KEY HANDLERS

05135

7383	3982	58A6	WORD KEYORD	53	05303
7384	3984	3D60	WORD MNUORD		05304
7385	3986	53B8	WORD KEYITRP	54	05305
7386	3988	3D64	WORD MNUITRP		05306
7387	398A	5072	WORD KEYSMOOTH	55	05307
7388	398C	3D6A	WORD MNUSMOOTH		05308
7389	398E	519E	WORD KEYHXP0	56	05309
7390	3990	3D7E	WORD MNUHXP0		05310
7391	3992	09BA	WORD KEYVXP0	57	05311
7392	3994	3D84	WORD MNUVXP0		05312
7393	3996	0AB5	WORD KEYVSCL	58	05313
7394	3998	3D8A	WORD MNUVSCL		05314
7395	399A	0AAA	WORD KEYVZR	59	05315
7396	399C	3D90	WORD MNUVZR		05316
7397	399E	0AB0	WORD KEYHSCL	5A	05317
7398	39A0	3D94	WORD MNUHSCL		05318
7399	39A2	26E8	WORD KEYREMOFF	5B	05319
7400	39A4	3EDC	WORD MNUREMOFF		05320
7401	39A6	26F8	WORD KEYIOCOFF	5C	05321
7402	39A8	3EF8	WORD MNUIOCOFF		05322
7403	39AA	2F62	WORD KEYSWH	5D	05323
7404	39AC	3D9A	WORD MNUSWH		05324
7405	39AE	26A8	WORD KEYRQSOFF	5E	05325
7406	39B0	3F06	WORD MNURQSOFF		05326
7407	39B2	49D4	WORD KEYNOP	5F	05327
7408	39B4	3C62	WORD MNUNOP		05328
7409	39B6	0D8E	WORD KEYVMDL	60	05329
7410	39B8	3E28	WORD MNUVMDL		05330
7411	39BA	4886	WORD KEYCLL	61	05331
7412	39BC	3D9E	WORD MNUCLL		05332
7413	39BE	478A	WORD KEYPREV	62	05333
7414	39C0	3DA2	WORD MNUPREV		05334
7415	39C2	4786	WORD KEYNEXT	63	05335
7416	39C4	3DA8	WORD MNUNEXT		05336
7417	39C6	4646	WORD KEYGSB	64	05337
7418	39C8	3DAE	WORD MNUGSB		05338
7419	39CA	464A	WORD KEYGOTO	65	05339
7420	39CC	3DB2	WORD MNUGOTO		05340
7421	39CE	4A68	WORD KEYIFYGTX	66	05341
7422	39D0	3DBC	WORD MNUIFYGTX		05342
7423	39D2	4970	WORD KEYRUN	67	05343
7424	39D4	3C48	WORD MNURUN		05344
7425	39D6	0D9A	WORD KEYVMDALT	68	05345
7426	39D8	3E2E	WORD MNUVMDALT		05346
7427	39DA	0DA6	WORD KEYVMDADD	69	05347
7428	39DC	3E36	WORD MNUVMDADD		05348
7429	39DE	0DB2	WORD KEYVMDCHOP	6A	05349
7430	39E0	3E3E	WORD MNUVMDCHOP		05350
7431	39E2	0D3E	WORD KEYVMDR	6B	05351
7432	39E4	3E46	WORD MNUVMDR		05352
7433	39E6	649E	WORD KEYEEX	6C	05353
7434	39E8	3D02	WORD MNUEEX		05354
7435	39EA	2708	WORD KEYREADX	6D	05355
7436	39EC	3DC6	WORD MNUREADX		05356
7437	39EE	0DCE	WORD KEYHMDA	6E	05357
7438	39F0	3E4C	WORD MNUHMDA		05358
7439	39F2	0DDA	WORD KEYHMDALT	6F	05359
7440	39F4	3E52	WORD MNUHMDALT		05360
7441	39F6	2D42	WORD KEYID	70	05361

7442	39F8	3C8E	WORD MNUID		05362
7443	39FA	0DF2	WORD KEYHMDR	71	05363
7444	39FC	3E52	WORD MNUHMDR		05364
7445	39FE	0DE6	WORD KEYHMDG10P	72	05365
7446	3A00	3E5A	WORD MNUHMDG10P		05366
7447	3A02	49C2	WORD KEYF	73	05367
7448	3A04	3C46	WORD MNUF		05368
7449	3A06	5FA6	WORD KEYGRS2RGT	74	05369
7450	3A08	30DE	WORD MNUCRS2RGT		05370
7451	3A0A	5FA0	WORD KEYGRS2LFT	75	05371
7452	3A0C	3DE4	WORD MNUCRS2LFT		05372
7453	3A0E	5DEC	WORD KEYOFF	76	05373
7454	3A10	3DEA	WORD MNUOFF		05374
7455	3A12	50D0	WORD KEYGRS2.1	77	05375
7456	3A14	3DEE	WORD MNUCRS2.1		05376
7457	3A16	49D4	WORD KEYNOP	78	05377
7458	3A18	3C62	WORD MNUNOP		05378
7459	3A1A	49D4	WORD KEYNOP	79	05379
7460	3A1C	3C62	WORD MNUNOP		05380
7461	3A1E	49D4	WORD KEYNOP	7A	05381
7462	3A20	3C62	WORD MNUNOP		05382
7463	3A22	6096	WORD KEYVPOD	7B	05383
7464	3A24	3DF6	WORD MNUVPOD		05384
7465	3A26	60A0	WORD KEYVPOP	7C	05385
7466	3A28	3DFC	WORD MNUVPOP		05386
7467	3A2A	2C8E	WORD KEYSAVE	7D	05387
7468	3A2C	3E02	WORD MNUSAVE		05388
7469	3A2E	49D4	WORD KEYNOP	7E	05389
7470	3A30	3C62	WORD MNUNOP		05390
7471	3A32	49D4	WORD KEYNOP	7F	05391
7472	3A34	0934	WORD MNUGPIB		05392
7473	3A36	49D4	WORD KEYNOP	80	05393
7474	3A38	3C62	WORD MNUNOP		05394
7475	3A3A	4938	WORD KEYPAUSE	81	05395
7476	3A3C	3C50	WORD MNUPAUSE		05396
7477	3A3E	49D4	WORD KEYNOP	82	05397
7478	3A40	3C62	WORD MNUNOP		05398
7479	3A42	1F18	WORD KEYRDOUT	83	05399
7480	3A44	3E08	WORD MNURDOUT		05400
7481	3A46	49D4	WORD KEYNOP	84	05401
7482	3A48	3C62	WORD MNUNOP		05402
7483	3A4A	195E	WORD KEYAQS	85	05403
7484	3A4C	3E0E	WORD MNUAQS		05404
7485	3A4E	0C02	WORD KEYBOTH	86	05405
7486	3A50	3E12	WORD MNUBOTH		05406
7487	3A52	099E	WORD KEYCYCLE	87	05407
7488	3A54	3C62	WORD MNUNOP		05408
7489	3A56	49D4	WORD KEYNOP	88	05409
7490	3A58	3C62	WORD MNUNOP		05410
7491	3A5A	49D4	WORD KEYNOP	89	05411
7492	3A5C	3C62	WORD MNUNOP		05412
7493	3A5E	49D4	WORD KEYNOP	8A	05413
7494	3A60	3C62	WORD MNUNOP		05414
7495	3A62	49D4	WORD KEYNOP	8B	05415
7496	3A64	3C62	WORD MNUNOP		05416
7497	3A66	49D4	WORD KEYNOP	8C	05417
7498	3A68	3C62	WORD MNUNOP		05418
7499	3A6A	49D4	WORD KEYNOP	8D	05419
7500	3A6C	3C62	WORD MNUNOP		05420

KEY HANDLERS

05135

7501	3A6E	49D4	WORD KEYNOP	8E	05421
7502	3A70	3C62	WORD MNUNOP		05422
7503	3A72	2A1C	WORD KEY2TEXT	8F	05423
7504	3A74	3E18	WORD MNU2TEXT		05424
7505	3A76	49D4	WORD KEYNOP	90	05425
7506	3A78	3C62	WORD MNUNOP		05426
7507	3A7A	49D4	WORD KEYNOP	91	05427
7508	3A7C	3C62	WORD MNUNOP		05428
7509	3A7E	4F0A	WORD KEYEXP	92	05429
7510	3A80	3E1E	WORD MNUEXP		05430
7511	3A82	595C	WORD KEYFALL	93	05431
7512	3A84	3E22	WORD MNUFALL		05432
7513	3A86	49D4	WORD KEYNOP	94	05433
7514	3A88	3C62	WORD MNUNOP		05434
7515	3A8A	49D4	WORD KEYNOP	95	05435
7516	3A8C	3C62	WORD MNUNOP		05436
7517	3A8E	49D4	WORD KEYNOP	96	05437
7518	3A90	3C62	WORD MNUNOP		05438
7519	3A92	49D4	WORD KEYNOP	97	05439
7520	3A94	3C62	WORD MNUNOP		05440
7521	3A96	49D4	WORD KEYNOP	98	05441
7522	3A98	3C62	WORD MNUNOP		05442
7523	3A9A	49D4	WORD KEYNOP	99	05443
7524	3A9C	3C62	WORD MNUNOP		05444
7525		*	WORD KEYNOP	9A	05445 DEL
7526		*			00028PATCH
7527		*	PROBLEM #22 - PATCH #28 (1 OF 8)		00028PATCH
7528		*			00028PATCH
7529		*	INSTALL HOOKS TO ALLOW FUTURE KEYBOARD FUNCTIONS		00028PATCH
7530		*			00028PATCH
7531	3A9E	49D4	WORD KEYNOP	FUTURE SHIFT KEY0 ROUTINE ENTRY ADDRESS	00028PATCH
7532		*			00028PATCH
7533		*	END OF PATCH #28, PROBLEM #22		00028PATCH
7534		*	WORD MNUNOP		05446 DEL
7535		*	WORD KEYNOP	9B	05447 DEL
7536		*	WORD MNUNOP		05448 DEL
7537		*	WORD KEYNOP	9C	05449 DEL
7538		*	WORD MNUNOP		05450 DEL
7539		*			00029PATCH
7540		*	PROBLEM #22 - PATCH #29 (2 OF 8)		00029PATCH
7541		*			00029PATCH
7542		*	INSTALL HOOKS TO ALLOW FUTURE KEYBOARD FUNCTIONS		00029PATCH
7543		*			00029PATCH
7544	3AA0	3C62	WORD MNUNOP	FUTURE SHIFT KEY0 MNEMONIC POINTER	00029PATCH
7545	3AA2	49D4	WORD KEYNOP	FUTURE SHIFT KEY. ROUTINE ENTRY ADDRESS	00029PATCH
7546	3AA4	3C62	WORD MNUNOP	FUTURE SHIFT KEY. MNEMONIC POINTER	00029PATCH
7547	3AA6	49D4	WORD KEYNOP	FUTURE SHIFT KEYCHS ROUTINE ENTRY ADDRESS	00029PATCH
7548	3AA8	3C62	WORD MNUNOP	FUTURE SHIFT KEYCHS MNEMONIC POINTER	00029PATCH
7549		*			00029PATCH
7550		*	END OF PATCH #29, PROBLEM #22		00029PATCH
7551	3AAA	49D4	WORD KEYNOP	9D	05451
7552	3AAC	3C62	WORD MNUNOP		05452
7553	3AAE	49D4	WORD KEYNOP	9E	05453
7554	3AB0	3C62	WORD MNUNOP		05454
7555	3AB2	49D4	WORD KEYNOP	9F	05455
7556	3AB4	3C62	WORD MNUNOP		05456
7557	3AB6	49D4	WORD KEYNOP	A0	05457
7558	3AB8	3C62	WORD MNUNOP		05458
7559	3ABA	6458	WORD KEYROLL	A1	05459

KEY HANDLERS

05135

7560	3ABC	3E68	WORD MNUROLL		05460
7561	3ABE	4E60	WORD KEYSGN	A2	05461
7562	3AC0	3C08	WORD MNUSGN		05462
7563	3AC2	4904	WORD KEYNOP	A3	05463
7564	3AC4	3C62	WORD MNUNOP		05464
7565	3AC6	4904	WORD KEYNOP	A4	05465
7566	3AC8	3C62	WORD MNUNOP		05466
7567	3ACA	5B18	WORD KEYFREQ	A5	05467
7568	3ACC	3E72	WORD MNUFREQ		05468
7569	3ACE	5EEC	WORD KEY2HCRD	A6	05469
7570	3AD0	3E78	WORD MNU2HCRD		05470
7571	3AD2	5E2E	WORD KEY2VCRD	A7	05471
7572	3AD4	3E7E	WORD MNU2VCRD		05472
7573	3AD6	4904	WORD KEYNOP	A8	05473
7574	3AD8	3C62	WORD MNUNOP		05474
7575	3ADA	4904	WORD KEYNOP	A9	05475
7576	3ADC	3C62	WORD MNUNOP		05476
7577			* WORD KEYNOP	AA	05477 DEL
7578			*		00030PATCH
7579			* PROBLEM #22 - PATCH #30 (3 OF 8)		00030PATCH
7580			*		00030PATCH
7581			* INSTALL HOOKS TO ALLOW FUTURE KEYBOARD FUNCTIONS		00030PATCH
7582			*		00030PATCH
7583	3ADE	4904	WORD KEYNOP	FUTURE SHIFT KEY1 ROUTINE ENTRY ADDRESS	00030PATCH
7584			*		00030PATCH
7585			* END OF PATCH #30, PROBLEM #22		00030PATCH
7586			* WORD MNUNOP		05478 DEL
7587			* WORD KEYNOP	AB	05479 DEL
7588			* WORD MNUNOP		05480 DEL
7589			* WORD KEYNOP	AC	05481 DEL
7590			* WORD MNUNOP		05482 DEL
7591			*		00031PATCH
7592			* PROBLEM #22 - PATCH #31 (4 OF 8)		00031PATCH
7593			*		00031PATCH
7594			* INSTALL HOOKS TO ALLOW FUTURE KEYBOARD FUNCTIONS		00031PATCH
7595			*		00031PATCH
7596	3AE0	3C62	WORD MNUNOP	FUTURE SHIFT KEY1 MNEMONIC POINTER	00031PATCH
7597	3AE2	4904	WORD KEYNOP	FUTURE SHIFT KEY2 ROUTINE ENTRY ADDRESS	00031PATCH
7598	3AE4	3C62	WORD MNUNOP	FUTURE SHIFT KEY2 MNEMONIC POINTER	00031PATCH
7599	3AE6	4904	WORD KEYNOP	FUTURE SHIFT KEY3 ROUTINE ENTRY ADDRESS	00031PATCH
7600	3AE8	3C62	WORD MNUNOP	FUTURE SHIFT KEY3 MNEMONIC POINTER	00031PATCH
7601			*		00031PATCH
7602			* END OF PATCH #31, PROBLEM #22		00031PATCH
7603	3AEA	4904	WORD KEYNOP	AD	05483
7604	3AEC	3C62	WORD MNUNOP		05484
7605	3AEE	4904	WORD KEYNOP	AE	05485
7606	3AF0	3C62	WORD MNUNOP		05486
7607	3AF2	4904	WORD KEYNOP	AF	05487
7608	3AF4	3C62	WORD MNUNOP		05488
7609	3AF6	4904	WORD KEYNOP	B0	05489
7610	3AF8	3C62	WORD MNUNOP		05490
7611	3AFA	6432	WORD KEYCLR	B1	05491
7612	3AFC	3E84	WORD MNUCLR		05492
7613	3AFE	4904	WORD KEYNOP	B2	05493
7614	3B00	3C62	WORD MNUNOP		05494
7615	3B02	4904	WORD KEYNOP	B3	05495
7616	3B04	3C62	WORD MNUNOP		05496
7617	3B06	4904	WORD KEYNOP	B4	05497
7618	3B08	3C62	WORD MNUNOP		05498

KEY HANDLERS

05135

7619	3B0A	49D4	WORD KEYNOP	B5	05499
7620	3B0C	3C62	WORD MNUNOP		05500
7621	3B0E	62E4	WORD KEY2P.W	B6	05501
7622	3B10	3E88	WORD MNU2P.W		05502
7623	3B12	626E	WORD KEY2PNT	B7	05503
7624	3B14	3E8E	WORD MNU2PNT		05504
7625	3B16	61F6	WORD KEY2CNS	B8	05505
7626	3B18	3E94	WORD MNU2CNS		05506
7627	3B1A	6194	WORD KEY2WFM	B9	05507
7628	3B1C	3E9A	WORD MNU2WFM		05508
7629		*	WORD KEYNOP	BA	05509 DEL
7630		*			00032PATCH
7631		*	PROBLEM #22 - PATCH #32 (5 OF 8)		00032PATCH
7632		*			00032PATCH
7633		*	INSTALL HOOKS TO ALLOW FUTURE KEYBOARD FUNCTIONS		00032PATCH
7634		*			00032PATCH
7635	3B1E	49D4	WORD KEYNOP	FUTURE SHIFT KEY4 ROUTINE ENTRY ADDRESS	00032PATCH
7636		*			00032PATCH
7637		*	END OF PATCH #32, PROBLEM #22		00032PATCH
7638		*	WORD MNUNOP		05510 DEL
7639		*	WORD KEYNOP	BB	05511 DEL
7640		*	WORD MNUNOP		05512 DEL
7641		*	WORD KEYNOP	BC	05513 DEL
7642		*	WORD MNUNOP		05514 DEL
7643		*			00033PATCH
7644		*	PROBLEM #22 - PATCH #33 (6 OF 8)		00033PATCH
7645		*			00033PATCH
7646		*	INSTALL HOOKS TO ALLOW FUTURE KEYBOARD FUNCTIONS		00033PATCH
7647		*			00033PATCH
7648	3B20	3C62	WORD MNUNOP	FUTURE SHIFT KEY4 MNEMONIC POINTER	00033PATCH
7649	3B22	49D4	WORD KEYNOP	FUTURE SHIFT KEY5 ROUTINE ENTRY ADDRESS	00033PATCH
7650	3B24	3C62	WORD MNUNOP	FUTURE SHIFT KEY5 MNEMONIC POINTER	00033PATCH
7651	3B26	49D4	WORD KEYNOP	FUTURE SHIFT KEY6 ROUTINE ENTRY ADDRESS	00033PATCH
7652	3B28	3C62	WORD MNUNOP	FUTURE SHIFT KEY6 MNEMONIC POINTER	00033PATCH
7653		*			00033PATCH
7654		*	END OF PATCH #33, PROBLEM #22		00033PATCH
7655	3B2A	49D4	WORD KEYNOP	BD	05515
7656	3B2C	3C62	WORD MNUNOP		05516
7657	3B2E	49D4	WORD KEYNOP	BE	05517
7658	3B30	3C62	WORD MNUNOP		05518
7659	3B32	49D4	WORD KEYNOP	BF	05519
7660	3B34	3C62	WORD MNUNOP		05520
7661	3B36	49D4	WORD KEYNOP	C0	05521
7662	3B38	3C62	WORD MNUNOP		05522
7663	3B3A	49D4	WORD KEYNOP	C1	05523
7664	3B3C	3C62	WORD MNUNOP		05524
7665	3B3E	49D4	WORD KEYNOP	C2	05525
7666	3B40	3C62	WORD MNUNOP		05526
7667	3B42	49D4	WORD KEYNOP	C3	05527
7668	3B44	3C62	WORD MNUNOP		05528
7669	3B46	49D4	WORD KEYNOP	C4	05529
7670	3B48	3C62	WORD MNUNOP		05530
7671	3B4A	49D4	WORD KEYNOP	C5	05531
7672	3B4C	3C62	WORD MNUNOP		05532
7673	3B4E	0D72	WORD KEYDOTS	C6	05533
7674	3B50	3EA0	WORD MNUDOTS		05534
7675	3B52	0D2A	WORD KEYVS	C7	05535
7676	3B54	3EA6	WORD MNUVS		05536
7677	3B56	0CFE	WORD KEYCLD	C8	05537

7578	3B58	3EAA	WORD MNUCLJ		05538
7679	3B5A	49D4	WORD KEYNOP	C9	05539
7580	3B5C	3C62	WORD MNUNOP		05540
7681			WORD KEYNOP	CA	05541 DEL
7582			*		00034PATCH
7683			* PROBLEM #22 - PATCH #34 (7 OF 8)		00034PATCH
7684			*		00034PATCH
7685			* INSTALL HOOKS TO ALLOW FUTURE KEYBOARD FUNCTIONS		00034PATCH
7686			*		00034PATCH
7687	3B5E	49D4	WORD KEYNOP	FUTURE SHIFT KEY7 ROUTINE ENTRY ADDRESS	00034PATCH
7688			*		00034PATCH
7689			* END OF PATCH #34, PROBLEM #22		00034PATCH
7690			WORD MNUNOP		05542 DEL
7691			WORD KEYNOP	CB	05543 DEL
7692			WORD MNUNOP		05544 DEL
7693			WORD KEYNOP	CC	05545 DEL
7694			WORD MNUNOP		05546 DEL
7695			*		00035PATCH
7696			* PROBLEM #22 - PATCH #35 (8 OF 8)		00035PATCH
7697			*		00035PATCH
7698			* INSTALL HOOKS TO ALLOW FUTURE KEYBOARD FUNCTIONS		00035PATCH
7699			*		00035PATCH
7700	3B60	3C62	WORD MNUNOP	FUTURE SHIFT KEY7 MNEMONIC POINTER	00035PATCH
7701	3B62	49D4	WORD KEYNOP	FUTURE SHIFT KEY8 ROUTINE ENTRY ADDRESS	00035PATCH
7702	3B64	3C62	WORD MNUNOP	FUTURE SHIFT KEY8 MNEMONIC POINTER	00035PATCH
7703	3B66	49D4	WORD KEYNOP	FUTURE SHIFT KEY9 ROUTINE ENTRY ADDRESS	00035PATCH
7704	3B68	3C62	WORD MNUNOP	FUTURE SHIFT KEY9 MNEMONIC POINTER	00035PATCH
7705			*		00035PATCH
7706			* END OF PATCH #35, PROBLEM #22		00035PATCH
7707	3B6A	49D4	WORD KEYNOP	CD	05547
7708	3B6C	3C62	WORD MNUNOP		05548
7709	3B6E	26C0	WORD KEYOPCON	CE	05549
7710	3B70	3EE4	WORD MNUOPCON		05550
7711	3B72	26D3	WORD KEYGERON	CF	05551
7712	3B74	3EBA	WORD MNUCERON		05552
7713	3B76	26E0	WORD KEYEXRON	D0	05553
7714	3B78	3EC8	WORD MNUEXRON		05554
7715	3B7A	49D4	WORD KEYNOP	D1	05555
7716	3B7C	3C62	WORD MNUNOP		05556
7717	3B7E	5714	WORD KEYDIFF	D2	05557
7718	3B80	3EAE	WORD MNUDIFF		05558
7719	3B82	5C44	WORD KEY2ORD	D3	05559
7720	3B84	3E84	WORD MNU2ORD		05560
7721	3B86	49D4	WORD KEYNOP	D4	05561
7722	3B88	3C62	WORD MNUNOP		05562
7723	3B8A	5642	WORD KEYHPLFT	D5	05563
7724	3B8C	3D72	WORD MNUHPLFT		05564
7725	3B8E	5646	WORD KEYHPRGT	D6	05565
7726	3B90	3D78	WORD MNUHPRGT		05566
7727	3B92	49D4	WORD KEYNOP	D7	05567
7728	3B94	3C62	WORD MNUNOP		05568
7729	3B96	0AD4	WORD KEY2VSCL	D8	05569
7730	3B98	3DCC	WORD MNU2VSCL		05570
7731	3B9A	0B06	WORD KEY2VZR	D9	05571
7732	3B9C	3DD2	WORD MNU2VZR		05572
7733	3B9E	0AC8	WORD KEY2HSCL	DA	05573
7734	3BA0	3DD8	WORD MNU2HSCL		05574
7735	3BA2	26F0	WORD KEYREMON	DB	05575
7736	3BA4	3ED5	WORD MNUREMON		05576

KEY HANDLERS

05135

7737	3BA6	2700	WORD KEYIOCON	DC	05577
7738	3BA8	3EF2	WORD MNUIOCON		05578
7739	3BAA	2F5A	WORD KEYSWL	DD	05579
7740	3BAC	3FDE	WORD MNUSW		05580
7741	3BAE	2680	WORD KEYROSON	DE	05581
7742	3BB0	3F00	WORD MNUROSON		05582
7743	3BB2	4904	WORD KEYNOP	DF	05583
7744	3BB4	3C62	WORD MNUNOP		05584
7745	3BB6	4904	WORD KEYNOP	E0	05585
7746	3BB8	3C62	WORD MNUNOP		05586
7747	3BBA	480C	WORD KEYCLP	E1	05587
7748	3BBC	3F12	WORD MNUCLP		05588
7749	3BBE	4874	WORD KEYLNN	E2	05589
7750	3BC0	3F20	WORD MNULNN		05590
7751	3BC2	4904	WORD KEYNOP	E3	05591
7752	3BC4	3C62	WORD MNUNOP		05592
7753	3BC6	4602	WORD KEYRTN	E4	05593
7754	3BC8	3F16	WORD MNURTN		05594
7755	3BCA	46D2	WORD KEYLBL	E5	05595
7756	3BCC	3DB8	WORD MNULBL		05596
7757	3BCE	4A60	WORD KEYIFXEQY	E6	05597
7758	3BD0	3F24	WORD MNUIFXEQY		05598
7759	3BD2	498C	WORD KEYSTART	E7	05599
7760	3BD4	3F1A	WORD MNUSTART		05600
7761	3BD6	4904	WORD KEYNOP	E8	05601
7762	3BD8	3C62	WORD MNUNOP		05602
7763	3BDA	4904	WORD KEYNOP	E9	05603
7764	3BDC	3C62	WORD MNUNOP		05604
7765	3BDE	4904	WORD KEYNOP	EA	05605
7766	3BE0	3C62	WORD MNUNOP		05606
7767	3BE2	4904	WORD KEYNOP	EB	05607
7768	3BE4	3C62	WORD MNUNOP		05608
7769	3BE6	4904	WORD KEYCLF	EC	05609
7770	3BE8	3C62	WORD MNUNOP		05610
7771	3BEA	2B3C	WORD KEYSNDX	ED	05611
7772	3BEC	3F2E	WORD MNUSENDX		05612
7773	3BEE	4904	WORD KEYNOP	EE	05613
7774	3BF0	3C62	WORD MNUNOP		05614
7775	3BF2	4904	WORD KEYNOP	EF	05615
7776	3BF4	3C62	WORD MNUNOP		05616
7777	3BF6	4904	WORD KEYNOP	F0	05617
7778	3BF8	3C62	WORD MNUNOP		05618
7779	3BFA	4904	WORD KEYNOP	F1	05619
7780	3BFC	3C62	WORD MNUNOP		05620
7781	3BFE	4904	WORD KEYNOP	F2	05621
7782	3C00	3C62	WORD MNUNOP		05622
7783	3C02	49C2	WORD KEYF	F3	05623
7784	3C04	3C46	WORD MNUF		05624
7785	3C06	4904	WORD KEYNOP	F4	05625
7786	3C08	3C62	WORD MNUNOP		05626
7787	3C0A	4904	WORD KEYNOP	F5	05627
7788	3C0C	3C62	WORD MNUNOP		05628
7789	3C0E	4904	WORD KEYNOP	F6	05629
7790	3C10	3C62	WORD MNUNOP		05630
7791	3C12	50C8	WORD KEYCRS1	F7	05631
7792	3C14	3F34	WORD MNUCRS1		05632
7793	3C16	4904	WORD KEYNOP	F8	05633
7794	3C18	3C62	WORD MNUNOP		05634
7795	3C1A	4904	WORD KEYNOP	F9	05635

05135

7796	3C1C	3C62	WORD MNUNOP		05636
7797	3C1E	49D4	WORD KEYNOP	FA	05637
7798	3C20	3C62	WORD MNUNOP		05638
7799	3C22	49D4	WORD KEYNOP	FB	05639
7800	3C24	3C62	WORD MNUNOP		05640
7801	3C26	49D4	WORD KEYNOP	FC	05641
7802	3C28	3C62	WORD MNUNOP		05642
7803	3C2A	49D4	WORD KEYNOP	FD	05643
7804	3C2C	3C62	WORD MNUNOP		05644
7805	3C2E	49D4	WORD KEYCLF	FE	05645
7806	3C30	3F2A	WORD MNUCLF		05646
7807	3C32	49D4	WORD KEYNOP	FF	05647
7808	3C34	3C62	WORD MNUNOP		05648
7809	3C36	FFFF	ENDTAB3	WORD \$FFFF	05649

7811			*			
7812			*	POINTER TO THIS TABLE OBTAINED FROM KEYTAB		
7813			*			
7814			*	POINTER TO FIRST BYTE OF STRING FOR KEY ---		
7815			*	WORD \$FLAG (MSB ON = KEEP MNJ DISPLAYED)		
7816			*	FCC 'STRING' ANY LENGTH STRING MNUM		
7817			*	BYTE 0 ZERO BYTE TERMINATES MNUMONIC		
7818			*	EVEN NEXT MNU IS ON WORD BOUND		
7819			*			
7820		3C38		MNUJTABL EQU *		05651
7821	3C39	41		MNUAVG10 FCC 'AVG10'		05652
	3C39	56				
	3C3A	47				
	3C3B	31				
	3C3C	38				
7822	3C3D	00		BYTE 0		05653
7823	3C3E	41		MNUAVG100 FCC 'AVG100'		05654
	3C3F	56				
	3C40	47				
	3C41	31				
	3C42	30				
	3C43	30				
7824	3C44	0000		WORD 0		05655
7825	3C46	46		MNUF FCC 'F'		05656
7826	3C47	00		BYTE 0		05657
7827	3C48	52		MNURJN FCC 'RUN'		05658
	3C49	55				
	3C4A	4E				
7828	3C4B	00		BYTE 0		05659
7829	3C4C	52		MNURQS FCC 'RQS'		05660
	3C4D	51				
	3C4E	53				
7830	3C4F	00		BYTE 0		05661
7831	3C50	50		MNUPAUSE FCC 'PAUSE'		05662
	3C51	41				
	3C52	55				
	3C53	53				
	3C54	45				
7832	3C55	00		BYTE 0		05663
7833	3C56	53		MNUSTOP FCC 'STOP'		05664
	3C57	54				
	3C58	4F				
	3C59	50				
7834	3C5A	0000		WORD 0		05665
7835	3C5C	53		MNUSTEP FCC 'STEP'		05666
	3C5D	54				
	3C5E	45				
	3C5F	50				
7836	3C60	0000		WORD 0		05667
7837	3C62	20		MNUJCP FCC ' '		05668
7838	3C63	00		BYTE 0		05669
7839	3C64	47		MNUGND FCC 'GND'		05670
	3C65	4E				
	3C66	44				
7840	3C67	00		BYTE 0		05671
7841	3C68	41		MNUAVG FCC 'AVG'		05672
	3C69	56				
	3C6A	47				
7842	3C6B	00		BYTE 0		05673

MNUMONIC TABLE

05550

7843	3C6C	41	MNUAQR	FCC	'AQR'	05674
	3C6D	51				
	3C6E	52				
7844	3C6F	00		BYTE	0	05675
7845	3C70	53	MNUSTORED	FCC	'STORED'	05676
	3C71	54				
	3C72	4F				
	3C73	52				
	3C74	45				
	3C75	44				
7846	3C76	0000		WORD	0	05677
7847	3C78	53	MNUSCOPE	FCC	'SCOPE'	05678
	3C79	43				
	3C7A	4F				
	3C7B	50				
	3C7C	45				
7848	3C7D	00		BYTE	0	05679
7849	3C7E	45	MNUEXECUTE	FCC	'EXECUTE'	05680
	3C7F	58				
	3C80	45				
	3C81	43				
	3C82	55				
	3C83	54				
	3C84	45				
7850	3C85	00		BYTE	0	05681
7851	3C86	50	MNUPRG	FCC	'PROGRAM'	05682
	3C87	52				
	3C88	4F				
	3C89	47				
	3C8A	52				
	3C8B	41				
	3C8C	4D				
7852	3C8D	00		BYTE	0	05683
7853	3C8E	49	MNUID	FCC	'ID'	05684
	3C8F	44				
7854	3C90	0000		WORD	0	05685
7855	3C92	54	MNUTEXT	FCC	'TEXT'	05686
	3C93	45				
	3C94	58				
	3C95	54				
7856	3C96	0000		WORD	0	05687
7857	3C98	45	MNUENTER	FCC	'ENTER'	05688
	3C99	4E				
	3C9A	54				
	3C9B	45				
	3C9C	52				
7858	3C9D	00		BYTE	0	05689
7859	3C9E	4C	MNULN	FCC	'LN'	05690
	3C9F	4E				
7860	3CA0	0000		WORD	0	05691
7861	3CA2	52	MNURISE	FCC	'RISE'	05692
	3CA3	49				
	3CA4	53				
	3CA5	45				
7862	3CA6	0000		WORD	0	05693
7863	3CA8	44	MNUDELAY	FCC	'DELAY'	05694
	3CA9	45				
	3CAA	4C				
	3CAB	41				

	3CAC	59				
7864	3CAD	00		3YTE	0	05695
7865	3CAE	57	MNUWIDTH	FCC	'WIDTH'	05696
	3CAF	49				
	3CB0	44				
	3CB1	54				
	3CB2	48				
7866	3CB3	00		3YTE	0	05697
7867	3CB4	43	MNUCRS1LFT	FCC	'CRS1<'	05698
	3CB5	52				
	3CB6	53				
	3CB7	31				
	3CB8	3C				
7868	3CB9	00		3YTE	0	05699
7869	3CBA	43	MNUCRS1RGT	FCC	'CRS1>'	05700
	3CBB	52				
	3CBC	53				
	3CBD	31				
	3CBE	3E				
7870	3CBF	00		3YTE	0	05701
7871	3CC0	30	MNU0	FCC	'0'	05702
7872	3CC1	00		3YTE	0	05703
7873	3CC2	2E	MNU.	FCC	'.'	05704
7874	3CC3	00		3YTE	0	05705
7875	3CC4	43	MNUCHS	FCC	'CHS'	05706
	3CC5	48				
	3CC6	53				
7876	3CC7	00		3YTE	0	05707
7877	3CC8	2F	MNUJDIV	FCC	'/'	05708
7878	3CC9	00		3YTE	0	05709
7879	3CCA	41	MNUAVG1000	FCC	'AVG1000'	05710
	3CCB	56				
	3CCC	47				
	3CCD	31				
	3CCE	30				
	3CCF	30				
	3CD0	30				
7880	3CD1	00		3YTE	0	05711
7881	3CD2	58	MNUX2Y	FCC	'X<>Y'	05712
	3CD3	3C				
	3CD4	3E				
	3CD5	59				
7882	3CD6	0000		WORD	0	05713
7883	3CD8	53	MNUJSGN	FCC	'SGN'	05714
	3CD9	47				
	3CDA	4E				
7884	3CDB	00		3YTE	0	05715
7885	3CDC	45	MNUENERGY	FCC	'ENERGY'	05716
	3CDD	4E				
	3CDE	45				
	3CDF	52				
	3CE0	47				
	3CE1	59				
7886	3CE2	0000		WORD	0	05717
7887	3CE4	41	MNUAREA	FCC	'AREA'	05718
	3CE5	52				
	3CE6	45				
	3CE7	41				
7888	3CE8	0000		WORD	0	05719

MNUMONIC TABLE

05650

7889	3CEA	50	MNUPER	FCC	'PER'	05720
	3CEB	45				
	3CEC	52				
7890	3CED	00		BYTE	0	05721
7891	3CEE	48	MNUHCRO	FCC	'HCRO'	05722
	3CEF	43				
	3CF0	52				
	3CF1	44				
7892	3CF2	0000		WORD	0	05723
7893	3CF4	56	MNUVCRD	FCC	'VCRD'	05724
	3CF5	43				
	3CF6	52				
	3CF7	44				
7894	3CF8	0000		WORD	0	05725
7895	3CFA	31	MNU1	FCC	'1'	05726
7896	3CFB	00		BYTE	0	05727
7897	3CFC	32	MNU2	FCC	'2'	05728
7898	3CFD	00		BYTE	0	05729
7899	3CFE	33	MNU3	FCC	'3'	05730
7900	3CFF	00		BYTE	0	05731
7901	3D00	2A	MNUMULT	FCC	'*'	05732
7902	3D01	00		BYTE	0	05733
7903	3D02	43	MNUCLX	FCC	'CLX'	05734
	3D03	4C				
	3D04	58				
7904	3D05	00		BYTE	0	05735
7905	3D06	52	MNURMS	FCC	'RMS'	05736
	3D07	40				
	3D08	53				
7906	3D09	00		BYTE	0	05737
7907	3D0A	40	MNUMEAN	FCC	'MEAN'	05738
	3D0B	45				
	3D0C	41				
	3D0D	4E				
7908	3D0E	0000		WORD	0	05739
7909	3D10	40	MNUJMD	FCC	'JMD'	05740
	3D11	49				
	3D12	44				
7910	3D13	00		BYTE	0	05741
7911	3D14	50	MNU2.W	FCC	'P/W'	05742
	3D15	2F				
	3D16	57				
7912	3D17	00		BYTE	0	05743
7913	3D18	50	MNUPNT	FCC	'PNT'	05744
	3D19	4E				
	3D1A	54				
7914	3D1B	00		BYTE	0	05745
7915	3D1C	43	MNUCNS	FCC	'CNS'	05746
	3D1D	4E				
	3D1E	53				
7916	3D1F	00		BYTE	0	05747
7917	3D20	57	MNUWFM	FCC	'WFM'	05748
	3D21	46				
	3D22	4D				
7918	3D23	00		BYTE	0	05749
7919	3D24	34	MNU4	FCC	'4'	05750
7920	3D25	00		BYTE	0	05751
7921	3D26	35	MNU5	FCC	'5'	05752
7922	3D27	00		BYTE	0	05753

MNUMONIC TABLE

05650

7923	3028	36	MNU6	FCC	'6'	05754
7924	3029	00		BYTE	0	05755
7925	302A	28	MNUPLUS	FCC	'+'	05756
7926	302B	00		BYTE	0	05757
7927	302C	53	MNUSQRT	FCC	'SQRT'	05758
	302D	51				
	302E	52				
	302F	54				
7928	3030	0000		WORD	0	05759
7929	3032	4D	MNUMAX	FCC	'MAX'	05760
	3033	41				
	3034	58				
7930	3035	00		BYTE	0	05761
7931	3036	4D	MNUMIN	FCC	'MIN'	05762
	3037	49				
	3038	4E				
7932	3039	00		BYTE	0	05763
7933	303A	50	MNUP2P	FCC	'P-P'	05764
	303B	2D				
	303C	58				
7934	303D	00		BYTE	0	05765
7935	303E	56	MNUVECT	FCC	'VECT'	05766
	303F	45				
	3040	43				
	3041	54				
7936	3042	0000		WORD	0	05767
7937	3044	54	MNUTIME	FCC	'TIME'	05768
	3045	49				
	3046	4D				
	3047	45				
7938	3048	0000		WORD	0	05769
7939	304A	43	MNJCLW	FCC	'CLW'	05770
	304B	4C				
	304C	57				
7940	304D	00		BYTE	0	05771
7941	304E	44	MNUJSH	FCC	'DSW'	05772
	304F	53				
	3050	57				
7942	3051	00		BYTE	0	05773
7943	3052	37	MNU7	FCC	'7'	05774
7944	3053	00		BYTE	0	05775
7945	3054	38	MNU8	FCC	'8'	05776
7946	3055	00		BYTE	0	05777
7947	3056	39	MNU9	FCC	'9'	05778
7948	3057	00		BYTE	0	05779
7949	3058	2D	MNUMINUS	FCC	'-'	05780
7950	3059	00		BYTE	0	05781
7951	305A	49	MNUINTG	FCC	'INTG'	05782
	305B	4E				
	305C	54				
	305D	47				
7952	305E	0000		WORD	0	05783
7953	3060	4F	MNJORD	FCC	'ORD'	05784
	3061	52				
	3062	44				
7954	3063	00		BYTE	0	05785
7955	3064	49	MNUITRP	FCC	'ITRP'	05786
	3065	54				
	3066	52				

MNUMONIC TABLE

05650

	3D67	50				
7956	3D68	0000		WORD	0	05787
7957	3D6A	53	MNUSMOOTH	FCC	'SMOOTH'	05788
	3D6B	40				
	3D6C	4F				
	3D6D	4F				
	3D6E	54				
	3D6F	48				
7958	3D70	0000		WORD	0	05789
7959	3D72	48	MNUHPLFT	FCC	'4PLFT'	05790
	3D73	50				
	3D74	4C				
	3D75	46				
	3D76	54				
7960	3D77	00		BYTE	0	05791
7961	3D78	48	MNUHPRGT	FCC	'HPRGT'	05792
	3D79	50				
	3D7A	52				
	3D7B	47				
	3D7C	54				
7962	3D7D	00		BYTE	0	05793
7963	3D7E	48	MNUHXPD	FCC	'HXPD'	05794
	3D7F	58				
	3D80	50				
	3D81	44				
7964	3D82	0000		WORD	0	05795
7965	3D84	56	MNUVXPD	FCC	'VXPD'	05796
	3D85	58				
	3D86	50				
	3D87	44				
7966	3D88	0000		WORD	0	05797
7967	3D8A	56	MNUVSCL	FCC	'VSCL'	05798
	3D8B	53				
	3D8C	43				
	3D8D	4C				
7968	3D8E	0000		WORD	0	05799
7969	3D90	56	MNUVZR	FCC	'VZR'	05800
	3D91	5A				
	3D92	52				
7970	3D93	00		BYTE	0	05801
7971	3D94	48	MNUHSCL	FCC	'HSCL'	05802
	3D95	53				
	3D96	43				
	3D97	4C				
7972	3D98	0000		WORD	0	05803
7973	3D9A	53	MNUJSH	FCC	'SH'	05804
	3D9B	57				
	3D9C	48				
7974	3D9D	00		BYTE	0	05805
7975	3D9E	43	MNUCLL	FCC	'CLL'	05806
	3D9F	4C				
	3DA0	4C				
7976	3DA1	00		BYTE	0	05807
7977	3DA2	50	MNUPREV	FCC	'PREV'	05808
	3DA3	52				
	3DA4	45				
	3DA5	56				
7978	3DA6	0000		WORD	0	05809
7979	3DA8	4E	MNUNEXT	FCC	'NEXT'	05810

MNUMONIC TABLE

05650

	3DA9	45				
	3DAA	58				
7980	3DAB	54				
	3DAC	0000		WORD	0	05811
7981	3DAE	47	MNUGS3	FCC	'GSB'	05812
	3DAF	53				
	3DB0	42				
7982	3DB1	00		BYTE	0	05813
7983	3DB2	47	MNUGOTO	FCC	'GOTO'	05814
	3DB3	4F				
	3DB4	54				
	3DB5	4F				
7984	3DB6	0000		WORD	0	05815
7985	3DB8	4C	MNJL3.	FCC	'L3L'	05816
	3DB9	42				
	3DBA	4C				
7986	3DBB	00		BYTE	0	05817
7987	3DBC	49	MNUIFYGTX	FCC	'IFY>X'	05818
	3DBD	46				
	3DBE	59				
	3DBF	3E				
	3DC0	58				
7988	3DC1	00		BYTE	0	05819
7989	3DC2	45	MNUEEK	FCC	'EEX'	05820
	3DC3	45				
	3DC4	58				
7990	3DC5	00		BYTE	0	05821
7991	3DC6	52	MNUREADX	FCC	'READX'	05822
	3DC7	45				
	3DC8	41				
	3DC9	44				
	3DCA	58				
7992	3DCB	00		BYTE	0	05823
7993	3DCC	3E	MNU2VSCL	FCC	'>VSCL'	05824
	3DCD	56				
	3DCE	53				
	3DCF	43				
	3DD0	4C				
7994	3DD1	00		BYTE	0	05825
7995	3DD2	3E	MNU2VZR	FCC	'>VZR'	05826
	3DD3	56				
	3DD4	5A				
	3DD5	52				
7996	3DD6	0000		WORD	0	05827
7997	3DD8	3E	MNU2HSCL	FCC	'>4SCL'	05828
	3DD9	48				
	3DDA	53				
	3DDB	43				
	3DDC	4C				
7998	3DDD	00		BYTE	0	05829
7999	3DDE	43	MNUCRS2RGT	FCC	'CRS2>'	05830
	3DDF	52				
	3DE0	53				
	3DE1	32				
	3DE2	3E				
8000	3DE3	00		BYTE	0	05831
8001	3DE4	43	MNUCRS2LFT	FCC	'CRS2<'	05832
	3DE5	52				
	3DE6	53				

	3DE7	32				
	3DE8	3C				
8002	3DE9	00		BYTE	0	05833
8003	3DEA	4F	MNUOFF	FCC	'OFF'	05834
	3DEB	46				
	3DEC	46				
8004	3DED	00		BYTE	0	05835
8005	3DEE	43	MNUCRS2.1	FCC	'CRS2-1'	05836
	3DEF	52				
	3DF0	53				
	3DF1	32				
	3DF2	20				
	3DF3	31				
8006	3DF4	0000		WORD	0	05837
8007	3DF6	56	MNUVPJN	FCC	'VPJN'	05838
	3DF7	50				
	3DF8	44				
	3DF9	4E				
8008	3DFA	0000		WORD	0	05839
8009	3DFC	56	MNUVPJP	FCC	'VPJP'	05840
	3DFD	50				
	3DFE	55				
	3DFE	50				
8010	3E00	0000		WORD	0	05841
8011	3E02	53	MNUSAVE	FCC	'SAVE'	05842
	3E03	41				
	3E04	56				
	3E05	45				
8012	3E06	0000		WORD	0	05843
8013	3E08	52	MNURDOUT	FCC	'RDOUT'	05844
	3E09	44				
	3E0A	4F				
	3E0B	55				
	3E0C	54				
8014	3E0D	00		BYTE	0	05845
8015	3E0E	41	MNUAQS	FCC	'AQS'	05846
	3E0F	51				
	3E10	53				
8016	3E11	00		BYTE	0	05847
8017	3E12	42	MNUBOTH	FCC	'BOTH'	05848
	3E13	4F				
	3E14	54				
	3E15	48				
8018	3E16	0000		WORD	0	05849
8019	3E18	3E	MNU2TEXT	FCC	'>TEXT'	05850
	3E19	54				
	3E1A	45				
	3E1B	58				
	3E1C	54				
8020	3E1D	00		BYTE	0	05851
8021	3E1E	45	MNUEXP	FCC	'EXP'	05852
	3E1F	58				
	3E20	50				
8022	3E21	00		BYTE	0	05853
8023	3E22	46	MNUFALL	FCC	'FALL'	05854
	3E23	41				
	3E24	4C				
	3E25	4C				
8024	3E26	0000		WORD	0	05855

MNUMONIC TABLE

05650

8025	3E28	56	MNUVMJL	FCC	'VMDL'	05856
	3E29	4D				
	3E2A	44				
	3E2B	4C				
8026	3E2C	0000		WORD	0	05857
8027	3E2E	56	MNUVMDALT	FCC	'VMDALT'	05858
	3E2F	4D				
	3E30	44				
	3E31	41				
	3E32	4C				
	3E33	54				
8028	3E34	0000		WORD	0	05859
8029	3E36	56	MNUVMDADD	FCC	'VMDADD'	05860
	3E37	4D				
	3E38	44				
	3E39	41				
	3E3A	44				
	3E3B	44				
8030	3E3C	0000		WORD	0	05861
8031	3E3E	56	MNUVMDCHOP	FCC	'VMDCHOP'	05862
	3E3F	4D				
	3E40	44				
	3E41	43				
	3E42	48				
	3E43	4F				
	3E44	50				
8032	3E45	00		BYTE	0	05863
8033	3E46	56	MNUVMDR	FCC	'VMOR'	05864
	3E47	4D				
	3E48	44				
	3E49	52				
8034	3E4A	0000		WORD	0	05865
8035	3E4C	48	MNUHMDA	FCC	'HMDA'	05866
	3E4D	4D				
	3E4E	44				
	3E4F	41				
8036	3E50	0000		WORD	0	05867
8037	3E52	48	MNUHMDALT	FCC	'HMDALT'	05868
	3E53	4D				
	3E54	44				
	3E55	41				
	3E56	4C				
	3E57	54				
8038	3E58	0000		WORD	0	05869
8039	3E5A	48	MNUHMDCHOP	FCC	'HMDCHOP'	05870
	3E5B	4D				
	3E5C	44				
	3E5D	43				
	3E5E	48				
	3E5F	4F				
	3E60	50				
8040	3E61	00		BYTE	0	05871
8041	3E62	48	MNUHMDB	FCC	'HMDB'	05872
	3E63	4D				
	3E64	44				
	3E65	42				
8042	3E66	0000		WORD	0	05873
8043	3E68	52	MNUROLL	FCC	'ROLL'	05874
	3E69	4F				

	3E6A	4C				
	3E6B	4C				
8044	3E6C	0000		WORD	0	05875
8045	3E6E	41	MNUABS	FCC	'ABS'	05876
	3E6F	42				
	3E70	53				
8046	3E71	00		BYTE	0	05877
8047	3E72	46	MNUFREQ	FCC	'FREQ'	05878
	3E73	52				
	3E74	45				
	3E75	51				
8048	3E76	0000		WORD	0	05879
8049	3E78	3E	MNU2HCRD	FCC	'>HCRD'	05880
	3E79	48				
	3E7A	43				
	3E7B	52				
	3E7C	44				
8050	3E7D	00		BYTE	0	05881
8051	3E7E	3E	MNU2VCRD	FCC	'>VCRD'	05882
	3E7F	56				
	3E80	43				
	3E81	52				
	3E82	44				
8052	3E83	00		BYTE	0	05883
8053	3E84	43	MNUCLR	FCC	'CLS'	05884
	3E85	4C				
	3E86	53				
8054	3E87	00		BYTE	0	05885
8055	3E88	3E	MNU2P.W	FCC	'>P/W'	05886
	3E89	50				
	3E8A	2F				
	3E8B	57				
8056	3E8C	0000		WORD	0	05887
8057	3E8E	3E	MNU2PNT	FCC	'>PNT'	05888
	3E8F	50				
	3E90	4E				
	3E91	54				
8058	3E92	0000		WORD	0	05889
8059	3E94	3E	MNU2CNS	FCC	'>CNS'	05890
	3E95	43				
	3E96	4E				
	3E97	53				
8060	3E98	0000		WORD	0	05891
8061	3E9A	3E	MNU2WFM	FCC	'>WFM'	05892
	3E9B	57				
	3E9C	46				
	3E9D	4D				
8062	3E9E	0000		WORD	0	05893
8063	3EA0	44	MNUDOTS	FCC	'DOTS'	05894
	3EA1	4F				
	3EA2	54				
	3EA3	53				
8064	3EA4	0000		WORD	0	05895
8065	3EA6	56	MNUVS	FCC	'VS'	05896
	3EA7	53				
8066	3EA8	0000		WORD	0	05897
8067	3EAA	43	MNUCLD	FCC	'CLD'	05898
	3EAB	4C				
	3EAC	44				

MNUMONIC TABLE

15650

8068	3EAD	00		BYTE	0		05899
8069	3EAE	44	MNUDIFF	FCC	'DIFF'		05900
	3EAF	49					
	3EB0	46					
	3EB1	46					
8070	3EB2	0000		WORD	0		05901
8071	3EB4	3E	MNU2ORD	FCC	'>3RD'		05902
	3EB5	4F					
	3EB6	52					
	3EB7	44					
8072	3EB8	0000		WORD	0		05903
8073	3EBA	43	MNUCERON	FCC	'CERON'		05904
	3EBB	45					
	3EBC	52					
	3EBD	4F					
	3EBE	4E					
8074	3EBF	00		BYTE	0		05905
8075	3EC0	43	MNUCEROFF	FCC	'CEROFF'		05906
	3EC1	45					
	3EC2	52					
	3EC3	4F					
	3EC4	46					
	3EC5	46					
8076	3EC6	0000		WORD	1		05907
8077	3EC8	45	MNUEXRON	FCC	'EXRON'		05908
	3EC9	58					
	3ECA	52					
	3ECB	4F					
	3ECC	4E					
8078	3ECD	00		BYTE	0		05909
8079	3ECE	45	MNUEXROFF	FCC	'EXROFF'		05910
	3ECF	58					
	3ED0	52					
	3ED1	4F					
	3ED2	46					
	3ED3	46					
8080	3ED4	0000		WORD	0		05911
8081	3ED6	52	MNUREMON	FCC	'REMON'		05912
	3ED7	45					
	3ED8	40					
	3ED9	4F					
	3EDA	4E					
8082	3EDB	00		BYTE	0		05913
8083	3EDC	52	MNUREMOFF	FCC	'REMOFF'		05914
	3EDD	45					
	3EDE	40					
	3EDF	4F					
	3EE0	46					
	3EE1	46					
8084	3EE2	0000		WORD	1		05915
8085	3EE4	4F	MNUOPCON	FCC	'OPCON'		05916
	3EE5	50					
	3EE6	43					
	3EE7	4F					
	3EE8	4E					
8086	3EE9	00		BYTE	0		05917
8087	3EEA	4F	MNUOPCOFF	FCC	'OPCOFF'		05918
	3EE3	50					
	3EEC	43					

MNUMONIC TABLE

35650

	3EED	4F				
	3EEE	46				
8088	3EEF	46				
8088	3EF0	0000				05919
8089	3EF2	49	MNUIO3ON	WORD	0	05920
	3EF3	4F				
	3EF4	43				
	3EF5	4F				
8090	3EF6	4E				
8090	3EF7	00				05921
8091	3EF8	49	MNUIO3OFF	BYTE	0	05922
	3EF9	4F				
	3EFA	43				
	3EFB	4F				
	3EFC	46				
	3EFD	46				
8092	3EFE	0000				05923
8093	3F00	52	MNURQ3ON	WORD	0	05924
	3F01	51				
	3F02	53				
	3F03	4F				
	3F04	4E				
8094	3F05	00				05925
8095	3F06	52	MNURQ3OFF	BYTE	0	05926
	3F07	51				
	3F08	53				
	3F09	4F				
	3F0A	46				
	3F0B	46				
8096	3F0C	0000				05927
8097	3F0E	53	MNUSWL	WORD	0	05928
	3F0F	57				
	3F10	4C				
8098	3F11	00				05929
8099	3F12	43	MNUCLP	BYTE	0	05930
	3F13	4C				
	3F14	58				
8100	3F15	00				05931
8101	3F16	52	MNURTN	BYTE	0	05932
	3F17	54				
	3F18	4E				
8102	3F19	00				05933
8103	3F1A	53	MNUSTART	BYTE	0	05934
	3F1B	54				
	3F1C	41				
	3F1D	52				
	3F1E	54				
8104	3F1F	00				05935
8105	3F20	4C	MNULNN	BYTE	0	05936
	3F21	4E				
	3F22	4E				
8106	3F23	00				05937
8107	3F24	49	MNUIFXEQY	BYTE	0	05938
	3F25	46				
	3F26	58				
	3F27	3D				
	3F28	59				
8108	3F29	00				05939
8109	3F2A	43	MNJCLF	BYTE	0	05940

	3F2B	4C				
	3F2C	46				
0110	3F2D	00		BYTE	0	05941
0111	3F2E	53	MNUSENDX	FCC	'SENDX'	05942
	3F2F	45				
	3F30	4E				
	3F31	44				
	3F32	50				
0112	3F33	00		BYTE	0	05943
0113	3F34	43	MNUCRS1	FCC	'CRS1'	05944
	3F35	52				
	3F36	53				
	3F37	31				
0114	3F38	0000		WORD	0	05945

SYMBOL TABLE FOR ALPHA EXPONENTS

05946

8116	3F3A	0070	SYMTA3	WORD \$0070,\$465E,\$FFD9	05947
	3F3C	465E			
	3F3E	FFD9			
8117	3F40	006E		WORD \$006E,\$44B8,\$FFE3	05948
	3F42	44B8			
	3F44	FFE3			
8118	3F46	0013		WORD \$0013,\$431C,\$FFED	05949
	3F48	431C			
	3F4A	FFED			
8119	3F4C	006D		WORD \$006D,\$4189,\$FFF7	05950
	3F4E	4189			
	3F50	FFF7			
8120	3F52	004B		WORD 'K,\$7D00,\$000A	05951
	3F54	7D00			
	3F56	000A			
8121	3F58	004D		WORD 'M,\$7A12,\$0014	05952
	3F5A	7A12			
	3F5C	0014			
8122	3F5E	0047		WORD 'G,\$7736,\$001E	05953
	3F60	7736			
	3F62	001E			
8123	3F64	0054		WORD 'T,\$746A,\$0028	05954
	3F66	746A			
	3F68	0028			
8124	3F6A	FFFF	TABEND	WORD \$FFFF	05955

***** ROM HEADER FOR LOW ROMS *****

05956

8126			*				
8127			*	DEFINE ROM HEADER FOR LOW ORDER ROMS			
8128			*				
8129	3FE0			ORG	\$3FE0	ROM TAILER FOR \$0000	05957
8130							
8131	3FE0	0404		WORD	\$0404,\$0809	ROM PART NO. XXXX FROM 160-XXXX-00	05958
	3FE2	0809					
8132	3FE4	30		FCC	'00'	HIGH ORDER ADDRESS NYBBLE (I.E. 0XXX)	05959
	3FE5	30					
8133	3FE6	40		FCC	'ML'	M = MOST SIGNIFICANT, L = LEAST SIGNIFICANT	05960
	3FE7	4C					
8134	3FE8	30	RJHV	FCC	'0101'	ROM VERSION '01'	05961
	3FE9	31					
	3FEA	30					
	3FEB	31					
8135	3FEC	0000		WORD	210	SPARE	05962
8136							
8137	3FF0	0000		WORD	\$0000,\$0000	PROM PART NO. XXXX FROM 160-XXXX-00	05963
	3FF2	0000					
8138	3FF4	30		FCC	'00'	HIGH ORDER ADDRESS NYBBLE (I.E. 0XXX)	05964
	3FF5	30					
8139	3FF6	40		FCC	'ML'	M = MOST SIGNIFICANT, L = LEAST SIGNIFICANT	05965
	3FF7	4C					
8140	3FF8	30	PROMV	FCC	'0000'	PROM VERSION '00'	05966
	3FF9	30					
	3FFA	30					
	3FFB	30					
8141	3FFC	0000		WORD	210	CHECK SUMS	05967
8142							

05968

8144

4000

JR3 \$4000

05969

KEYPROG

05970

```

0146 *
0147 * KEYEXECUTE --- HANDLE 'EXECJTE' KEY
0148 *
0149 * LEVEL 1
0150 *
0151 * KEYGEKE EQU * 05971
0152 4000 C020 DAA0 MOV PROGRAM,R0 'EXEC' ENTERED FROM GPIB WHILE 05972
0153 4004 1301 JEQ **4 IN DIRECT MODE IS NOP 05973
0154 4006 0380 RTW 05974
0155 4008 C820 37EA D968 KYBJEX MOV GEXEKEY, LASTKEY 05975
0156 400E 0720 D94A SETO FATAL TURN OFF ALL ERROR FLAGS 05976
0157 4012 0720 D94C SETO WARNING 05977
0158 4016 0420 6CF4 BLW BUZZIT WARN USER OF MODE SWITCH 05978
0159 401A 0720 DAA0 SETO PROGRAM FLAG EXECUTION MODE 05979
0160 401E 0720 D972 SETO RDTFLAG UPDATE ENTIRE DISPLAY 05980
0161 4022 06A0 12BE BL CLRTXT INITIALIZE SCREEN 05981
0162 4026 C820 DABC D9C6 MOV PROREALT, REALTIME RESET DISPLAY STATUS WORDS TO WHAT THEY 05982
0163 402C C820 D96C D9CA MOV PRODSPRO, DSPRO 05983
0164 4032 C820 D96E D9C8 MOV PRODSPRT, DSPRLT 05984
0165 4038 C020 DAE2 MOV MAXPROG, R0 05985
0166 403C 9810 382D CB *R0, ENDKEYB IS LAST BYTE A 'E-O-P'? 05986
0167 4040 1302 JEQ **6 05987
0168 4042 04E0 D94C CLR WARNING WARN USER OF FULL PROGRAM IF NOT 05988
0169 4046 0720 DACC SETO OPROGLN MAKE SURE 'PL #' GETS DISPLAYED 05989
0170 404A 0380 RTW 05990
0171 *
0172 * KEYPROG --- HANDLE 'PROGRAM' KEY
0173 *
0174 * LEVEL 1
0175 *
0176 * KEYSPROG EQU * 05991
0177 404C C020 DAA0 MOV PROGRAM, R0 'PROG' ENTERED FROM GPIB WHILE IN 05992
0178 4050 1601 JNE **4 PROGRAM MODE IS NOP 05993
0179 4052 0380 RTW 05994
0180 * KEYSPROG EQU * 05995
0181 4054 C820 333A DAB6 MOV C1, SCRNLFLAG 05996
0182 405A 0720 D94A SETO FATAL 05997
0183 405E 0420 6CF4 BLW BUZZIT WARN USER OF MODE SWITCH 05998
0184 4062 C820 D9C6 DABC MOV REALTIME, PROREALT SAVE DISPLAY STATUS WORDS 05999
0185 4068 C820 D9CA D96C MOV DSPRO, PRODSPRO 06000
0186 406E C820 D9C8 D96E MOV DSPRLT, PRODSPRT 06001
0187 4074 4820 336E D9CA SZC MW8KHZ, DSPRO SET UP MODE WORDS FOR PROGRAM DISPLAY 06002
0188 407A 4820 3380 D9CA SZC MWDRLT, DSPRO 06003
0189 4080 4820 3380 D9C8 SZC MWDRLT, DSPRLT 06004
0190 4086 04E0 DAA0 CLR PROGRAM 06005
0191 408A C820 333A D9C6 MOV C1, REALTIME 06006
0192 4090 06A0 12BE BL CLRTXT 06007
0193 4094 0720 DAA2 SETO PROMODE 06008
0194 4098 0420 4A1E BLW LASTNXT SET LINE 99 UP 06009
0195 409C 0420 42FE BLW TXTSCRN 06010
0196 40A0 0380 RTW 06011

```



```

8198 *
8199 * FWDNXT - RETURNS LOCATION OF FIRST COMMAND IN NEXT LINE
8200 *
8201 * LEVEL 5 ROUTINE
8202 *
8203 * INPUT: R0 - LOCATION TO START THE SEARCH
8204 * OUTPUT: R0 - UPDATED TO CONTAIN THE LOCATION OF
8205 * THE START OF THE NEXT LINE
8206 *
8207 FWDNXT EQU * 06013
8208 40A2 9810 382D CB *R0,ENDKEYB 06014
8209 40A6 1308 JEQ FWDSTOP ALREADY AT END OF PROGRAM 06015
8210 40A8 9810 3805 CB *R0,LNNKEYB 06016
8211 40AC 1602 JNE **6 SKIP PAST 'LNN' 06017
8212 40AE 05C0 INCF R0 06018
8213 40B0 10F8 JMP FWDNXT 06019
8214 40B2 9830 37EF CB *R0+,NXTKEYB FOUND A 'NEXT' COMMAND? 06020
8215 40B6 16F5 JNE FWDNXT ..NO, KEEP LOOKING 06021
8216 40B8 0698 FWDSTOP BL *R11 ..YES, R0 ALREADY POINTING PAST THE 'NEXT' 06022
  
```

RVS NXT - RETURNS LOCATION OF FIRST COMMAND IN CURRENT LINE 06023

0218				*				
0219				*	RVS NXT - RETURNS LOCATION OF FIRST COMMAND IN CURRENT LINE			
0220				*				
0221				*	LEVEL 5 ROUTINE			
0222				*				
0223				*	INPJT: R0 - LOCATION TO START THE SEARCH			
0224				*	OUTPUT: R0 - UPDATED TO CONTAIN THE LOCATION OF			
0225				*	THE START OF THE CURRENT LINE			
0226				*				
0227			40BA	*	RVS NXT	EQU		06024
0228	40BA	0600			DEC	R0		06025
0229	40BC	0800	0AEO		C	R0,PROGMEM	DON'T BACK UP PAST START OF PROGRAM	06026
0230	40C0	1108			JLT	RVSTJP		06027
0231	40C2	9810	37EF		CB	*R0,NXTKEYB	IS IT A 'NEXT'?	06028
0232	40C6	16F9			JNE	RVS NXT	NO, LOOK FOR ONE	06029
0233	40C8	0600			DEC	R0		06030
0234	40CA	9810	3805		C3	*R0,LNNKEYB	MAKE SURE IT ISN'T PART OF 'LNN'	06031
0235	40CE	13F5			JEQ	RVS NXT		06032
0236	40D0	0580			INC	R0	GET COMMAND FOLLOWING THE 'NEXT' FOUND	06033
0237	40D2	0580			INC	R0	BY SEARCHING BACKWARD	06034
0238	40D4	0698			BL	*R11		06035

8240				*			
8241				*	CLRPTR ---	CLEARs RETURN POINTERS	
8242				*			
8243				*	LEVEL 5	ROUTINE	
8244				*			
8245				*	DESTROYS	R7 AND R8	
8246				*			
8247			4036		CLRPTR	EQU *	06037
8248	40D6	C1E0	33E8		MOV	GOSUBLEN,R7	06038
8249	40DA	0208	DA4E		LI	R8,GOSUBUF	06039
8250	40DE	04F8			CLR	*R8+	06040
8251	40E0	0607			DEC	R7	06041
8252	40E2	16FD			JNE	*-4	06042
8253	40E4	045B			B	*R11	06043

8255					*				
8256					*	PROGIN	---	HANDLE ALL PROGRAM ENTRY AND DIRECT TO EDIT ROUTINES	
8257					*			AS NECESSARY	
8258					*				
8259					*			KEYTRE LEVEL ROUTINE	
8260					*				
8261					*			NOTE: SEE FLOWCHARTS FOR ADDITIONAL INFORMATION	
8262					*				
8263			40E6			PROGIN	EQU *		06045
8264	40E6	C820	333A	D975			MOV	C1,INJSE SET BUSY FLAG	06046
8265	40EC	04E0	DAE6				CLR	RQSNUM TURN OFF SRQ'S	06047
8266	40F0	06A0	32AE				BL	RQS	06048
8267	40F4	05A0	DAB6				INC	SCRNFLAG	06049
8268	40F8	8820	D966	3806			C	KEY,NXTKEYG 'NEXT' FROM GPIB SAME AS 'NEXT' FROM KYBD	06050
8269	40FE	1603					JNE	*+8	06051
8270	4100	C820	37EE	D966			MOV	NXTKEY,KEY	06052
8271	4106	8820	DAAC	333A			C	CLPFLAG,C1 WAS CLP BEING CONSIDERED?	06053
8272	410C	1626					JNE	NOCLP	06054
8273	410E	8820	D966	37EE			C	KEY,NXTKEY YES, IS THIS A CONFIRMATION?	06055
8274	4114	1603					JNE	NOWIPE	06056
8275	4116	0420	472E				BLWP	KEYNXT2 YES, SO KILL PROGRAM	06057
8276	411A	1010					JMP	NOPURG	06058
8277			411C			NOWIPE	EQU *		06059
8278	411C	04E0	DAAC				CLR	CLPFLAG	06060
8279			4120			PJRGNKT	EQU *		06061
8280	4120	C060	DAAA				MOV	EDITPNT,R1	06062
8281	4124	0601					DEC	R1	06063
8282	4126	8801	DAE0				C	R1,PROGMEM	06064
8283	412A	1107					JLT	CHKFST	06065
8284	412C	9811	37EF				CB	*R1,NXTKEYB WAS THE LAST KEY ENTERED A 'NEXT'?	06066
8285	4130	1612					JNE	NOPURG	06067
8286	4132	0601					DEC	R1	06068
8287	4134	9831	3805				CB	*R1+,LNNKEYB MAKE SURE IT WASN'T 'LNN'	06069
8288	4138	130E					JEQ	NOPURG	06070
8289	413A	0581				CHKFST	INC	R1	06071
8290	413C	9811	37EF				CB	*R1,NXTKEYB	06072
8291	4140	160A					JNE	NOPURG	06073
8292	4142	0700					SETJ	R0 DELETE SECOND 'NEXT'	06074
8293	4144	0420	4566				BLWP	MOVSUB ..IF TWO 'NEXT' KEYS IN A ROW	06075
8294	4148	0620	DAAE				DEC	LINE99 DECREMENT LAST LINE NUMBER	06076
8295	414C	C820	DAAA	DAA6			MOV	EDITPNT,LINEPNT POINT TO FRONT OF CURRENT LINE	06077
8296	4152	04E0	DAA2				CLR	PROMODE	06078
8297	4156	0450	0208			NOPJRG	B	XCUTDON	06079
8298			415A			NOCLP	EQU *		06080
8299	415A	C060	D966				MOV	KEY,R1	06081
8300	415E	8801	3808				C	R1,STEPKEYG IS IT STEP KEY?	06082
8301	4162	1603					JNE	*+8	06083
8302	4164	04E0	D94A				CLR	FATAL STEP KEY NOT PROGRAMMABLE	06084
8303	4168	10F5					JMP	NOPURG	06085
8304	416A	8801	37E6				C	R1,PROGKEY IS IT THE KEYBOARD EXECUTE KEY?	06086
8305	416E	1604					JNE	CHKEXE	06087
8306	4170	C820	37EA	D966			MOV	GEXEKEY,KEY IF IN PROG MODE, CHANGE TO EXECUTE	06088
8307	4176	1029					JMP	SPEC.TST	06089
8308	4178	8801	37EA			C-KEXE	C	R1,GEXEKEY CHECK FOR EXECUTE FROM GPIB	06090
8309	417C	1326					JEQ	SPEC.TST IF SO JUST GO TO EXECUTE HANDLER	06091
8310	417E	8801	37FE				C	R1,GPRSKEY PROGRAM FROM GPIB IS NOP	06092
8311	4182	1323					JEQ	SPEC.TST	06093
8312	4184	C060	DAA2				MOV	PROMODE,R1 FIRST CHANGE SINCE EXECUTE MODE??	06094
8313	4188	1311					JEQ	NOTFST	06095

PRJGIN

05044

8314	418A	1510		JGT	NOTFST	..NO, DON'T UPDATE EDIT POINTER	06096	
8315	418C	3820	DAA8 DAA4	MOV	PROGLN,LINENJM	FIRST CHANGE,	06097	
8316	4192	0420	4584	BLWP	GETLINE	GET TO START OF CURRENT LINE	06098	
8317	4196	3820	DAA6 DAAA	MOV	LINEPNT,EDITPNT		06099	
8318	419C	C820	DAA4 DAA8	MOV	LINENUM,PROGLN	GET NEW LINE NUMBER	06100	
8319	41A2	06A0	40D5	BL	CLRPTX	CLEAR RETURN POINTERS	06101	
8320	41A6	C820	333A DAB0	MOV	C1,MJVFLAG	START EXECUTION AT EDIT POINTER	06102	
8321	41AC	C050	DAAA	NOTFST	MOV	EDITPNT,R1	06103	
8322	41B0	0601		DEC	R1		06104	
8323	41B2	9811	37F3	CB	*R1,SHIFTKEYB	WAS LAST KEY A 'SHIFT'?	06105	
8324	41B6	1609		JNE	SPEC.TST		06106	
8325	41B8	0601		DEC	R1		06107	
8326	41BA	9811	3805	CB	*R1,LNNKEYB	MAKE SURE IT WASN'T PART OF 'LNN'	06108	
8327	41BE	1305		JEQ	SPEC.TST		06109	
8328	41C0	0620	DAAA	DEC	EDITPNT		06110	
8329	41C4	0700		SETD	R0		06111	
8330	41C6	0420	4565	BLWP	MOVSUB	YES --- WIPE IT OUT	06112	
8331			41CA	SPEC.TST	EQU *		06113	
8332	41CA	C050	DAAA	MOV	EDITPNT,R1		06114	
8333	41CE	C2E0	DAB2	MOV	LNNFLAG,R11		06115	
8334	41D2	1511		JGT	NOSPEC	WE ARE INPUTTING A 'LNN'	06116	
8335	41D4	0200	37DA	LI	R0,SPEC.PROG	CHECK TO SEE IF KEY NEEDS SPECIAL HANDLING	06117	
8336			4108	SPEC.LOP	EQU *		06118	
8337	41D8	8810	3802	C	*R0,ENDPNT	END OF SPECIAL TABLE?	06119	
8338	41DC	130C		JEQ	NOSPEC		06120	
8339	41DE	8C20	D966	C	KEY,*R0+	DOES KEY MATCH TABLE?	06121	
8340	41E2	1302		JEQ	SPEC.FND		06122	
8341	41E4	05C0		INCT	R0		06123	
8342	41E6	10F8		JMP	SPEC.LOP		06124	
8343			41E8	SPEC.FND	EQU *		06125	
8344	41E8	C810	D948	MOV	*R0,KEYXFR2	IF KEY IS SPECIAL FUNCTION, PLACE ADDRESS IN SPEC	06126	
8345	41EC	0300	000F	LI	R5,\$F	ALLOW KEYBOARD INTERRUPTS TO OCCUR	06127	
8346	41F0	0420	D946	BLWP	KEYXFR	AND EXECUTE BRANCH TO IT	06128	
8347	41F4	1095		JMP	PURGNXT		06129	
8348			41F6	NOSPEC	EQU *		06130	
8349	41F6	C020	D966	MOV	KEY,R0	IF NOT A SPECIAL KEY, GET MNUMONIC FOR IT	06131	
8350	41FA	0A20		SLA	R0,2	KEY * 4	06132	
8351	41FC	0220	3836	AI	R0,KEYTAB	R0 = KEY HANDLER	06133	
8352	4200	C100		MOV	R0,R4		06134	
8353	4202	05C4		INCT	R4		06135	
8354	4204	C114		MOV	*R4,R4		06136	
8355	4206	D114		MOVB	*R4,R4	R4=FIRST MNUMONIC LETTER	06137	
8356	4208	0984		SRL	R4,8		06138	
8357	420A	0284	0020	CI	R4,\$20		06139	
8358	420E	1388		JEQ	PURGNXT	CHECK FOR TWO 'NEXT' KEYS IN A ROW	06140	
8359	4210	C2E0	DAB2	MOV	LNNFLAG,R11		06141	
8360	4214	1501		JGT	LNNDIGT	WE ARE INPUTTING A 'LNN'	06142	
8361	4216	102C		JMP	INPUTIT		06143	
8362			4218	LNNDIGT	EQU *		06144	
8363	4218	0205	37DE	LI	R5,CLLKEY	IS IT 'CLL' KEY?	06145	
8364	421C	8835	D965	C	*R5+,KEY		06146	
8365	4220	1607		JNE	LNNUMB		06147	
8366	4222	04E0	DAB2	CLR	LNNFLAG	IF SO 'LNN' NO LONGER IN PROGRESS	06148	
8367	4226	3815	D948	MOV	*R5,KEYXFR2		06149	
8368	422A	0420	D946	BLWP	KEYXFR	CLEAR THE LINE	06150	
8369	422E	1093		JMP	NOPURG		06151	
8370	4230	0601		LNNJMB	DEC	R1	THIS IS LNN NUMBER	06152
8371	4232	0091		MOVB	*R1,R2	YES, THIS IS IT'S NUMBER	06153	
8372	4234	30C2		MOV	R2,R3		06154	

8373	4236	0983			SRL R3,8		06155
8374	4239	0243	000F		ANDI R3,8F		06156
8375	423C	09C2			SRL R2,12		06157
8376	423E	0224	FFD8		AI R4,\$FFD8	IS THIS A NUMERIC KEY?	06158
8377	4242	1112			JLT INERR		06159
8378	4244	0284	0009		CI R4,9		06160
8379	4248	150F			JGT INERR		06161
8380	424A	0282	000A		CI R2,8A	IS FIRST NUMBER UNDEFINED?	06162
8381	424E	1605			JNE TRYA2		06163
8382	4250	0A44			SLA R4,4	THIS IS VALID DIGIT	06164
8383	4252	A103			A R3,R4		06165
8384	4254	0A84			SLA R4,8		06166
8385	4256	D444			MOVB R4,*R1	SAVE NUMBER CREATED SO FAR	06167
8386	4259	1050			JMP PROGDN		06168
8387			425A	TRYA2	EQU *		06169
8388	425A	0A42			SLA R2,4		06170
8389	425C	A102			A R2,R4		06171
8390	425E	0A84			SLA R4,8		06172
8391	4260	D444			MOVB R4,*R1	SAVE COMPLETED NUMBER	06173
8392	4262	04E0	DAB2		CLR LNNFLAG		06174
8393	4266	1049			JMP PROGDN		06175
8394			4268	INERR	EQU *		06176
8395	4268	04E0	D94A		CLR FATAL		06177
8396	426C	0450	4128		B PURGNXT		06178
8397			4270	INPJTIT	EQU *		06179
8398	4270	04C6			CLR R6	'LNN' KEY FLAG	06180
8399	4272	8820	D957	3805	C KEYB,LNNKEYB		06181
8400	4278	1601			JNE **4		06182
8401	427A	0706			SETD R6	INPJT 'LNN'	06183
8402	427C	C0E0	DAA2		MOV PROMODE,R3		06184
8403	4280	150D			JGT INPUTMOD		06185
8404			4282	MOJEEJIT	EQU *		06186
8405	4282	C160	DAE2		MOV MAXPROG,R5		06187
8406	4286	C186			MOV R6,R5	'LNN' REQUIRES AN EXTRA BYTE	06188
8407	4288	1301			JEQ **4		06189
8408	428A	0605			DEC R5		06190
8409	428C	0605			DEC R5		06191
8410	428E	9815	382D		CB *R5,ENDKEYB	IS THERE ROOM FOR TWO KEYS?	06192
8411	4292	16EA			JNE INERR		06193
8412	4294	8820	DAAE	33E4	C _INE99,ENJLINE	DOES LINE #999 EXIST?	06194
8413	429A	15E6			JGT INERR	IF SO, PROGRAM LINE MAX IS FILLED	06195
8414			429C	INPJT40D	EQU *		06196
8415	429C	C160	DAE2		MOV MAXPROG,R5		06197
8416	42A0	C186			MOV R6,R6	NEED AN EXTRA BYTE IF 'LNN'	06198
8417	42A2	1301			JEQ **4		06199
8418	42A4	0605			DEC R5		06200
8419	42A6	9815	382D		CB *R5,ENDKEYB	IS LAST KEY A 'E-O-P'?	06201
8420	42AA	16DE			JNE INERR	IF NOT, PROGRAM STORE IS FULL	06202
8421	42AC	0200	0001		LI R0,1		06203
8422	42B0	0420	4566		BLWP MOVSU3		06204
8423	42B4	C058	DAAA		MOV EDITPNT,R1		06205
8424	42B8	D460	D967		MOVB KEYB,*R1		06206
8425	42BC	C186			MOV R6,R6		06207
8426	42BE	130D			JEQ NOTLNN		06208
8427	42C0	D581			INC R1		06209
8428	42C2	0420	4566		BLWP MOVSUB	MAKE ROOM FOR LABEL NUMBER	06210
8429	42C6	0203	AA8D		LI R3,\$AA8D	NUMBER IS NOW UNDEFINED	06211
8430	42CA	D443			MOV3 R3,*R1		06212
8431	42CC	C820	333A	DAB2	MOV C1,LNNFLAG	'LNN' NOW IN PROGRESS	06213

PROGIN

05044

0432	42D2	0720	DA18		SETD	ROUPJT	MAKE SJRE READOUT IS UPDATED	06214
0433	42D6	05A0	DAAA		INC	EDITPNT	GET PAST 'LNN' KEYCODE AND LABEL #	06215
0434	42DA	05A0	DAAA	NJT.NN	INC	EDITPNT		06216
0435	42DE	00E0	DAA2		MOV	PROMODE,R3		06217
0436	42E2	1509			JGT	INPUTDON		06218
0437	42E4	0200	0001		LI	R0,1		06219
0438	42E8	0420	4566		BLW	MOVSUB		06220
0439	42EC	0581			INC	R1		06221
0440	42EE	0460	37EF		MOV3	NXTKEYB,*R1	INSERT A 'NEXT'	06222
0441	42F2	0420	4A1E		BLW	LASTNXT		06223
0442			42F6	INPUTDON	EQU	*		06224
0443	42F6	05E0	DAA2		INGT	PROMODE	NOW IN ENTRY MODE	06225
0444	42FA	0460	0208	PROGDON	B	XGUTDON		06226

```

0446 *
0447 *      TXTSCRN --- CREATE TEXT FOR PROGRAM DISPLAY
0448 *
0449 *      LEVEL 3
0450 *
0451 *      NOTE:
0452 *          THIS ROUTINE SETS UP THE 16 LINES OF PROGRAM TEXT THAT IS
0453 *          DISPLAYED WHILE IN PROGRAM EDIT OR ENTRY MODE.  THE "CURRENT"
0454 *          LINE" (THE LINE CONTAINING THE COMMAND POINTED TO BY THE
0455 *          EDITPINTER) IS THE 8TH LINE ON THE SCREEN AND IS DENOTED BY A
0456 *          CARET.  7 LINES PRECEDING AND 8 LINES FOLLOWING THE "CURRENT
0457 *          LINE" ARE ALSO DISPLAYED, IF THEY EXIST.
0458 *          40 CHARACTERS ARE ALLOWED PER DISPLAY LINE.  WHEN THIS
0459 *          CHARACTER COUNT IS EXCEEDED WHILE IN ENTRY MODE, A "NEXT"
0460 *          COMMAND IS AUTOMATICALLY INSERTED INTO PROGRAM STORE AND A
0461 *          NEW EDIT DISPLAY LINE IS STARTED.
0462 *
0463 42FE      DBA0      TXTSCRN WORD  WPLVL3      06228
0464 4300      4302      WORD  *+2      06229
0465 4302      8820      DAAC      333A      C CLPFLAG,C1      IS CLP BEING CONSIDERED? 06230
0466 4308      1601      JNE *+4      06231
0467 430A      0380      RTWP      YES --- DO NOT DISPLAY TEXT 06232
0468 430C      C020      DAB6      MOV SCRNFALG,R0 06233
0469 4310      1501      JGT *+4      06234
0470 4312      0380      RTWP      06235
0471 4314      04E0      DAB6      CLR SCRNFALG 06236
0472      4318      BEGNSCRN EQU *      06237
0473 4318      C260      0012      MOV 18(R13),SOFT GET STACK POINTER 06238
0474 431C      0200      0001      LI R0,1      06239
0475 4320      0649      DECT SOFT      INITIALIZE STACK FOR TEXT FORMATTER 06240
0476 4322      C640      MOV R0,*SOFT 06241
0477 4324      0649      DECT SOFT 06242
0478 4326      C640      MOV R0,*SOFT 06243
0479 4328      C020      DAA8      MOV PROGLN,R0 06244
0480 432C      1502      JGT *+6      06245
0481 432E      1301      JEQ *+4      06246
0482 4330      04C0      CLR R0      IF PROGLN IS NEGATIVE, SET IT TO ZERO 06247
0483 4332      04C1      CLR R1      R1 = # BLANK LINES 06248
0484 4334      0220      FFF9      AI R0,-7      R0 HAS PROGRAM LINE NUMBER BEING DISPLAYED 06249
0485 4338      1503      JGT *+8      06250
0486 433A      C040      MOV R0,R1      CURRENT LINE < 8 06251
0487 433C      0741      ABS R1      06252
0488 433E      04C0      CLR R0      START AT LINE #0 06253
0489 4340      C081      MOV R1,R2      R2 = SCREEN LINE # PRESENTLY ON 06254
0490      4342      B_NK EQU *      06255
0491 4342      1305      JEQ BLNKDON 06256
0492 4344      06A0      12FA      BL NULLIN 06257
0493 4348      06A0      12DE      BL ADVLIN      ADVANCE BLANK LINES 06258
0494 434C      0601      DEC R1 06259
0495 434E      18F9      JMP BLNK 06260
0496      4350      BLNKDON EQU *      06261
0497 4350      0207      0008      LI R7,8      06262
0498 4354      C160      DAAA      MOV EDITPNT,R5      START SEARCH AT EDITPNT 06263
0499 4358      9815      37EF      CB *R5,NXTKEYB 06264
0500 435C      1601      JNE *+4      06265
0501 435E      0587      INC R7 06266
0502      4360      NXTHUNT EQU *      06267
0503 4360      9815      37EF      CB *R5,NXTKEYB      FIND A 'NXT'? 06268
0504 4364      1305      JEQ TXTNXT      YES?? 06269
    
```


8505	4366	0505		DEC R5	HUNT BACKWARDS	06270
8506	4369	8805	DAE0	C R5,PROGHEM	DON'T GO BEYOND START OF PROGRAM...	06271
8507	436C	1108		JLT STRTXT		06272
8508	436E	10F8		JMP NXTHUNT		06273
8509			4370	TXTXT EQU *		06274
8510	4370	0605		DEC R5	BACK ONE PAST POTENTIAL 'NEXT'	06275
8511	4372	9815	3805	CB *R5,LNNKEYB	IS THAT ONE A 'LNN'?	06276
8512	4376	13F4		JEQ NXTHUNT	IF SO, DON'T COUNT IT	06277
8513	4378	0607		DEC R7	IF NOT, COUNT IT	06278
8514	437A	16F2		JNE NXTHUNT		06279
8515	437C	0585		INC R5		06280
8516			437E	STRTXT EQU *		06281
8517	437E	0585		INC R5		06282
8518			4380	NEXTLIN EQU *		06283
8519	4380	04E0	D98A	CLR PROGRS		06284
8520	4384	8802	3358	C R2,C16		06285
8521	4388	1101		JLT *+4		06286
8522	438A	0380		RTWP	SCREEN DONE	06287
8523	438C	0204	DA62	LI R4,TEXTBJFR	R4=BUFFER POINTER	06288
8524	4390	04C6		CLR R6		06289
8525	4392	C1C0		MOV R0,R7		06290
8526	4394	30A0	3374	DIV C100,R6		06291
8527	4398	C206		MOV R6,R8		06292
8528	439A	04C6		CLR R6		06293
8529	439C	30A0	334C	DIV C10,R6	GET DIGITS FOR LINE NUMBER	06294
8530	43A0	0226	0030	AI R6,\$30		06295
8531	43A4	J227	0030	AI R7,\$30		06296
8532	43A8	0228	0030	AI R8,\$30		06297
8533	43AC	0A86		SLA R6,8		06298
8534	43AE	0A87		SLA R7,8		06299
8535	43B0	0A88		SLA R8,8		06300
8536	43B2	D520	3433	MOV3 SPACBYT,*R4	START W/ SPACE	06301
8537	43B6	02E0	DAE0	MOV PROGHEM,R11		06302
8538	43BA	981B	3820	CB *R11,ENDKEYB	IS PROGRAM NULL?	06303
8539	43BE	1320		JEQ TXTQUIT		06304
8540			43C0	NOCRT EQU *		06305
8541	43C0	8800	DAA8	C R0,PROGLN	IS THIS 'CURRENT LINE'?	06306
8542	43C4	160F		JNE EDITMODE		06307
8543	43C6	D520	34DD	MOV3 CARROT,*R4	YES, SO PUT CARROT INSTEAD	06308
8544	43CA	8820	DAA2	C PROMODE,C0		06309
8545	43D0	1309		JEQ EDITMODE		06310
8546	43D2	1108		JLT EDITMODE		06311
8547			43D4	NONUMLIN EQU *		06312
8548	43D4	0584		INC R4	IN ENTRY MODE ==> NO LINE NUMBER	06313
8549	43D6	0020	3433	MOV3 SPACBYT,*R4+		06314
8550	43DA	0020	3433	MOV3 SPACBYT,*R4+		06315
8551	43DE	0020	3433	MOV3 SPACBYT,*R4+		06316
8552	43E2	1007		JMP WRITELIN		06317
8553			43E4	EDITMODE EQU *		06318
8554	43E4	8800	DAAE	C R0,LINE99	IF LAST LINE, DON'T PUT UP LINE NUMBER	06319
8555	43E8	13F5		JEQ NONUMLIN		06320
8556	43EA	0584		INC R4		06321
8557	43EC	0008		MOV3 R8,*R4+		06322
8558	43EE	0006		MOV3 R6,*R4+		06323
8559	43F0	0007		MOV3 R7,*R4+		06324
8560			43F2	WRITELIN EQU *		06325
8561	43F2	0020	3433	MOV3 SPACBYT,*R4+		06326
8562	43F6	C0E0	3342	MOV C5,R3	R3=LINE LENGTH	06327
8563			43FA	NEXTCAR EQU *		06328

8564	43FA	0406		CLR R6		06329
8565	43FC	01B5		MOVB *R5+,R6	R6 = KEY TO DISPLAY	06330
8566	43FE	9806	382D	CB R6,ENDKEYB		06331
8567	4402	1623		JNE CHKNXT		06332
8568	4404	04E0	09BA	CLR PROGRS	THIS IS E-J-P	06333
8569	4408	0204	DA62	LI R4,TXTBUFR		06334
8570	440C	9814	34DJ	CB *R4,CARROT	IF THIS CURRENT LINE?	06335
8571	4410	1303		JEQ *+8	IF SO, BE SURE TO DISPLAY IT	06336
8572	4412	8803	3344	C R3,C6		06337
8573	4416	110F		JLT TXTEND	IF LAST LINE IS ONLY LINE NUMBER, SKIP IT	06338
8574	4418	1004		JMP *+10		06339
8575			441A	TXTQUIT EQU *		06340
8576	441A	0520	34DD	MOVB CARROT,*R4		06341
8577	441E	00E0	333A	MOV C1,R3		06342
8578	4422	0649		DECT SOFT		06343
8579	4424	C644		MOV R4,*SOFT		06344
8580	4426	0649		DECT SOFT		06345
8581	4428	C643		MOV R3,*SOFT	TEXT LENGTH	06346
8582	442A	06A0	136C	BL TEXT		06347
8583	442E	06A0	12FA	BL NULLIN		06348
8584	4432	06A0	12DE	BL ADVLIN		06349
8585			4436	TXTEND EQU *		06350
8586	4436	0282	000F	CI R2,15		06351
8587	443A	1506		JGT ENDSCRN		06352
8588	443C	06A0	12FA	BL NULLIN		06353
8589	4440	06A0	12DE	BL ADVLIN		06354
8590	4444	0582		INC R2		06355
8591	4446	10F7		JMP TXTEND		06356
8592			4448	ENDSCRN EQU *		06357
8593	4448	0380		RTWP		06358
8594			444A	CHKNXT EQU *	THIS IS NOT E-0-P	06359
8595	444A	9806	37EF	CB R6,NXTKEY3		06360
8596	444E	1612		JNE MORTXT		06361
8597	4450	0204	DA62	LI R4,TXTBJFR		06362
8598	4454	0649		DECT SOFT		06363
8599	4456	C644		MOV R4,*SOFT		06364
8600	4458	0649		DECT SOFT		06365
8601	445A	C643		MOV R3,*SOFT		06366
8602	445C	06A0	136C	BL TEXT		06367
8603	4460	06A0	12FA	BL NULLIN		06368
8604	4464	06A0	12DE	BL ADVLIN		06369
8605	4468	0580		INC R0		06370
8606	446A	0582		INC R2	COUNT LINE IN SCREEN	06371
8607	446C	04E0	09BA	CLR PROGRS		06372
8608	4470	0460	4380	B NEXTLINE		06373
8609			4474	MORTXT EQU *		06374
8610	4474	9806	3805	CB R6,LNNKEY3		06375
8611	4478	1646		JNE NOLABL		06376
8612	447A	C320	09BA	MOV PROGRS,R12		06377
8613	447E	1305		JEQ *+12		06378
8614	4480	0D20	3433	MOVB SPACBYT,*R4+	SPACE IT OUT IF FOLLOWING NUMERIC INPJT	06379
8615	4484	0583		INC R3		06380
8616	4486	04E0	09BA	CLR PROGRS	NUMERIC ENTRY NO LONGER IN PROGRESS	06381
8617	448A	0207	4C00	LI R7,\$4C00	"L"	06382
8618	448E	0D07		MOVB R7,*R4+		06383
8619	4490	01B5		MOVB *R5+,R6		06384
8620	4492	0986		SRL R6,8		06385
8621	4494	C1C6		MOV R6,R7		06386
8622	4496	0946		SRL R6,4		06387

8623	4498	0247	000F		ANDI R7,\$F	06388
8624	449C	0226	0030		AI R6,\$30	06389
8625	44A0	0227	0030		AI R7,\$30	06390
8626	44A4	0208	004E		LI R8,'N' 'N'	06391
8627	44A8	0286	003A		CI R6,\$3A	06392
8628	44AC	1101			JLT **4	06393
8629	44AE	0188			MOV R8,R6	06394
8630	44B0	0287	003A		CI R7,\$3A	06395
8631	44B4	1101			JLT **4	06396
8632	44B6	01C8			MOV R8,R7	06397
8633	44B8	0A86			SLA R5,8	06398
8634	44BA	0A87			SLA R7,8	06399
8635	44BC	0D06			MOVB R6,*R4+	06400
8636	44BE	0D07			MOVB R7,*R4+	06401
8637	44C0	0D20	3433		MOVB SPACBYT,*R4+	06402
8638	44C4	0223	0004		AI R3,4	06403
8639			44C8	NEXTC2	EQU *	06404
8640	44C8	8803	3366		C R3,C40	06405
8641	44CC	1501			JGT **4	06406
8642	44CE	1095			JMP NEXTCHAR	06407
8643	44D0	0605			DEC R5 POINT AHEAD OF OFFENDING KEY	06408
8644	44D2	0020	DAB2		MOV LNNFLAG,R0	06409
8645	44D6	1301			JEQ **4	06410
8646	44D8	0605			DEC R5 BACK UP ONE MORE IF LNN JUST WENT UP	06411
8647	44DA	0200	0001		LI R0,1	06412
8648	44DE	0120	DAAA		MOV EDITPNT,R4 SAVE EDITPNT	06413
8649	44E2	0805	DAAA		MOV R5,EDITPNT MAKE ROOM FOR NXT IN FRONT OF OFFENDING CHARACTER	06414
8650	44E6	0420	4566		BLWP MOVSU3	06415
8651	44EA	0560	37EF		MOVB NXTKEYB,*R5 FORCE ANOTHER NEXT IN LINE	06416
8652	44EE	0804	DAAA		MOV R4,EDITPNT	06417
8653	44F2	05A0	DAAA		INC EDITPNT	06418
8654	44F6	05A0	DAAA		INC PROGLN	06419
8655	44FA	0420	4A1E		BLWP LASTNXT	06420
8656	44FE	0420	6CE8		BLWP LITERR	06421
8657	4502	0460	4318		B 3EGNSCRN	06422
8658			4506	NOLAB	EQU *	06423
8659	4506	0966			SRL R6,6 KEY *4	06424
8660	4508	0226	3836		AI R6,KEYTAB POINT TO KEY HANDLER	06425
8661	450C	01F6			MOV *R6+,R7 KEY HANDLER	06426
8662	450E	0216			MOV *R6,R8	06427
8663	4510	0287	649E		CI R7,KEYB IS THIS NUMERIC KEY?	06428
8664	4514	1617			JNE NOTXTNUM	06429
8665	4516	0D38			MOVB *R8+,*R4+ YES --- 1 CHAR ONLY	06430
8666	4518	0583			INC R3	06431
8667	451A	0605			DEC R5	06432
8668	451C	01B5			MOVB *R5+,R6 GET KEY AGAIN	06433
8669	451E	0986			SRL R6,8	06434
8670	4520	8806	382A		C R5,CHSKEY CHS AND EEX NEED 3 CHARS	06435
8671	4524	1305			JEQ TXT3JUT	06436
8672	4526	05A0	D98A		INC PROGRS NUMERIC ENTRY FLAG	06437
8673	452A	8806	3826		C R6,EEXKEY	06438
8674	452E	16CC			JNE NEXTC2	06439
8675			4530	TXT3JUT	EQU *	06440
8676	4530	0D38			MOVB *R8+,*R4+	06441
8677	4532	0D38			MOVB *R8+,*R4+	06442
8678	4534	05C3			INCR R3	06443
8679	4536	01E0	D93A		MOV PROGRS,R7	06444
8680	453A	15C6			JGT NEXTC2 IS CHS EMBEDDED IN NUMERIC STRING??	06445
8681	453C	0D20	3433		MOVB SPACBYT,*R4+ NO, SPACE AFTER CHS	06446

8682	4540	0583		INC R3	06447
8683	4542	10C2		JMP NEXTC2	06448
8684			4544	NOTXTNUM EQU *	06449
8685	4544	C320	D98A	MOV PROGRS,R12 WAS LAST KEY A NUMBER?	06450
8686	4548	1305		JEQ **12	06451
8687	454A	DD20	3433	MOV3 SPACBYT,*R4+ YES, SPACE IT	06452
8688	454E	0583		INC R3	06453
8689	4550	04E0	D98A	CLR PROGRS	06454
8690	4554	0538		MOV3 *R8+,*R4 XFR MNUMONIC	06455
8691	4556	1303		JEQ ENDMNU	06456
8692	4558	0584		INC R4	06457
8693	455A	0583		INC R3	06458
8694	455C	10FB		JMP *-8	06459
8695			455E	ENDMNU EQU *	06460
8696	455E	DD20	3433	MOV3 SPACBYT,*R4+ OVERWRITE ZERO W/ SPACE	06461
8697	4562	0583		INC R3	06462
8698	4564	10B1		JMP NEXTC2	06463

MOVSUB

05464

```

8700 *
8701 *      MOVSUB --- MOVE PROGRAM UP OR DOWN # BYTES IN R0
8702 *      ALLWS INSERTION AND DELETION OF PROGRAM
8703 *      COMMANDS INTO PROGRAM STORE
8704 *
8705 *
8706 *      LEVEL 4
8707 *
8708 4566      DBC0      MOVSUB  WORD  WPLV_4      06465
8709 4568      456A      456A      WORD  *+2      06466
8710 456A      C1E0      DAE2      MOV  MAXPROG,R7  06467
8711 456E      05C7      INCT  R7      06468
8712 4570      C01D      MOV  *R13,R0    HOW FAR TO MOVE  06469
8713 4572      110D      JLT  MOVBACK   06470
8714 4574      1501      JGT  *+4      06471
8715 4576      0380      RTWP      NO MOVE???  06472
8716 4578      C087      MOV  R7,R2    06473
8717 457A      C0C2      MOV  R2,R3    06474
8718 457C      6080      S   R0,R2    06475
8719 457E      D402      MOVB *R2,*R3  06476
8720 4580      0603      DEC  R3      06477
8721 4582      0602      DEC  R2      06478
8722 4584      8802      DAAA      C   R2,EDITPNT  MOVE ONLY DOWN TO EDIT POINTER  06479
8723 4588      15FA      JGT  *-10    06480
8724 458A      13F9      JEQ  *-12    06481
8725 458C      0380      RTWP      06482
8726 458E      458E      MOVBACK EQU  *      06483
8727 458E      0587      INC  R7      06484
8728 4590      C0A0      DAAA      MOV  EDITPNT,R2  06485
8729 4594      C0C2      MOV  R2,R3    06486
8730 4596      6080      S   R0,R2    06487
8731 4598      DCF2      MOVB *R2+,*R3+  06488
8732 459A      81C2      C   R2,R7    MOVE ONLY TO END OF PROGRAM BOUNDARY  06489
8733 459C      11FD      JLT  *-4      06490
8734 459E      0603      DEC  R3      06491
8735 45A0      DCE0      382D      MOVB ENDKEYB,*R3+  CAP OFF PROGRAM WITH 'E-O-P'  06492
8736 45A4      81C3      C   R3,R7    06493
8737 45A6      11FC      JLT  *-6      06494
8738 45A8      0380      RTWP      06495

```

8740				*					
8741				*	GETLINE --- GET START OF DESIRED LINE				
8742				*	GETPNT --- GET LINE NUMBER FOR EDITPNT				
8743				*					
8744				*	FOR GETLINE ---				
8745				*	INPJT --- LINE NUMBER DESIRED IN 'LINENUM'				
8746				*	OUTPUT--- ADDRESS OF START OF THAT LINE IN 'LINEPNT'				
8747				*					
8748				*	FOR GETPNT ---				
8749				*	INPUT --- PROGRAM ADDRESS DESIRED TO FIND IN 'EDITPNT'				
8750				*	OUTPUT--- CORRESPONDING LINE NUMBER IN 'LINENUM'				
8751				*	NEXT EXECUTABLE ADDRESS IN 'LINEPNT'				
8752				*					
8753				*	IF LINE NUMBER NOT FOUND (BEYOND END), THEN LINENUM SET TO -1				
8754				*	AND LINEPNT TO 'E-O-P'				
8755				*	IF LINE NUMBER REQUESTED IS \$FFFF THEN LINENUM IS SET TO LAST STORE				
8756				*					
8757				*	LEVEL 4				
8758				*					
8759	45AA		DBC0	GETPNT	WORD	WPLVL4			06497
8760	45AC		45AE		WORD	*+2			06498
8761	45AE	3120	333A		MOV	C1,R4			06499
8762	45B2	1003			JMP	*+8			06500
8763	45B4		DBC0	GETLINE	WORD	WPLVL4			06501
8764	45B6		45B8		WORD	*+2			06502
8765	45B8	04C4			CLR	R4			06503
8766	45BA	C020	DAE0		MOV	PRGEM,R0	R0=ADDRESS POINTER		06504
8767	45BE	C050	DAA4		MOV	LINENUM,R1	R1=# TO LOOK FOR		06505
8768	45C2	04C2			CLR	R2	R2=CURRENT LINE		06506
8769			45C4	LINEHUNT	EQU	*			06507
8770	45C4	C104			MOV	R4,R4			06508
8771	45C6	1603			JNE	PNTHUNT			06509
8772	45C8	8042			C	R2,R1	FOUND THE LINE LOOKING FOR?		06510
8773	45CA	1316			JEQ	HITPNT	..YES, STOP SEARCHING		06511
8774	45CC	1004			JMP	*+10			06512
8775			45CE	PNTHUNT	EQU	*			06513
8776	45CE	8020	DAAA		C	EDITPNT,R0	LOOKING FOR ADDRESS, NOT LINE NUMBER		06514
8777	45D2	1312			JEQ	HITPNT			06515
8778	45D4	1111			JLT	HITPNT			06516
8779	45D6	9810	3823		CB	*R0,ENDKEY8	AT THE END OF THE PROGRAM?		06517
8780	45DA	130A			JEQ	NOLINE	IF SO, LINE NOT FOUND		06518
8781	45DC	9810	3805		CB	*R0,LNNKEY3	IGNORE NEXT COMMAND IF 'LNN'		06519
8782	45E0	1602			JNE	*+6			06520
8783	45E2	05C0			INCT	R0			06521
8784	45E4	10EF			JMP	LINEHUNT			06522
8785	45E6	9830	37EF		CB	*R0+,NXTKEY8	FOUND 'NEXT' KEY?		06523
8786	45EA	16EC			JNE	LINEHUNT	..NO, KEEP SEARCHING		06524
8787	45EC	0582			INC	R2	..YES, INCREMENT THE LINE COUNT		06525
8788	45EE	10EA			JMP	LINEHUNT			06526
8789			45F0	NOLINE	EQU	*	DID NOT FIND THE REQUESTED LINE NUMBER		06527
8790	45F0	0281	FFFF		CI	R1,\$FFFF	HAS REQUESTED LINE # = FFFF?		06528
8791	45F4	1301			JEQ	*+4	..YES, LEAVE R2 EQUAL TO LAST STORED LINE #		06529
8792	45F6	0702			SETD	R2	..NO, SET R2 TO FFFF TO SHOW LINE NOT FOUND		06530
8793			45F8	HITPNT	EQU	*			06531
8794	45F8	C800	DAA6		MOV	R0,LINEPNT	RETURN ADDRESS OF NEXT EXECUTABLE ADDRESS		06532
8795	45FC	C802	DAA4		MOV	R2,LINENUM	RETURN NEW LINE NUMBER		06533
8796	4600	0380			RTWP				06534

8798				*					
8799				*	KEYRTN --- HANDLE RTN KEY				
8800				*					
8801				*	LEVEL 1				
8802				*					
8803				*	ONLY REACHED IN IDLE OR EXECUTE STATES				
8804				*					
8805				*					
8806			4502		KEYRTN EQU *				06536
8807	4602	0200	DA4E		LI R0,GOSUBUF	GET POINTER BUFFER			06537
8808	4606	C060	33E8		MOV GOSUBLEN,R1	GET BUFFER LENGTH			06538
8809	460A	0601			DEC R1				06539
8810	460C	0A11			SLA R1,1				06540
8811	460E	A001			A R1,R0	SEARCH THROUGH BUFFER BACKWARDS			06541
8812	4610	C060	33E8		MOV GOSUBLEN,R1	LOOK FOR NON-ZERO BUFFER LOCATION			06542
8813	4614	C090			MOV *R0,R2	IF NON-ZERO, GOT A RETURN POINTER			06543
8814	4616	1605			JNE HITRTN				06544
8815	4618	0640			DECT R0				06545
8816	461A	0601			DEC R1				06546
8817	461C	16FB			JNE *-B				06547
8818	461E	04E8	094A		CLR FATAL	NO RETURN POINTER FOUND			06548
8819	4622	0380			RTWP				06549
8820			4624		HITRTN EQU *				06550
8821	4624	C810	DAAA		MOV *R0,EDITPNT				06551
8822	4628	0400			CLR *R0	WIPE POINTER			06552
8823	462A	0420	45AA		BLWP GETPNT	GET LINE NUMBER FOR THIS ADDRESS			06553
8824	462E	C820	DAA4	DAA8	MOV LINENUM,PROGLN				06554
8825	4634	C820	DAA5	DAAA	MOV LINEPNT,EDITPNT	NEXT EXECUTABLE ADDRESS			06555
8826	463A	C820	333A	DA30	MOV C1,MOVFLAG	DON'T INCREMENT EDITPNT			06556
8827	4640	05A0	DAB8		INC STPROGF	EXECUTE NEXT LINE IF GSB LAST THING IN LINE			06557
8828	4644	0380			RTWP				06558

8830				*													
8831				*	KEYGSB	---	HANDLE	GSB	KEY								
8832				*	KEYGOTO	---	HANDLE	GOTO	KEY								
8833				*													
8834				*	LEVEL	1											
8835				*													
8836				*	ONLY	REACHED	IN	IDLE	OR	EXECUTE	STATES						
8837				*													
8838				*													
8839				*	NOTE:	ONLY	10	LEVELS	OF	SUBROUTINE	NESTING	ALLOWED					
8840				*													
8841			4646		KEYGSB	EQU	*				06560						
8842	4646	0706			SETD	R6					06561						
8843	4648	1001			JMP	*+4					06562						
8844			464A		KEYGOTO	EQU	*				06563						
8845	464A	04C6			CLR	R6					06564						
8846	464C	06A0	690E		BL	POPREG		GET	LINE	NUMBER	TO	BRANCH	TO	06565			
8847	4650	C142			MOV	R2,R5					06566						
8848	4652	C101			MOV	R1,R4					06567						
8849	4654	C0C0			MOV	R0,R3					06568						
8850	4656	1635			JNE	GOTOERR		MUST	BE	CONSTANT	06569						
8851	4658	06A0	75A0		BL	FP2INT					06570						
8852	465C	1132			JLT	GOTOERR		MUST	BE	>=0	06571						
8853	465E	8801	33E4		C	R1,ENDLINE					06572						
8854	4662	152F			JGT	GOTOERR		MUST	BE	<100	06573						
8855	4664	E820	3348	D972	SOC	CLINE15,RDTFLAG					06574						
8856	466A	8801	DAAE		C	R1,LINE99					06575						
8857	466E	1107			JLT	NOWARN					06576						
8858	4670	0720	DAC2		SETD	EOPFLG		IF	AT	END	OF	PROGRAM	DON'T	DISPLAY	PL	#	06577
8859	4674	C060	DAAE		MOV	LINE99,R1					06578						
8860	4678	04E0	D94C		CLR	WARNING					06579						
8861	467C	1012			JMP	GOTO<		SET	UP	EDIT	POINTER	TO	POINT	AT	E-O-P	06580	
8862			467E		NOWARN	EQU	*				06581						
8863	467E	C186			MOV	R6,R6					06582						
8864	4680	1310			JEQ	GOTOK					06583						
8865	4682	C020	33E8		MOV	GOSUBLEN,R0					06584						
8866	4686	0206	DA4E		LI	R6,GOSUBUF					06585						
8867	468A	C086			MOV	*R6+,R2					06586						
8868	468C	1303			JEQ	HITSPOT		LOOK	FOR	EMPTY	GSB	POINTER	SPOT	06587			
8869	468E	0600			DEC	R0					06588						
8870	4690	16FC			JNE	*-6					06589						
8871	4692	1017			JMP	GOTOERR		GSB	NESTING	TOO	DEEP	06590					
8872			4694		HITSPOT	EQU	*				06591						
8873	4694	0646			DECT	R6		POINT	BACK	TO	SPOT	FOUND	06592				
8874	4696	C5A0	DAAA		MOV	EDITPNT,*R6		SAVE	RETURN	POINTER	06593						
8875	469A	C020	DAA0		MOV	PROGRAM,R0		IF	ENTERED	FROM	IDLE	MODE,	06594				
8876	469E	1101			JLT	*+4		DON'T	INCREMENT	RETURN	POINTER	06595					
8877	46A0	0596			INC	*R6		POINT	PAST	GSB	06596						
8878			46A2		GOTOK	EQU	*				06597						
8879	46A2	C801	DAA4		MOV	R1,LINENUM					06598						
8880	46A6	0420	45B4		BLWP	GETLINE		GET	NEW	LOCATION	TO	BRANCH	TO	06599			
8881	46AA	C820	DAA5	DAAA	MOV	LINEPNT,EDITPNT					06600						
8882	46B0	C820	DAA4	DAA8	MOV	LINENUM,PROGLN		GET	NEW	LINE	NUMBER	06601					
8883	46B6	C820	333A	DA30	MOV	C1,MOVFLAG					06602						
8884	46BC	0720	DAB4		SETD	PROGSTEP		RETURN	TO	IDLE	IF	IN	STEP	MODE	06603		
8885	46C0	0380			RTW						06604						
8886			46C2		GOTJERR	EQU	*				06605						
8887	46C2	04E0	D94A		CLR	FATAL					06606						
8888	46C6	C085			MOV	R5,R2					06607						

8889	46C8	C044	MOV R4,R1	06608
8890	46CA	C003	MOV R3,R0	06609
8891	46CC	06A0 6962	BL PSHREG	06610
8892	46D0	0380	RTMP	06611

8894				*				
8895				*	KEYLBL --- HANDLE LBL KEY			
8896				*				
8897				*	LEVEL 1			
8898				*				
8899				*				
8900				*	ONLY REACHED IN IDLE OR EXECJTE STATES			
8901				*				
8902			46D2		KEYLBL EQU *			06613
8903	46D2	06A0	690E		BL POPREG GET X			06614
8904	46D6	C142			MOV R2,R5			06615
8905	46D8	C101			MOV R1,R4			06616
8906	46DA	C030			MOV R0,R3			06617
8907	46DC	16F2			JNE GOTOERR X MUST BE A CONSTANT			06618
8908	46DE	06A0	75A0		BL FP2INT			06619
8909	46E2	11EF			JLT GOTOERR MUST BE >= 0			06620
8910	46E4	8801	33E4		C R1,ENDLINE			06621
8911	46E8	15EC			JGT GOTOERR MUST BE < 100			06622
8912	46EA	E820	3348	D972	SOC CLINE16,ROTFLLAG			06623
8913	46F0	04C0			CLR R0			06624
8914	46F2	3C20	334C		DIV C10,R0			06625
8915	46F6	0A40			SLA R0,4			06626
8916	46F8	E040			SOC R0,R1 R1 NOW HAS 3CD LABEL TO SEARCH FOR			06627
8917	46FA	C020	DAE0		MOV PROGHEM,R0			06628
8918	46FE	CA81			SLA R1,8			06629
8919	4700	04C2			CLR R2			06630
8920			4702		SRC4LNN EQJ *			06631
8921	4702	9810	37EF		CB *R0,NXTKEYB			06632
8922	4706	1603			JNE *+8			06633
8923	4708	0582			INC R2 COUNT LINES			06634
8924	470A	0580			INC R0 GOTO NEXT WORD			06635
8925	470C	10FA			JMP SRCHLNN			06636
8926	470E	9830	3805		CB *R0+,LNNKEYB IS IT A LABEL?			06637
8927	4712	1602			JNE *+6			06638
8928	4714	9070			CB *R0+,R1 YES, IS IT THE RIGHT LABEL?			06639
8929	4716	1304			JEQ HITLNN			06640
8930	4718	9810	382D		CB *R0,ENDKEYB			06641
8931	471C	16F2			JNE SRCHLNN			06642
8932	471E	1031			JMP GOTOERR IF LABEL # NOT IN PROGRAM, ERROR			06643
8933			4720		HITLNN EQU *			06644
8934	4720	3042			MOV R2,R1 FOUND CORRECT LABEL IN LINE			06645
8935	4722	06A0	757E		BL INT2FP CONVERT LINE NUMBER TO FP			06646
8936	4726	04C0			CLR R0			06647
8937	4728	06A0	6962		BL PSHREG PUSH LINE # ONTO X STACK			06648
8938	472C	0380			RTWP			06649

KEYNEXT / KEYPREV / KEYSTEP

06650

```

8940 *
8941 * KEYNEXT --- HANDLE 'NXT' KEY
8942 * KEYPREV --- HANDLE 'PREV' KEY
8943 *
8944 * LEVEL 1
8945 *
8946 * KEYNXT2 EQU * 'NEXT' IS 'CLP' CONFIRMATION HERE 06651
8947 472E 472E DB60 WORD WPLVL1 06652
8948 4730 4732 WORD *+2 06653
8949 4732 C020 DAEO MOV PROGRAM,R0 06654
8950 4736 C800 DAAA MOV R0,EDITPNT 06655
8951 473A DC20 382D MOVB ENDKEYB,*R0* FILL PROGRAM MEMORY WITH 'FF' 06656
8952 473E 8800 DAE2 C R0,MAXPROG 06657
8953 4742 1501 JGT WIPEDON 06658
8954 4744 10FA JMP *-10 06659
8955 * WIPEDON EQU * 06660
8956 4746 04E0 DAAC CLR CLPFLAG SET VARIABLES TO SHOW NO PROGRAM 06661
8957 474A 0720 DAA8 SETJ PROGLN 06662
8958 474E 04E0 DAA4 CLR LINENUM 06663
8959 4752 0720 DAA2 SETJ PROMODE 06664
8960 4756 C820 DAEO DAAA MOV PROGRAM,EDITPNT 06665
8961 475C 04E0 DAAE CLR LINE99 06666
8962 4750 05A0 DAB6 INC SCRNFLAG 06667
8963 4764 0420 42FE BLWP TXTSCRN 06668
8964 4768 0380 RTWP 06669
8965 * KEYSTEP EQU * 06670
8966 476A C820 333C DAA0 MOV C2,PROGRAM BEGIN STEP MODE 06671
8967 4770 C820 DAA8 DAB4 MOV PROGLN,PROGSTEP 06672
8968 4776 C020 DAAA MOV EDITPNT,R0 06673
8969 477A 9810 382D CB *R0,ENDKEYB WARNING IF STEP AT E-O-P 06674
8970 477E 1602 JNE *+6 06675
8971 4780 04E0 D94C CLR WARNING 06676
8972 4784 0380 RTWP 06677
8973 * KEYSNEXT EQU * 06678
8974 * KEYNEXT EQU * 06679
8975 4786 0704 SETJ R4 06680
8976 4788 1001 JMP *+4 06681
8977 * KEYPREV EQU * 06682
8978 478A 04C4 CLR R4 06683
8979 478C 04C5 CLR R5 R5=SLEW NUMBER 06684
8980 478E 0201 7FFF LI R1,$7FFF LOAD INITIAL SLEWING DELAY 06685
8981 * MOV KBCODE,R6 NOTE KEY WE CAME IN WITH 06685 DEL
8982 * 00025PATCH
8983 * PROBLEM #20 - PATCH #25 (1 OF 2) 00025PATCH
8984 * 00025PATCH
8985 * CORRECTS CONTINUOUS NEXT/PREV SLEWING PROBLEM 00025PATCH
8986 * 00025PATCH
8987 4792 01A0 E00F MOVB KBCODE+1,R6 SAVE KEYCODE BYTE 00025PATCH
8988 * 00025PATCH
8989 * END OF PATCH #25, PROBLEM #20 00025PATCH
8990 4796 C020 DAA0 MOV PROGRAM,R0 06687
8991 479A 1309 JEQ NXTOK 06688
8992 479C 115C JLT NXTERR 'PREV' NOT ALLOWED IN DIRECT MODE 06689
8993 * NEXTOK EQU * 06690
8994 479E 05A0 DAA8 INC PROGLN 06691
8995 47A2 05A0 DAAA INC EDITPNT GET PAST 'NEXT' COMMAND 06692
8996 47A6 C820 333A DAB0 MOV C1,MOVFLAG BEGIN EXECUTION AT CURRENT COMMAND 06693
8997 47AC 0380 RTWP 06694
8998 * NEXTOK EQU * 06695

```

KEYNEXT / KEYPREV / KEYSTEP

06650

8999	47AE	C020	DAA8	MOV	PROGLN,R0	'NEXT' WHEN NO PROGRAM IS ERROR	06696
9000	47B2	1151		JLT	NXTERR		06697
9001	47B4	C020	DAA2	MOV	PROMODE,R0		06698
9002	47B8	131B		JEQ	NONXT		06699
9003	47BA	111A		JLT	NONXT		06700
9004	47BC	C020	DAAA	MOV	EDITPNT,R0		06701
9005	47C0	9810	37EF	CB	*R0,NXTKEYB	IS NEXT ALREADY THERE?	06702
9006	47C4	133F		JEQ	NEXTQUIT	IF SO, DON'T INSERT	06703
9007	47C6	8820	DAAE	C	LINE99,ENDLINE		06704
9008	47CC	1544		JGT	NXTERR		06705
9009	47CE	C1E0	DAE2	MOV	MAXPROG,R7		06706
9010	47D2	9817	382D	CB	*R7,ENDKEYB	IS LAST KEY 'E-O-P'?	06707
9011	47D6	163F		JNE	NXTERR	IF NOT, PROGRAM STORE FULL	06708
9012	47D8	0200	0001	LI	R0,1		06709
9013	47DC	0420	4565	BLW>	MOVSUB		06710
9014	47E0	C020	DAAA	MOV	EDITPNT,R0		06711
9015	47E4	D420	37EF	MOVB	NXTKEYB,*R0	STORE 'NXT' IN PROGRAM STORE	06712
9016	47E8	0420	4A1E	BLW>	LASTNXT	SET UP LINE99	06713
9017	47EC	05A0	DAAA	INC	EDITPNT		06714
9018		47F0		NONXT	EQU *		06715
9019	47F0	C020	DAAA	MOV	EDITPNT,R0	GET CURRENT LINE	06716
9020	47F4	C104		MOV	R4,R4	WHAT KEY DID I COME IN WITH?	06717
9021	47F6	130B		JEQ	BACKUP		06718
9022	47F8	8820	DAA8	C	PROGLN,LINE99	IF 'NXT', AM I AT END?	06719
9023	47FE	132B		JEQ	NXTERR		06720
9024	4800	05A0	DAA4	INC	LINENUM		06721
9025	4804	06A0	40A2	BL	FWDNXT	GET NEXT LINE	06722
9026	4808	05A0	DAA8	INC	PROGLN	AND INCREMENT CURRENT LINE POINTER	06723
9027	480C	1008		JMP	MOVIT		06724
9028		480E		BACKJ>	EQU *		06725
9029	480E	0600		DEC	R0		06726
9030	4810	8800	DAE0	C	R0,PROGMEM		06727
9031	4814	1120		JLT	NXTERR		06728
9032	4816	06A0	403A	BL	RVS NXT		06729
9033	481A	0620	DAA8	DEC	PROGLN		06730
9034		481E		MOVIT	EQU *		06731
9035	481E	C800	DAAA	MOV	R0,EDITPNT		06732
9036		4822		CHKSLW	EQU *		06733
9037	4822	05A0	DAB5	INC	SCRNFLAG	FLAG TXTSCRN TO UPDATE SCREEN	06734
9038	4826	0420	42FE	BLWP	TXTSCRN		06735
9039	482A	C145		MOV	R5,R5	IS IT THE FIRST 'NEXT' OR 'PREV'?	06736
9040	482C	1302		JEQ	NXTLOP	IF SO DELAY LONGER	06737
9041	482E	0201	1000	LI	R1,\$1000	SLEWING DELAY FACTOR	06738
9042		4832		NXT.OP	EQU *		06739
9043	4832	04E8	DAA2	CLR	PROMODE		06740
9044	4836	0705		SETD	R5		06741
9045				*	C KBCODE,R6		06742 DEL
9046				*			00026PATCH
9047				*	PROBLEM #20 - PATCH #26		00026PATCH
9048				*			00026PATCH
9049				*	CORRECTS CONTINUOUS NEXT/PREV SLEWING PROBLEM		00026PATCH
9050				*			00026PATCH
9051	4838	9806	E00F	CB	R6,KBCODE+1	COMPARE TO PREVIOUS KEYCODE	00026PATCH
9052				*			00026PATCH
9053				*	END OF PATCH #26, PROBLEM #20		00026PATCH
9054	483C	160F		JNE	NEXTDONE		06743
9055	483E	0601		DEC	R1		06744
9056	4840	15F8		JGT	NXTLOP		06745
9057	4842	1005		JMP	NONXT		06746

9058			4844	NEXTQUIT EQU *		06747
9059	4844	04E0	DAA2	CLR PROMJDE	NOW IN EDIT MODE	06748
9060	4848	C104		MOV R4,R4		06749
9061	484A	130D		JEQ NEXTPRV		06750
9062	484C	05A0	DAA8	INC PROGLN		06751
9063	4850	05A0	DAAA	INC EDITPNT		06752
9064	4854	10E5		JMP CHKSLEW		06753
9065			4856	NXTERR EQU *		06754
9066	4856	04E0	094A	CLR FATAL		06755
9067	485A	0380		RTWP		06756
9068			485C	NEXTJONE EQU *		06757
9069	485C	05A0	DAB6	INC SCRNFAG		06758
9070	4860	0420	42FE	BLWP TXTSCRN		06759
9071	4864	0380		RTWP		06760
9072			4866	NEXTPRV EQU *		06761
9073	4866	C020	DAAA	MOV EDITPNT,R0		06762
9074	486A	06A0	403A	BL RVS NXT		06763
9075	486E	C800	DAAA	MOV R0,EDITPNT		06764
9076	4872	10D7		JMP CHKSLEW		06765

9078				*					
9079				*	KEYLNN	---	HANDLE LNN KEY		
9080				*					
9081				*	LEVEL	1			
9082				*					
9083				*					
9084				*	ONLY REACHED IN IDLE OR EXECUTE STATES				
9085				*					
9086			4874		KEYLNN	EQU *			06767
9087	4874	C020	DAAD			MOV PROGRAM, R0			06768
9088	4878	1103				JLT LNNERR			06769
9089	487A	05A0	DAAA			INC EDITPNT			06770
9090	487E	0360				RTWP			06771
9091	4880	04E0	D94A		LNNERR	CLR FATAL	'LNN' IN IDLE MODE IS ERROR		06772
9092	4884	0360				RTWP			06773

KEYCLL

05774

```

9094 *
9095 * KEYCLL --- HANDLE CLL KEY
9096 *
9097 * LEVEL 1
9098 *
9099 KEYCLL EQU * 06775
9100 4886 C020 DAA0 MOV PROGRAM,R0 06776
9101 488A 1303 JEQ *+8 06777
9102 488C 04E0 D94A CLR FATAL ERROR IF NOT IN PROGRAM ENTRY MODE 06778
9103 4890 0380 RTWP 06779
9104 4892 C820 DAA8 DAA4 MOV PROGLN,LINENUM GET TO START OF CURRENT LINE 06780
9105 4898 0420 4534 BLWP GETLINE 06781
9106 489C C020 DAA6 MOV LINEPNT,R0 06782
9107 48A0 9810 382D CB *R0,ENDKEYB ERROR IF CLEARING END OF PROGRAM 06783
9108 48A4 1603 JNE CLRLIN 06784
9109 48A6 04E0 D94A CLR FATAL 06785
9110 48AA 100E JMP CLRLAST 06786
9111 48AC C040 CLRLIN MOV R0,R1 SAVE BEGINNING OF CURRENT LINE 06787
9112 48AE 06A0 40A2 BL FWDNXT 06788
9113 48B2 C801 DAAA MOV R1,EDITPNT 06789
9114 48B6 6001 S R1,RJ 06790
9115 48B8 0740 ABS R0 06791
9116 48BA 0500 NEG R0 06792
9117 48BC 0420 4566 BLWP MOVSUB DELETE THE PROGRAM LINE 06793
9118 48C0 04E0 DAA2 CLR PROMODE CHANGE TO EDIT MODE 06794
9119 48C4 0420 4A1E BLWP LASTNXT SET UP LINE99 06795
9120 48C8 C060 DAED CLRLAST MOV PROGMEM,R1 06796
9121 48CC 9811 382D CB *R1,ENDKEYB 06797
9122 48D0 1604 JNE *+10 CLEARED THE LAST LINE IN THE PROGRAM? 06798
9123 48D2 0720 DAA8 SETJ PROGLN YES, SET UP VARIABLES FOR NO PROGRAM 06799
9124 48D6 0720 DAA2 SETJ PROMODE 06800
9125 48DA 0380 RTWP 06801

```

9127				*				
9128				*	KEYCLP --- HANDLE 'CLP' KEY			
9129				*				
9130				*	LEVEL 1			
9131				*				
9132			48DC		KEYCLP EQU *			06803
9133	48DC	C020	DAA0		MOV PROGRAM,R0			06804
9134	48E0	1303			JEQ *+8			06805
9135	48E2	04E0	D94A		CLR FATAL	ERROR IF NOT IN PROGRAM INPUT MODE		06806
9136	48E6	0380			RTWP			06807
9137	48E8	C820	333A	DAAC	MOV C1,CLPFLAG			06808
9138	48EE	06A0	12B2		BL INITXT	WIPE SCREEN		06809
9139	48F2	0200	0007		LI R0,7			06810
9140	48F6	06A0	12DE		BL ADVLIN			06811
9141	48FA	0600			DEC R0			06812
9142	48FC	16FC			JNE *-6			06813
9143	48FE	0200	4916		LI R0,DESTROY			06814
9144	4902	0649			DECT SOFT			06815
9145	4904	C640			MOV R0,*SOFT	POINT TO MESSAGE		06816
9146	4906	0649			DECT SOFT			06817
9147	4908	C660	3364		MOV C33,*SOFT	LINE LENGTH		06818
9148	490C	06A0	136C		BL TEXT			06819
9149	4910	0420	6CF4		BLWP BUZZIT	WARN USER		06820
9150	4914	0380			RTWP			06821
9151	4916		50		DESTROY FCC "PRESS NEXT TO CONFIRM DESTRUCTION"			06822
	4917		52					
	4918		45					
	4919		53					
	491A		53					
	491B		20					
	491C		4E					
	491D		45					
	491E		58					
	491F		54					
	4920		20					
	4921		54					
	4922		4F					
	4923		20					
	4924		43					
	4925		4F					
	4926		4E					
	4927		46					
	4928		49					
	4929		52					
	492A		4D					
	492B		20					
	492C		44					
	492D		45					
	492E		53					
	492F		54					
	4930		52					
	4931		55					
	4932		43					
	4933		54					
	4934		49					
	4935		4F					
	4936		4E					

9154				*				
9155				*	KEYPAUSE --- HANDLE 'PAUSE' KEY			
9156				*				
9157				*	LEVEL 1			
9158				*				
9159				*	ONLY REACHED IN IDLE OR EXECUTE STATES			
9160				*				
9161			4938	*	KEYPAUSE EQU *			06825
9162	4938	C020	DAA0		MOV PROGRAM,R0	CLEAR KEY FIELD IN DIRECT MODE		06826
9163	493C	1507			JGT N0FKEY	TAKE DOWN 'F' AFTER 'PAUSE' ENTERED		06827
9164	493E	C820	380A	D965	MOV N0PKEY,KEY			06828
9165	4944	06A0	0316		BL SETLAST			06829
9166	4948	0420	17A5		BLW ² KEYU ²	PUT UP BLANKS IN MNEMONIC FIELD		06830
9167	494C	0720	D972	N0FKEY	SET0 RDIFLAG	UPDATE CALCULATOR READOUT FIRST		06831
9168	4950	0420	1430		BLW ² FROUT			06832
9169	4954	0201	0004		LI R1,4	R0,R1 CONTAIN DELAY VARIABLES		06833
9170	4958	0200	7FFF		LI R0,\$7FFF	LOAD IN DELAY FOR PAUSE		06834
9171	495C	0600			DEC R0			06835
9172	495E	15FE			JGT *-2			06836
9173	4960	0601			DEC R1			06837
9174	4962	15FA			JGT *-10			06838
9175	4964	C020	098A		MOV PROGRS,R0	IF IN NUMERIC ENTRY MODE,		06839
9176	4968	1302			JEQ *+6	PUT NUMBERS BACK UP		06840
9177	496A	0460	49F0		B NUMDSP			06841
9178	496E	0380			RTW ²			06842

9180				*				
9181				*	KEYRUN	---	HANDLE RUN KEY	
9182				*				
9183				*	LEVEL	1		
9184				*				
9185				*	ONLY REACHED	IN IDLE OR EXECUTE STATES		
9186				*				
9187			4970	*	KEYRJN	EQJ	*	06844
9188	4970	C020	DAA0		MOV	PROGRAM,R0	RUN DURING EXECUTION IS TREATED AS NOP	06845
9189	4974	150A			JGT	NORUN		06846
9190	4976	C820	333A		MOV	C1,PROGRAM	CHANGE TO EXECUTE STATE (RUN)	06847
9191	497C	C020	DAAA		MOV	EDITPNT,R0		06848
9192	4980	9810	3820		CB	*R0,ENDKEYS	RUN AT END OF PROGRAM IS ERROR	06849
9193	4984	1602			JNE	*+6		06850
9194	4986	04E0	D94A		CLR	FATAL		06851
9195	498A	0380			NORJN	RTW ^D		06852

KEYSTART

06853

```

9197 *
9198 * KEYSTART --- HANDLE START KEY
9199 *
9200 * LEVEL 1
9201 *
9202 * ONLY REACHED IN IDLE OR EXECUTE STATES
9203 *
9204 * KEYSTART EQU * 06854
9205 498C C020 DAED MOV PROGMEM,R0 06855
9206 4990 C800 DAAA MOV R0,EDITPNT 06856
9207 4994 9810 382D CB *R0,ENDKEYB START WITH NO PROGRAM IS ERROR 06857
9208 4998 1603 JNE **8 06858
9209 499A 04E0 D94A CLR FATAL 06859
9210 499E 0380 RTWP 06860
9211 49A0 0720 DAB4 SETD PROGSTEP RETURN TO IDLE IF IN STEP MODE 06861
9212 49A4 8820 333C DAA0 C C2,PROGRAM DON'T CHANGE TO RUN IF IN STEP MODE 06862
9213 49AA 1303 JEQ **8 06863
9214 49AC C820 333A DAA0 MOV C1,PROGRAM CHANGE TO RUN MODE 06864
9215 49B2 06A0 40D6 BL CLRPTR CLEAR RETURN POINTERS 06865
9216 49B6 04E0 DAA8 CLR PROGLN GO BACK TO PROGRAM LINE # 0 06866
9217 49BA C820 333A DAB0 MOV C1,MOVFLAG 06867
9218 49C0 0380 RTWP 06868

```

KEYF / KEYCLF / KEYNOP

06859

```

9220 *
9221 * HANDLE 'F' AND 'CLF' AND 'NOP' KEYS
9222 *
9223 * LEVEL 1
9224 *
9225 * KEYF EQU * 06870
9226 49C2 C020 DAA0 MOV PROGRAM,R0 06871
9227 49C6 130F JEQ CHKPRG 06872
9228 49C8 0420 17A6 BLWP KEYUP 06873
9229 49CC 49CC SETGOLD EQU * 06874
9230 49CC C020 3376 D94E MOV CH80,GOLD 06875
9231 49D2 0380 RTWP 06876
9232 49D4 KEYNOP EQU * 06877
9233 49D4 KEYCLF EQU * 06878
9234 49D4 C020 D98A MOV PROGRS,R0 NUMERIC ENTRY IN PROGRESS? 06879
9235 49D8 150B JGT NUMDSP PUT BACK ANY NUMBERS REMOVED 06880
9236 49DA C020 DAA0 MOV PROGRAM,R0 06881
9237 49DE 1302 JEQ *+6 06882
9238 49E0 0420 17A6 BLWP KEYUP 06883
9239 49E4 0380 RTWP 06884
9240 49E6 49E6 CHKPRG EQU * 06885
9241 49E6 C020 37F3 D966 MOV SHIFTKEY3,KEY SET KEY TO 1ST SHIFT KEYCODE 06886
9242 49EC 05CE INCT R14 NON-STANDARD RETURN TO PUT INTO PROGRAM TEXT 06887
9243 49EE 10EE JMP SETGOLD 06888
9244 49F0 49F0 NUMDSP EQU * 06889
9245 49F0 C1E0 D9C6 MOV REALTIME,R7 06890
9246 49F4 1101 JLT *+4 06891
9247 49F6 0380 RTWP 06892
9248 49F8 0560 DACA INV OKEY MAKE SURE NEXT KEY GETS DISPLAYED 06893
9249 49FC 0649 DECT SOFT 06894
9250 49FE C660 3358 MOV C16,*SOFT 06895
9251 4A02 0649 DECT SOFT 06896
9252 4A04 C660 33F4 MOV CHARSTART,*SOFT 06897
9253 4A08 0619 DEC *SOFT 06898
9254 4A0A 0649 DECT SOFT 06899
9255 4AGC 0200 DA52 LI R0,TXTBJFR 06900
9256 4A10 C64C MOV R0,*SOFT 06901
9257 4A12 0649 DECT SOFT 06902
9258 4A14 C660 335A MOV C17,*SOFT 06903
9259 4A18 06A0 136C BL TEXT 06904
9260 4A1C 0380 RTWP 06905

```

LASTNXT

05936

```

9262 *
9263 * LASTNXT --- SET UP LINE99 WITH HIGHEST LINE NUMBER IN PROGRAM STORE
9264 *
9265 * LEVEL 4
9266 *
9267 4A1E 0BC0 LASTNXT WORD WPLVL4 06907
9268 4A20 4A22 WORD *+2 06908
9269 4A22 C020 DAE0 MOV PROGEM,R0 START SEARCH AT BEGINNING OF PROGRAM MEMORY 06909
9270 4A26 04E0 DAAE CLR LINE99 INITIALIZE LINE99 TO 0 06910
9271 4A2A LN,000 EQU * 06911
9272 4A2A 8800 DAE2 C R0,MAXPROG IF AT END OF PROGRAM STOP 06912
9273 4A2E 1101 JLT *+4 06913
9274 4A30 0380 RTWP 06914
9275 4A32 9810 3820 CB *R0,ENDKEYB STOP IF AT END OF PROGRAM 06915
9276 4A36 1305 JEQ STOPLN 06916
9277 4A38 06A0 40A2 BL FWDNXT IF 'NEXT' INCREMENT LAST LINE NUMBER 06917
9278 4A3C 05A0 DAAE INC LINE99 06918
9279 4A40 10F4 JMP LNLOOP 06919
9280 4A42 0380 STOPLN RTWP 06920

```

KEYMULT / KEYDIV / KEYPLUS / KEYMINUS / KEYIFXEY / KEYIFYGTX 06922

```

9283 *
9284 * HANDLE X KEY
9285 *
9286 * LEVEL 1
9287 *
9288 * POP X AND Y, MULTIPLY THEM, AND PUSH THE RESULT ONTO X
9289 *
9290 *
9291 * VARIABLES USED (STRICT THROUGH SCALES RUN, ALTERS SOME AFTER) ---
9292 * R0 COUNTER
9293 * R1,R2 WORKING REGS
9294 * R3,R4 NUM1
9295 * R5,R6 NUM2
9296 * R7 DATA POINTER 1
9297 * R8 DATA POINTER 2
9298 * R12 OFFSET 1
9299 * R13 OFFSET 2
9300 * R14 # WFMS BEING MULTIPLIED
9301 * R15 KEY TYPE FLAG
9302 * WFM1,WFM2 WFM NUMBERS BEING MULTIPLIED (WFM2 < 0 IF FP)
9303 * W1HEAD,W2HEAD HEADER ADDRESSES FOR WFM1 AND WFM2
9304 * EXP11,EXP12 VEXP(1)
9305 * EXP21,EXP22 VEXP(2)
9306 * MAXART1,2 JP MAX MAGNITUDE IN SCALING RUN
9307 * NEGFLG NEGATIVE FLAG FOR INPUT DATA
9308 * DATA1 DATA POINTER TO START OF WFM1
9309 * DATA2 DATA POINTER TO START OF WFM2
9310 *
9311 *
9312 * NOTE ---
9313 * IF 2 FP NUMBERS ARE INPUT, A SIMPLE TRANSFER TO
9314 * FMPY IS MADE WITH THE RESULT PUSHED.
9315 * IF 2 WFMS ARE INPUT, A NORMAL RUN THROUGH PRE-SCALE
9316 * AND FINAL MPY ARE MADE.
9317 * IF 1 FP NUMBER AND 1 WFM ARE INPUT, THE WFM VERT. EXPONENT
9318 * IS MULTIPLIED BY THE FP NUMBER TO CREATE THE W0 EXPONENT.
9319 * THE OFFSET IN WFM-X IS ADDED TO EACH OF THE POINTS TO CREATE
9320 * THE WFM-0 POINTS. WFM-0 THEN HAS NO OFFSET. A BLP TO HEDFIX
9321 * REINSERTS A OFFSET AND SELECTS A 'PROPER' VERTICAL EXPONENT
9322 * AS REQUIRED.
9323 *
9324 * KEYMULT EQU * 06923
9325 4A44 04E0 DA30 CLR POINT 06924
9326 4A48 1012 JMP KEYMATH 06925
9327 * KEYDIV EQU * 06926
9328 4A4A 0720 DA30 SET0 POINT 06927
9329 4A4E 100F JMP KEYMATH 06928
9330 * KEYPLUS EQU * 06929
9331 4A50 C820 333A DA30 MOV C1,POINT 06930
9332 4A56 100B JMP KEYMATH 06931
9333 * KEYMINUS EQU * 06932
9334 4A58 C820 333C DA30 MOV C2,POINT 06933
9335 4A5E 1007 JMP KEYMATH 06934
9336 * KEYIFXEY EQU * 06935
9337 4A60 C820 333E DA30 MOV C3,POINT 06936
9338 4A66 1003 JMP KEYMATH 06937
9339 * KEYIFYGTX EQU * 06938
9340 4A68 C820 3340 DA30 MOV C4,POINT 06939
9341 4A6E EQU * 06940

```

KEYMJLT / KEYDIV / KEYPLUS / KEYMINUS / KEYIFXEQY / KEYIFYGTX 06922

```

9342 *
9343 * IT IS NOW SET UP FOR WFM * WFM
9344 4A6E 06A0 4CCE BL MATHARG 06941
9345 4A72 03E0 0A30 MOV POINT,R15 06942
9346 4A76 038E MOV R14,R14 06943
9347 4A78 154B JGT HAVWFM 06944
9348 4A7A E820 3348 0972 SOC CLINE16,ROTFLAG 06945
9349 4A80 0649 DECT SOFT BOTH ARE F? --- OPERATE ON THEM 06946
9350 4A82 0646 MOV R6,*SOFT 06947
9351 4A84 0649 DECT SOFT 06948
9352 4A86 0645 MOV R5,*SOFT 06949
9353 4A88 0043 MOV R3,R1 06950
9354 4A8A 0084 MOV R4,R2 06951
9355 4A8C 03CF MOV R15,R15 06952
9356 4A8E 1128 JLT DIVFP2 06953
9357 4A90 132A JEQ MPYFP2 06954
9358 4A92 060F DEC R15 06955
9359 4A94 132B JEQ ADDFP2 06956
9360 4A96 060F DEC R15 06957
9361 4A98 1320 JEQ SUBFP2 06958
9362 4A9A 060F DEC R15 LOGICAL COMPARE REQUESTED 06959
9363 4A9C 131A JEQ EQFP2 06960
9364 4A9E 0420 7524 BLWP FPCMPR 06961
9365 4AA2 1505 JGT DROPLIN 06962
9366 4AA4 1304 JEQ DROPLIN 06963
9367 4AA6 HOLDLIN EQU * 06964
9368 4AA6 0820 333A 095A MOV C1,TRUE 06965
9369 4AAC 102B JMP IFXBACK 06966
9370 4AAE DROPLIN EQU * 06967
9371 4AAE 04E0 095A CLR TRUE 06968
9372 4AB2 0020 0AAD MOV PROGRAM,R0 06969
9373 4AB6 1126 JLT IFXBACK 06970
9374 4AB8 0020 0AAA MOV EDITPNT,R0 06971
9375 4ABC 06A0 40A2 BL FWDNXT IF 'FALSE' BEGIN EXECUTION AT NEXT LINE 06972
9376 4AC0 0800 0AAA MOV R0,EDITPNT UPDATE EDIT POINTER 06973
9377 4AC4 05A0 0AA8 INC PROGLN UPDATE LINE POINTER 06974
9378 4AC8 0820 333A 0A30 MOV C1,MOV=LAG START EXECUTION AT CURRENT COMMAND 06975
9379 4ACE 101A JMP IFXBACK 06976
9380 4AD0 0380 RTWP 06977
9381 4AD2 EQFP2 EQU * 06978
9382 4AD2 0420 7524 BLWP FPCMPR 06979
9383 4AD6 13E7 JEQ HOLDLIN 06980
9384 4AD8 10EA JMP DROPLIN 06981
9385 4ADA SUBFP2 EQU * 06982
9386 4ADA 0420 6F60 BLWP FPSUB 06983
9387 4ADE 1008 JMP FPDON 06984
9388 4AE0 DIVFP2 EQU * 06985
9389 4AE0 0420 7050 BLWP FPDIV 06986
9390 4AE4 1005 JMP FPDON 06987
9391 4AE6 MPYFP2 EQU * 06988
9392 4AE6 0420 70AA BLWP FPMPY 06989
9393 4AEA 1002 JMP FPDON 06990
9394 4AEC ADDFP2 EQU * 06991
9395 4AEC 0420 6F84 BLWP FPADD 06992
9396 4AF0 FPDON EQU * 06993
9397 4AF0 04C0 CLR R0 06994
9398 4AF2 0079 MOV *SOFT+,R1 06995
9399 4AF4 00B9 MOV *SOFT+,R2 06996
9400 4AF6 ENDMPY EQU * 06997

```

9401	4AF6	0260	33DE		MOV SOFTST,SJFT		06998
9402	4AFA	0229	FFFA		AI SOFT,-6		06999
9403	4AFE	06A0	6962		BL PSHREG	PUSH RESULT TO USER STACK	07000
9404	4B02	1002			JMP *+6		07001
9405			4B04	IFXBACK	EQU *		07002
9406	4B04	0229	0004		AI SOFT,4		07003
9407	4B08	C379			MOV *SOFT+,R13		07004
9408	4B0A	C3B9			MOV *SOFT+,R14		07005
9409	4B0C	C3F9			MOV *SOFT+,R15		07006
9410	4B0E	0380			RTWP		07007
9411			4B10	HAVWFM	EQU *		07008
9412	4B10	028F	0003		CI R15,3		07009
9413	4B14	1102			JLT *+6		07010
9414	4B16	0460	4C34		B CMPWFM	IF R15>2 THEN LOGICAL COMPARES REQUESTED	07011
9415	4B1A	0720	D972		SETO RDTFLAG		07012
9416	4B1E	028E	0002		CI R14,2	CHECK FOR 2 WAVEFORMS	07013
9417	4B22	1303			JEQ CHKFACT		07014
9418	4B24	0420	69A8		BLWP OPWH2W0	TRANSFER OPWFM HEADER TO W0 HEADER	07015
9419	4B28	1062			JMP XFRIT		07016
9420			4B2A	CHKFACT	EQU *		07017
9421	4B2A	C160	D99A		MOV W1HEAD,R5	GET WFM1 HEADER ADDRESS	07018
9422	4B2E	C1A0	D99C		MOV W2HEAD,R6	GET WFM2 HEADER ADDRESS	07019
9423	4B32	C0C5			MOV R5,R3		07020
9424	4B34	A0E0	3340		A HEXP,R3		07021
9425	4B38	C106			MOV R6,R4		07022
9426	4B3A	A120	3340		A HEXP,R4	GET HORIZONTAL SCALE OF WFM 2	07023
9427	4B3E	C074			MOV *R4+,R1		07024
9428	4B40	C094			MOV *R4,R2		07025
9429	4B42	0073			C *R3+,R1	COMPARE MANTISSAS	07026
9430	4B44	160A			JNE BADHSCL	IF NOT EQUAL THEN SET SCALE = 1	07027
9431	4B46	8093			C *R3,R2	EXPONENTS MUST BE EQUAL ALSO	07028
9432	4B48	1608			JNE BADHSCL		07029
9433	4B4A	C0C5			MOV R5,R3		07030
9434	4B4C	C106			MOV R6,R4		07031
9435	4B4E	A0E0	3350		A HSCALD,R3	GET HORIZ UNIT ADDRESS FOR WFM1	07032
9436	4B52	A120	3350		A HSCALD,R4	GET HORIZ UNIT ADDRESS FOR WFM2	07033
9437	4B56	8513			C *R3,*R4	CHECK IF UNITS EQUAL	07034
9438	4B58	1309			JEQ XFRHZ		07035
9439	4B5A	04E0	D99E	BADHSCL	CLR HVFLAG	IF UNITS OR SCALE NOT EQUAL	07036
9440	4B5E	04E0	D9A0		CLR UNITFLG	CLEAR UNITS AND SET SCALE = 1	07037
9441	4B62	0420	4C8A		BLWP CHGSCL		07038
9442	4B66	04E0	D94C		CLR WARNING	ALSO CAUSE WARNING	07039
9443	4B6A	100A			JMP CHKVSCL		07040
9444	4B6C	C120	DA04	XFRHZ	MOV W0HEAD,R4	TRANSFER SCALE TO WFM 0	07041
9445	4B70	C004			MOV R4,R0		07042
9446	4B72	A120	3340		A HEXP,R4		07043
9447	4B76	C001			MOV R1,*R4+		07044
9448	4B78	C502			MOV R2,*R4		07045
9449	4B7A	A020	3350		A HSCALD,R0	TRANSFER UNITS TO WFM 0	07046
9450	4B7E	C413			MOV *R3,*R0		07047
9451			4B80	CHKVSCL	EQU *		07048
9452	4B80	A160	334C		A VSCALD,R5	GET ADDRESS OF VERTICAL UNITS FOR WFM1	07049
9453	4B84	A1A0	334C		A VSCALD,R5	GET ADDRESS OF VERTICAL UNITS FOR WFM2	07050
9454	4B88	8595			C *R5,*R6		07051
9455	4B8A	1307			JEQ CHKOP		07052
9456	4B8C	0720	D99E		SETO HVFLAG	SET VERTICAL SCALE FACTORS AND UNITS	07053
9457	4B90	04E0	D9A0		CLR UNITFLG		07054
9458	4B94	0420	4C8A		BLWP CHGSCL		07055
9459	4B98	102A			JMP XFRIT		07056

9460			4B9A	CHKOP	EQU *		07057
9461	4B9A	C120	DAD4		MOV W0HEAD,R4		07058
9462	4B9E	C004			MOV R4,R0		07059
9463	4BA0	A120	3338		A VEXP,R4		07060
9464	4BA4	CD20	D9A2		MOV EXP11,*R4+	MOVE VERTICAL SCALE OF WFM1 TO WFM 0	07061
9465	4BA8	C520	D9A4		MOV EXP12,*R4		07062
9466	4BAC	C0CF			MOV R15,R3	CHECK OPERATION TYE	07063
9467	4BAE	1507			JGT ADDOP		07064
9468	4BB0	0720	D99E		SETO HVFLAG	SET VERTICAL UNITS BLANK	07065
9469	4BB4	0720	D9A0		SETO UNITFLG		07066
9470	4BB8	0420	4C8A		BLWP CHGSC-		07067
9471	4BBC	1010			JMP XFRIT		07068
9472			4BBE	ADDOP	EQU *		07069
9473	4BBE	0649			DECT SOFT		07070
9474	4BC0	C660	D9A8		MOV EXP22,*SOFT		07071
9475	4BC4	0649			DECT SOFT		07072
9476	4BC6	C660	D9A6		MOV EXP21,*SOFT		07073
9477	4BCA	C060	D9A2		MOV EXP11,R1		07074
9478	4BCE	C0A0	D9A4		MOV EXP12,R2		07075
9479	4BD2	0420	7524		BLWP FPCMPR	COMPARE VERTICAL SCALE FACTORS IF + OR -	07076
9480	4BD6	1506			JGT XFRVRT		07077
9481	4BD8	1305			JEQ XFRVRT		07078
9482	4BDA	C520	D9A8		MOV EXP22,*R4	MOVE IN LARGEST VERTICAL SCALE FACTOR	07079
9483	4BDE	0644			DECT R4		07080
9484	4BE0	C520	D9A6		MOV EXP21,*R4		07081
9485	4BE4	A020	334C	XFRVRT	A VSCALD,R0		07082
9486	4BE8	C415			MOV *R5,*R0		07083
9487	4BEA	05C9			INCT SOFT	POP EXP21 AND EXP22 OFF STACK	07084
9488	4BEC	05C9			INCT SOFT		07085
9489			4BEE	XFRIT	EQU *		07086
9490	4BEE	06A0	407E		BL SCALWFM		07087
9491	4BF2	0420	78E4		BLWP SCALE	AUTOSCALE	07088
9492	4BF6	C1E0	D9B6		MOV DATA1,R7	REPOINT DATA TO START	07089
9493	4BFA	C220	D9B8		MOV DATA2,R8		07090
9494	4BFE	C3A0	DAD2		MOV W0ADD,R14	NO DATA POINTER	07091
9495	4C02	C020	D970		MOV RESOLV,R0	START COUNTER	07092
9496			4C05	MPYLOP	EQU *		07093
9497	4C06	06A0	40E2		BL SCALWORK		07094
9498	4C0A	0649		MPY2	DECT SOFT	PUSH AUTOSCALE VALUE	07095
9499	4C0C	C660	D930		MOV MAXART2,*SOFT		07096
9500	4C10	0649			DECT SOFT		07097
9501	4C12	C660	D9AE		MOV MAXART1,*SOFT		07098
9502	4C16	0420	70A0		BLWP FPMPYZ	DATA * SCALE	07099
9503	4C1A	C079			MOV *SOFT+,R1		07100
9504	4C1C	05C9			INCT SOFT	PASS UP ZERO EXPONENT ON STACK	07101
9505	4C1E	5060	D9AA		S MINART1,R1	SUBTRACT OFFSET	07102
9506	4C22	CF81			MOV R1,*R14+	STORE RESULT	07103
9507	4C24	0600			DEC R0	COUNTER	07104
9508	4C26	16EF			JNE MPYLOP		07105
9509	4C28	0200	0001		LI R0,1	PUSH W0 ON STACK	07106
9510	4C2C	04C1			CLR R1		07107
9511	4C2E	0649			DECT SOFT		07108
9512	4C30	0460	4AF6		B ENDMPY		07109
9513			4C34	C4PW=M	EQU *		07110
9514	4C34	J229	FFFC		AI R9,-4		07111
9515	4C38	C1E0	D9B6		MOV DATA1,R7		07112
9516	4C3C	C220	D9B8		MOV DATA2,R8		07113
9517	4C40	C020	D970		MOV RESOLV,R0		07114
9518	4C44	C060	D954		MOV CURSOR,R1		07115

9519	4C48	1300			JEQ CMPLOP		07116
9520	4C4A	00A0	D956		MOV CURS1,R2	ONE CURSOR UP	07117
9521	4C4E	6002			S R2,R0		07118
9522	4C50	0A12			SLA R2,1		07119
9523	4C52	A1C2			A R2,R7	START COMPARISON AT CURSOR 1	07120
9524	4C54	A202			A R2,R8		07121
9525	4C56	0601			DEC R1		07122
9526	4C58	1305			JEQ CMPLOP		07123
9527	4C5A	0020	D958		MOV CURS2,R0	IF CURSOR 2 UP, STOP COMPARISON	07124
9528	4C5E	6020	D956		S CURS1,R0	AT CURSOR 2	07125
9529	4C62	0580			INC R0		07126
9530			4C64	CMPLOP	EQU *		07127
9531	4C64	06A0	4DE2		BL SCALWORK		07128
9532	4C68	028F	0003		CI R15,3		07129
9533	4C6C	1503			JGT CMPGT		07130
9534	4C6E	C041			MOV R1,R1		07131
9535	4C70	1608			JNE DROPLIN2		07132
9536	4C72	1002			JMP CMPEND		07133
9537			4C74	CMPGT	EQU *		07134
9538	4C74	C041			MOV R1,R1		07135
9539	4C76	1507			JGT HOLDLIN2		07136
9540			4C78	CMPEND	EQU *		07137
9541	4C78	0600			DEC R0		07138
9542	4C7A	16F4			JNE CMPLOP		07139
9543	4C7C	028F	0003		CI R15,3		07140
9544	4C80	1302			JEQ HOLDLIN2		07141
9545			4C82	DROPLIN2	EQU *		07142
9546	4C82	0460	4AAE		B DROPLIN		07143
9547			4C86	HOLDLIN2	EQU *		07144
9548	4C86	0450	4AA6		B HOLDLIN		07145

CHGSCL - SET SCALE FACTOR TO 1 AND BLANK UNITS

07146

```

9550 *
9551 * SET SCALE FACTOR TO 1 AND BLANK UNITS
9552 *
9553 * LEVEL 2 ROUTINE
9554 *
9555 * INPUT - HVFLAG: =0, CHANGE HORIZONTAL
9556 * <> 0, CHANGE VERTICAL
9557 *
9558 * UNITFLG = 0 SET SCALE TO 1 AND BLANK UNITS
9559 * <> 0 BLANK UNITS ONLY
9560 *
9561 * OUTPUT- CHANGES THE HEADER OF WFM 0
9562 *
9563 4C8A D380 CHGSCL WORD WPLVL2 LEVEL 2 ROUTINE 07147
9564 4C8C 4C8E WORD **2 07148
9565 4C8E C26D 0012 MOV 16(R13),SOFT GET SOFTSTACK POINTER 07149
9566 4C92 C050 DAD4 MOV W0HEAD,R1 GET WFM 0 HEADER ADDRESS 07150
9567 4C96 0203 6EA8 LI R3,NEWHSCL MOVE THE SCALE FOR HORIZ. 07151
9568 4C9A C120 3350 MOV HSCALD,R4 07152
9569 4C9E C020 D99E MOV HVFLAG,R0 07153
9570 4CA2 1304 JEQ CHKFLG 07154
9571 4CA4 0203 6EE0 LI R3,NEWVSCL UNLESS CHANGING THE VERTICAL 07155
9572 4CA8 C120 334C MOV VSCALD,R4 07156
9573 4CAC C020 D9A0 CHKFLG MOV UNITFLG,R0 07157
9574 4CB0 160A JNE SKIPSC_ BLANK UNITS ONLY 07158
9575 4CB2 0649 DEGT SOFT 07159
9576 4CB4 C660 333A MOV FP1E,*SOFT SET SCALE FACTOR = 1 07160
9577 4CB8 0649 DEGT SOFT 07161
9578 4CBA C660 3404 MOV FP1M,*SOFT 07162
9579 4CBE 0649 DEGT SOFT PUSH WFM 0 ONTO STACK 07163
9580 4CC0 C660 3338 MOV C0,*SOFT 07164
9581 4CC4 0413 BLWP *R3 07165
9582 4CC6 A101 SKIPSC_ A R1,R4 GET UNITS ADDRESS 07166
9583 4CC8 C520 342E MOV NULWRD,*R4 MOVE IN A BLANK 07167
9584 4CCC 0380 RTWP 07168

```

9586				*					
9587				*				SET ARGUMENTS FOR BINARY MATH FUNCTIONS	
9588				*					
9589				*				BASTARD LEVEL 5 ROUTINE	
9590				*					
9591				*				INPJT --- 30TH ARGUMENTS ON JSER STACK	
9592				*					
9593				*				OUTPUT -- USER STACK POPPED ONCE	
9594				*			R3,R4	NUM1	
9595				*			R5,R6	NUM2	
9596				*			R7	DATA POINTER 1	
9597				*			R8	DATA POINTER 2	
9598				*			R12	OFFSET 1	
9599				*			R13	OFFSET 2	
9600				*			R14	NUMBER OF WFMS INPUT	
9601				*			WFM1,WFM2	WFM NUMBERS BEING OPERATED ON (<0 IF FP)	
9602				*			W1HEAD,W2HEAD	HEADER ADDRESSES OF WFM1 AND WFM2	
9603				*			EXP11,EXP12	VEXP(1)	
9604				*			EXP21,EXP22	VEXP(2)	
9605				*			DATA1	DATA POINTER TO START OF WFM1	
9606				*			DATA2	DATA POINTER TO START OF WFM2	
9607				*			R13,R14,R15	ON SOFTWARE STACK	
9608				*					
9609			4CCE		MATHARG	EQU *			07170
9610	4CCE	3260	333E			MOV SOFTST,SOFT			07171
9611	4CD2	0649				DECT SOFT		SAVE RETURN REGISTERS	07172
9612	4CD4	C64F				MOV R15,*SOFT			07173
9613	4CD6	0649				DECT SOFT			07174
9614	4CD8	C64E				MOV R14,*SOFT			07175
9615	4CDA	0649				DECT SOFT			07176
9616	4CDC	C64D				MOV R13,*SOFT			07177
9617	4CDE	0649				DECT SOFT			07178
9618	4CE0	C64B				MOV R11,*SOFT			07179
9619	4CE2	04CE				CLR R14		CLR WFM COUNT	07180
9620	4CE4	0720	D995			SETO WFM1		FLAG NO WFMS	07181
9621	4CE8	0720	D998			SETO WFM2			07182
9622	4CEC	C03A				MOV *USER+,R0		CHECK FLAG ON X	07183
9623	4CEE	1303				JEQ FPRST			07184
9624	4CF0	058E				INC R14		COUNT WFM	07185
9625	4CF2	C81A	D995			MOV *USER,WFM1		SAVE WFM #	07186
9626	4CF5	C0FA			FPRST	MOV *USER+,R3		PULL WFM # OR CONSTANT TO REGISTER	07187
9627	4CF8	C13A				MOV *USER+,R4			07188
9628	4CFA	C03A				MOV *USER+,R0		NEXT FLAG	07189
9629	4CFC	1303				JEQ FPSCND			07190
9630	4CFE	058E				INC R14		COUNT WFM	07191
9631	4D00	C81A	D998			MOV *USER,WFM2		XFR WFM #	07192
9632			4D04		FPSCND	EQU *			07193
9633	4D04	C17A				MOV *USER+,R5		GET WFM #	07194
9634	4D06	C1BA				MOV *USER+,R6			07195
9635						* R14 HAS COUNT OF NUMBER OF WFMS			
9636	4D08	8820	DA30	333C		C POINT,C2			07196
9637	4D0E	1584				JGT *+10			07197
9638	4D10	06A0	6918			BL POPSTK		IF MATH FUNCTION CALLED, POP X	07198
9639	4D14	06A0	6918			BL POPSTK		IF MATH FUNCTION CALLED, POP Y	07199
9640	4D18	C060	D995			MOV WFM1,R1			07200
9641	4D1C	1113				JLT TRYWFM2			07201
9642	4D1E	C043				MOV R3,R1		NUM1 IS WFM, NUM2 = ?	07202
9643	4D20	06A0	6984			BL ADRWFM		GET WFM POINTERS	07203
9644	4D24	C339				MOV *SOFT+,R12		HEADER ADDRESS	07204

MATHARG --- GET ARGUMENTS FOR BINARY MATH FUNCTIONS

07159

9645	4D26	C00C		MOV R12,R0		07205
9646	4D28	C80C	D99A	MOV R12,W1HEAD	SAVE HEADER ADDRESS OF WFM1	07206
9647	4D2C	A320	3348	A VOFFAB,R12	OFFSET ADDRESS	07207
9648	4D30	C31C		MOV *R12,R12	OFFSET(1)	07208
9649	4D32	A020	3338	A VEXP,R0	VERTICAL EXPONENT ADDRESS	07209
9650	4D36	C830	D9A2	MOV *R0+,EXP11	GET EXPONENT	07210
9651	4D3A	C810	D9A4	MOV *R0,EXP12		07211
9652	4D3E	C1F9		MOV *SOFT+,R7	WFM DATA START ADDRESS	07212
9653	4D40	C807	D9B6	MOV R7,DATA1	SAVE IT	07213
9654			4D44	TRYWFM2 EQU *		07214
9655	4D44	C060	D998	MOV WFM2,R1		07215
9656	4D48	1118		JLT ARGDONE		07216
9657	4D4A	C045		MOV R5,R1	NUM2 IS WFM	07217
9658	4D4C	D649		DECT SOFT	SAVE R12	07218
9659	4D4E	C64C		MOV R12,*SOFT		07219
9660	4D50	D6A0	6984	BL ADRWFM	GET ADDRESSES FOR WFM2	07220
9661	4D54	C039		MOV *SOFT+,R0	HEADER ADDRESS	07221
9662	4D56	C800	D99C	MOV R0,W2HEAD	SAVE HEADER ADDRESS OF WFM2	07222
9663	4D5A	C340		MOV R0,R13		07223
9664	4D5C	A350	3348	A VJFFAB,R13		07224
9665	4D60	C350		MOV *R13,R13	OFFSET(2)	07225
9666	4D62	A020	3338	A VEXP,R0		07226
9667	4D66	C830	D9A5	MOV *R0+,EXP21	EXP(2)	07227
9668	4D6A	C810	D9A8	MOV *R0,EXP22		07228
9669	4D6E	C239		MOV *SOFT+,R8	DATA POINTER	07229
9670	4D70	C808	D9B8	MOV R8,DATA2		07230
9671	4D74	C339		MOV *SOFT+,R12	RESTORE R12	07231
9672	4D76	C1E0	D9B6	MOV DATA1,R7		07232
9673			4D7A	ARGDONE EQU *		07233
9674	4D7A	C2F9		MOV *SOFT+,R11		07234
9675	4D7C	D45B		B *R11		07235

9677				*	DO SCALING RUN FOR FOUR-FUNCTION MATH KEYS	
9678				*		
9679				*	BASTARD LEVEL 5 ROUTINE	
9680				*		
9681			4D7E	SCALWFM	EQU *	07237
9682	4D7E	C80B	DA30		MOV R11,POINT SAVE RETURN POINTER	07238
9683	4D82	C020	D970		MOV RESOLV,R0 COUNTER	07239
9684	4D86	0600			DEC R0	07240
9685	4D88	06A0	4DE2		BL SCALWORK GET FIRST DATA POINT	07241
9686	4D8C	C801	D9AE		MOV R1,MAXART1	07242
9687	4D90	C801	D9AA		MOV R1,MINART1	07243
9688	4D94	C802	D9B0		MOV R2,MAXART2	07244
9689	4D98	C802	D9AC		MOV R2,MINART2	07245
9690			4D9C	LOPSCALX	EQU *	07246
9691	4D9C	06A0	4DE2		BL SCALWORK GET NEXT DATA POINT	07247
9692	4DA0	0649			DECT SOFT	07248
9693	4DA2	C660	D9B0		MOV MAXART2,*SOFT	07249
9694	4DA6	0649			DECT SOFT	07250
9695	4DA8	C660	D9AE		MOV MAXART1,*SOFT	07251
9696	4DAC	0420	7524		BLWP FPCMPR	07252
9697	4DB0	1104			JLT MPYNOMAX	07253
9698	4DB2	C801	D9AE		MOV R1,MAXART1	07254
9699	4DB6	C802	D9B0		MOV R2,MAXART2	07255
9700			4DBA	MPYNOMAX	EQU *	07256
9701	4DBA	05C9			INCT SOFT	07257
9702	4DBC	C660	D9AC		MOV MINART2,*SOFT	07258
9703	4DC0	0649			DECT SOFT	07259
9704	4DC2	C660	D9AA		MOV MINART1,*SOFT	07260
9705	4DC6	0420	7524		BLWP FPCMPR <?	07261
9706	4DCA	1504			JGT MPYNOMIN	07262
9707	4DCC	C801	D9AA		MOV R1,MINART1	07263
9708	4DD0	C802	D9AC		MOV R2,MINART2	07264
9709			4DD4	MPYNOMIN	EQU *	07265
9710	4DD4	0229	0004		AI SOFT,4	07266
9711	4DD8	0600			DEC R0	07267
9712	4DDA	15E0			JGT LOPSCALX	07268
9713	4DDC	C2E0	DA30		MOV POINT,R11	07269
9714	4DE0	045B			B *R11	07270

SCALWORK --- DO MAIN WORK FOR SCALING RUN OF WFMS

07271

```

9716 *
9717 * SCALWORK ---
9718 *
9719 * DO MAIN WORK FOR SCALING RUN OF WFMS AND CALCULATE NEW DATA VALUE
9720 *
9721 * INPUTS ---
9722 * R15 =-1 FOR DIV/ =0 FOR MPY/ =1 FOR ADD/ =2 FOR SUBTRACT/ >=3F
9723 * WFM1,WFM2 HAVE WFM NUMBERS TO OPERATE ON (-1 IF FP)
9724 * R3,R4 HAVE NUMBER 1
9725 * R5,R6 HAVE NUMBER 2
9726 * R7,R8 HAVE DATA POINTERS FOR WFM1 AND WFM2
9727 * R12,R13 HAVE OFFSETS FOR WFM1 AND WFM2
9728 * EXP11,EXP12 HAS VEXP FOR WFM1
9729 * EXP21,EXP22 HAS VEXP FOR WFM2
9730 *
9731 * OUTPUT ---
9732 * ALL ABOVE IS UNCHANGED
9733 * R1,R2 HAS DATA VALUE FOR NXT POINT OF WFM
9734 * R1 HAS 0,NEGATIVE,OR POSITIVE VALU FOR =<> ON LOGICAL COMPARES
9735 *
9736 * BASTARD LEVEL 5 ROUTINE
9737 *
9738 * SCALWORK EQU * 07272
9739 4DE2 C060 D998 MOV WFM2,R1 07273
9740 4DE6 110C JLT SCALW1 07274
9741 4DE8 C078 MOV *R6+,R1 NUM2 IS WFM 07275
9742 4DEA A040 A R13,R1 07276
9743 4DEC 04C2 CLR R2 07277
9744 4DEE 0649 DECT SOFT 07278
9745 4DF0 C660 D9A5 MOV EXP22,*SOFT 07279
9746 4DF4 0649 DECT SOFT 07280
9747 4DF6 C660 D9A6 MOV EXP21,*SOFT 07281
9748 4DFA 0420 78AA BLWP FMPY GET DATA VALUE FOR NUM2 POINT 07282
9749 4DFE 1004 JMP SCALW2 07283
9750 * SCALW1 EQU * 07284
9751 4E00 0649 DECT SOFT NUM2 IS FP 07285
9752 4E02 C646 MOV R6,*SOFT 07286
9753 4E04 0649 DECT SOFT 07287
9754 4E06 C645 MOV R5,*SOFT 07288
9755 * SCALW2 EQU * 07289
9756 4E08 C060 D996 MOV WFM1,R1 07290
9757 4E0C 110E JLT SCALW3 07291
9758 4E0E C077 MOV *R7+,R1 NUM1 IS WFM 07292
9759 4E10 A04C A R12,R1 07293
9760 4E12 04C2 CLR R2 07294
9761 4E14 0649 DECT SOFT 07295
9762 4E16 C660 D9A4 MOV EXP12,*SOFT 07296
9763 4E1A 0649 DECT SOFT 07297
9764 4E1C C660 D9A2 MOV EXP11,*SOFT 07298
9765 4E20 0420 78AA BLWP FMPY VALUE FOR DATA POINT 07299
9766 4E24 C079 MOV *SOFT+,R1 PUT IN REGISTERS 07300
9767 4E26 C0B9 MOV *SOFT+,R2 07301
9768 4E28 1002 JMP SCALW4 07302
9769 * SCALW3 EQU * 07303
9770 4E2A C043 MOV R3,R1 NUM1 IS FP 07304
9771 4E2C C0B4 MOV R4,R2 07305
9772 * SCALW4 EQU * 07306
9773 4E2E 028F 0002 CI R15,2 07307
9774 4E32 150F JGT SCALSUB 07308

```

SCALWORK --- DO MAIN WORK FOR SCALING RUN OF W=MS

07271

9775	4E34	03CF		MOV R15,R15	WHAT OPERATION WAS REQUESTED?	07309
9776	4E36	1104		JLT SCALDIV		07310
9777	4E38	1506		JGT SCALADD		07311
9778	4E3A	0420	70AA	BLWP FPMPY	MPY WAS REQUESTED	07312
9779	4E3E	100B		JMP SCALDATA		07313
9780			4E40	SCALDIV EQU *		07314
9781	4E40	0420	7050	BLWP FPOIV	DIV WAS REQUESTED	07315
9782	4E44	100B		JMP SCALDATA		07316
9783			4E46	SCALADD EQU *		07317
9784	4E46	83E0	333A	C C1,R15	ADD OR SUB?	07318
9785	4E4A	1603		JNE SCALSUB		07319
9786	4E4C	0420	6F84	BLWP FPADD	ADD WAS REQUESTED	07320
9787	4E50	1002		JMP SCALDATA		07321
9788			4E52	SCALSUB EQU *		07322
9789	4E52	0420	6F60	BLWP FPSUB		07323
9790			4E56	SCALDATA EQU *		07324
9791	4E56	C079		MOV *SOFT+,R1	PULL FINAL DATA VALUE TO R1 AND R2	07325
9792	4E58	C039		MOV *SOFT+,R2		07326
9793	4E5A	045B		B *R11		07327


```

9795 *
9796 * HANDLE SGN AND ABS KEYS
9797 *
9798 * LEVEL 1 ROUTINE
9799 *
9800 * CREATE WD AS 0 OR 1 VOLT ON A POINT-BY-POINT BASIS
9801 * COMPARISON OF OPW SIGN
9802 *
9803 4E5C 04C6 KEYABS CLR R6 07329
9804 4E5E 1001 JMP *+4 07330
9805 4E60 0706 KEYSN SET0 R6 07331
9806 4E62 0720 D972 SET0 RDTFLAG 07332
9807 4E66 06A0 690E BL POPREG 07333
9808 4E6A 1336 JEQ CONST2 07334
9809 4E6C C060 DAD2 MOV WOADD,R1 07335
9810 4E70 C0A0 D95C MOV OPWFMD,R2 07336
9811 4E74 C0E0 D970 MOV RESOLV,R3 07337
9812 4E78 C160 D95E MOV OPWFMD,R5 07338
9813 4E7C A150 3348 A VJFFAB,R5 POINT TO OFFSET 07339
9814 4E80 0204 0667 SIGNLOP LI R4,$0667 07340
9815 4E84 C032 MOV *R2+,R0 07341
9816 4E86 A015 A *R5,R0 ADD OFFSET FOR SIGN 07342
9817 4E88 1505 JGT CHKDONE 07343
9818 4E8A 04C4 CLR R4 07344
9819 4E8C C000 MOV R0,R0 07345
9820 4E8E 1302 JEQ CHKDONE 07346
9821 4E90 0204 F999 LI R4,$F999 07347
9822 4E94 0740 CHKDONE ABS R0 07348
9823 4E96 1901 JNO *+4 07349
9824 4E98 0600 DEC R0 07350
9825 4E9A 6015 S *R5,R0 REMOVE OFFSET AGAIN 07351
9826 4E9C 1902 JNO *+6 07352
9827 4E9E 0200 7FFF LI R0,$7FFF 07353
9828 4EA2 C186 MOV R6,R6 07354
9829 4EA4 1601 JNE *+4 07355
9830 4EA6 C100 MOV R0,R4 07356
9831 4EA8 CC44 MOV R4,*R1+ 07357
9832 4EAA 0603 DEC R3 07358
9833 4EAC 16E9 JNE SIGNLOP 07359
9834 4EAE 0420 69A8 BLWP OPWH2WD 07360
9835 4EB2 C186 MOV R6,R6 07361
9836 4EB4 1308 JEQ SIGNDON 07362
9837 4EB6 C060 DAD4 MOV WOHED,R1 HEADER NEEDS MODS FOR SGN 07363
9838 4EBA A050 3348 A VJFFAB,R1 07364
9839 4EBE 04D1 CLR *R1 NO OFFSET IN SGN 07365
9840 4EC0 0720 D99E SET0 HVFLAG MOD VERTICAL 07366
9841 4EC4 04E0 D9A0 CLR UNITFLG SET SCALE = 1 AND BLANK UNITS 07367
9842 4EC8 0420 4C8A BLWP CHGSCL 07368
9843 4ECC EQU * SIGNDON 07369
9844 4ECC C020 333A MOV C1,R0 07370
9845 4ED0 04C1 CLR R1 07371
9846 4ED2 06A0 6962 BL PSHREG 07372
9847 4ED6 0380 RTWP 07373
9848 4ED8 C186 CONST2 MOV R6,R6 07374
9849 4EDA 1310 JEQ ABSCON 07375
9850 4EDC C0E0 3404 MOV FP1M,R3 07376
9851 4EE0 C120 333A MOV FP1E,R4 07377
9852 4EE4 C041 MOV R1,R1 07378
9853 4EE6 1505 JGT SINCON 07379
    
```

9854	4EE8	0503		NEG R3		07380
9855	4EEA	C041		MOV R1,R1		07381
9856	4EEC	1102		JLT SINCON		07382
9857	4EEE	04C3		CLR R3		07383
9858	4EF0	04C4		CLR R4		07384
9859	4EF2	C043	SINCON	MOV R3,R1		07385
9860	4EF4	C084		MOV R4,R2		07386
9861	4EF6	06A0	6962	BL PSHREG		07387
9862	4EFA	0380		RTWP		07388
9863	4EFC	0741	ABSCON	ABS R1		07389
9864	4EFE	1902		JND *+6	CHECK FOR ABS(\$8000)	07390
9865	4F00	0911		SRL R1,1	SHIFT TO \$4000	07391
9866	4F02	0582		INC R2	INCREMENT EXPONENT TO OFFSET SHIFT	07392
9867	4F04	06A0	6962	BL PSHREG		07393
9868	4F08	0380		RTWP		07394

9870					*****			
9871					**			**
9872					**	'EXP', 'LN', & 'SQRT' KEY HANDLERS		**
9873					**			**
9874					**	LEVEL 1 ROUTINE		**
9875					**			**
9876					**	NOTE ---		**
9877					**	SINCE THESE ROUTINE ARE SINGLE ARGUMENT FUNCTIONS THEY		**
9878					**	ARE IMPLEMENTED IN THE SAME MAIN ROUTINE WITH ONLY THE		**
9879					**	SUBROUTINE CALLED BE VARIED. EACH KEY LOADS THE ADDRESS		**
9880					**	OF THE SUBROUTINE TO BE USED IN R6. THIS IS THEN USED		**
9881					**	AS AN INDIRECT REGISTER FOR CALLING THE ROUTINE.		**
9882					**			**
9883					*****			*****
9884	4F0A	0206	7382	KEYEXP	LI	R6,FPNEXP	SET FLAG FOR EXPONENTIATION	07396
9885	4F0E	1005			JMP	POPARG		07397
9886	4F10	0206	7298	KEYLN	LI	R6,FPNLOG	SET FLAG FOR NATURAL LOGARITHM	07398
9887	4F14	1002			JMP	POPARG		07399
9888	4F16	0206	748A	KEYSQRT	LI	R6,FP SQRT	SET FLAG FOR SQUARE ROOT	07400
9889	4F1A	06A0	690E	POPARG	BL	POPREG	POP ARGUMENT OFF USER STACK	07401
9890	4F1E	1610			JNE	WFMARG	IF R0 <> 0, ARGUMENT IS A WAVEFORM	07402
9891								
9892					*****	INPUT ARGUMENT IS A FLOATING POINT NUMBER		*****
9893								
9894	4F20	0206	7382		CI	R6,FPNEXP	LN & SQRT USE ABS(X)	07403
9895	4F24	1309			JEQ	EXPF		07404
9896	4F26	0741			ABS	R1		07405
9897	4F28	1903			JNO	*+8		07406
9898	4F2A	0911			SRL	R1,1	CHECK FOR 8000	07407
9899	4F2C	0582			INC	R2		07408
9900	4F2E	1002			JMP	*+6		07409
9901	4F30	1503			JGT	EXPF		07410
9902	4F32	1302			JEQ	EXPF		07411
9903	4F34	04E0	D94C		CLR	WARNING	NEGATIVE ARGUMENT FOR LN & SQRT IS WARNING	07412
9904	4F38	0649		EXPF	DECT	SOFT	IF ARGUMENT IS A FLOATING POINT NUMBER	07413
9905	4F3A	C642			MOV	R2,*SOFT	PUSH IT ONTO SOFTSTACK	07414
9906	4F3C	0649			DECT	SOFT		07415
9907	4F3E	C641			MOV	R1,*SOFT		07416
9908	4F40	0416			BLW	*R6	EVALUATE CORRECT FUNCTION	07417
9909	4F42	1902			JNO	*+6		07418
9910	4F44	04E0	D94A		CLR	FATAL	IF OVERFLOW, THERE WAS A DOMAIN ERROR	07419
9911	4F48	04C0			CLR	R0	FLAG FOR FLOATING POINT NUMBER	07420
9912	4F4A	C079			MOV	*SOFT+,R1	POP COMPUTED VALUE OFF SOFTSTACK	07421
9913	4F4C	C0B9			MOV	*SOFT+,R2		07422
9914	4F4E	06A0	6962		BL	PSHREG	PUSH COMPUTED VALUE ONTO USER STACK	07423
9915	4F52	E820	3348	D972	SOC	CLINE16,RDTFLAG	UPDATE READOUT LINE # 16	07424
9916	4F58	0380			RTW		RETURN TO CALLER	07425
9917								
9918					*****	INPUT ARGUMENT IS A WAVEFORM		*****
9919								
9920	4F5A	C0E0	D95E	WFMARG	MOV	OPWFMH,R3	GET HEADER ADDRESS FOR OPW	07426
9921	4F5E	C103			MOV	R3,R4		07427
9922	4F60	A0E0	3348		A	VOFFAB,R3	GET VERTICAL OFFSET FOR THIS WAVEFORM	07428
9923	4F64	C0D3			MOV	*R3,R3		07429
9924	4F66	A120	3338		A	VEXP,R4	GET VERTICAL MULTIPLIER FOR THIS WAVEFORM	07430
9925	4F6A	C074			MOV	*R4+,R1		07431
9926	4F6C	C094			MOV	*R4,R2		07432
9927	4F6E	0420	69A8		BLW	OPWH2W0	TRANSFER HEADER FROM OPW TO W0	07433
9928	4F72	0720	D99E		SET0	HVFLAG	BLANK OUT VERTICAL UNITS	07434

9929	4F76	0720	D9A0	SET0	UNITFL3		07435
9930	4F7A	0420	4C8A	BLWP	CHGSCL		07436
9931							
9932				*****	FIND THE MINIMUM AND THE MAXIMUM OF THE WAVEFORM	*****	
9933							
9934	4F7E	C320	D970	MOV	RESOLV,R12	GET WAVEFORM'S RESOLUTION	07437
9935	4F82	C120	D95C	MOV	OPWFMD,R4	GET DATA ADDRESS FOR OPW	07438
9936	4F86	C1F4		MOV	*R4+,R7	MOVE FIRST DATA POINT TO BOTH	07439
9937	4F88	A1C3		A	R3,R7	MAXIMUM AND MINIMUM	07440
9938	4F8A	0286	73B2	CI	R6,FPNEXP	CHECK IF ABSOLUTE VALUE SHOULD BE TAKEN	07441
9939	4F8E	1303		JEQ	*+8		07442
9940	4F90	0747		ABS	R7		07443
9941	4F92	1901		JND	*+4	CHECK FOR \$8000	07444
9942	4F94	0607		DEC	R7		07445
9943	4F96	C207		MOV	R7,R8		07446
9944	4F98	060C		DEC	R12		07447
9945	4F9A	C174		WFMMAX	MOV	*R4+,R5	07448
9946	4F9C	A143		A	R3,R5	ADD VERTICAL OFFSET TO DATA POINT	07449
9947	4F9E	0286	73B2	CI	R6,FPNEXP	CHECK IF ABSOLUTE VALUE SHOULD BE TAKEN	07450
9948	4FA2	1303		JEQ	*+8		07451
9949	4FA4	0745		ABS	R5	GET ITS ABSOLUTE VALUE	07452
9950	4FA6	1901		JND	*+4	CHECK FOR \$8000	07453
9951	4FA8	0605		DEC	R5		07454
9952	4FAA	01C5		C	R5,R7	CHECK TO SEE IF THIS IS A NEW MAXIMUM	07455
9953	4FAC	1503		JGT	LNEWMAX		07456
9954	4FAE	0205		C	R5,R8	CHECK TO SEE IF THIS IS A NEW MINIMUM	07457
9955	4FB0	1103		JLT	LNEWMIN		07458
9956	4FB2	1003		JMP	LNEXTPT		07459
9957	4FB4	C1C5		LNEWMAX	MOV	R5,R7	07460
9958	4FB6	1001		JMP	LNEXTPT	REPLACE MAXIMUM WITH NEW VALUE	07461
9959	4FB8	C205		LNEWMIN	MOV	R5,R8	07462
9960	4FBA	060C		LNEXTPT	DEC	R12	07463
9961	4FBC	15EE		JGT	WFMMAX	IF MORE POINTS GET NEXT POINT	07464
9962							
9963				*****	EVALUATE THE FUNCTION AT THE MINIMUM AND MAXIMUM	*****	
9964							
9965	4FBE	0649		DECT	SOFT	EVALUATE THE FUNCTION USING MINIMUM	07465
9966	4FC0	0409		CLR	*SOFT		07466
9967	4FC2	0649		DECT	SOFT		07467
9968	4FC4	C648		MOV	R8,*SOFT		07468
9969	4FC6	0420	70AA	BLWP	FPMPY	MULTIPLY POINT BY ITS VERTICAL MULTIPLIER	07469
9970	4FCA	0416		BLWP	*R6	EVALUATE FUNCTION	07470
9971	4FCC	0649		DECT	SOFT		07471
9972	4FCE	0409		CLR	*SOFT		07472
9973	4FD0	0649		DECT	SOFT	EVALUATE FUNCTION AT MAXIMUM	07473
9974	4FD2	0647		MOV	R7,*SOFT		07474
9975	4FD4	0420	70AA	BLWP	FPMPY		07475
9976	4FD8	0416		BLWP	*R6	EVALUATE FUNCTION	07476
9977	4FDA	C839	D9AE	XPSAME	MOV	*SOFT+,MAXART1	07477
9978	4FDE	C839	D9B0	MOV	*SOFT+,MAXART2	SETUP VALUES FOR AUTOSCALE	07478
9979	4FE2	C839	D9AA	MOV	*SOFT+,MINART1		07479
9980	4FE6	C839	D9AC	MOV	*SOFT+,MINART2		07480
9981	4FEA	0420	78E4	BLWP	SCALE	SCALE NEW WAVEFORM TO 1-2-5 RATIO	07481
9982							
9983				*****	EVALUATE THE FUNCTION OVER THE ENTIRE WAVEFORM	*****	
9984							
9985	4FEE	C120	D95C	MOV	OPWFMD,R4	GET OPWFMD DATA ADDRESS	07482
9986	4FF2	C160	DA02	MOV	WOADD,R5	GET WO'S DATA ADDRESS	07483
9987	4FF6	C020	D970	MOV	RESOLV,R0		07484

9988	4FFA	C1C1		MOV	R1,R7	SAVE OLD VEXP	07485
9989	4FFC	C202		MOV	R2,R8		07486
9990	4FFE	C047		CALCFM	MOV	R7,R1	RESTORE ORIGINAL VEXP
9991	5000	C088		MOV	R8,R2		07488
9992	5002	0649		DECT	SOFT		07489
9993	5004	0409		CLR	*SOFT		07490
9994	5006	0649		DECT	SOFT	PUT NEXT DATA POINT ON STACK	07491
9995	5008	C674		MOV	*R4+,*SOFT		07492
9996	500A	A643		A	R3,*SOFT	ADD OLD VOFFAB TO DATA POINT	07493
9997	500C	0286	7332	CI	R6,FPNEXP	CHECK IF ABSOLUTE VALUE SHOULD BE TAKEN	07494
9998	5010	1308		JEQ	EXPWM		07495
9999	5012	0759		ABS	*SOFT	TAKE ABSOLUTE VALUE OF THIS POINT	07496
10000	5014	1902		JNO	*+6	CHECK FOR \$8000	07497
10001	5016	0619		DEC	*SOFT	SET ABS(\$8000) TO \$7FFF	07498
10002	5018	1002		JMP	*+6		07499
10003	501A	1503		JGT	EXPWM	NEGATIVE ARGUMENT FOR LN & SQRT IS WARNING	07500
10004	501C	1302		JEQ	EXPWM		07501
10005	501E	04E0	D94C	CLR	WARNING		07502
10006	5022	0420	70AA	EXPWM	BLWP	FMPY	MULTIPLY THIS BY OLD VEXP
10007	5026	0416		BLWP	*R6	COMPUTE THE CORRECT FUNCTION	07504
10008	5028	C060	D9AE	MOV	MAXART1,R1		07505
10009	502C	C0A0	D980	MOV	MAXART2,R2	RETRIEVE SCALE MULTIPLIER FROM AUTOSCALE	07506
10010	5030	0420	70A0	BLWP	FMPYZ	PUT DATA WITHIN A SCREENS WIDTH	07507
10011	5034	C579		MOV	*SOFT+,*R5	MOVE THIS POINT TO W0	07508
10012	5036	05C9		INCT	SOFT		07509
10013	5038	6D60	D9AA	S	MINART1,*R5+	POSITION POINT ONSCREEN	07510
10014	503C	0600		DEC	R0	SEE IF THERE ARE MORE POINTS TO PROCESS	07511
10015	503E	15DF		JGT	CALCFM		07512
10016	5040	0580		INC	R0	FLAG TO INDICATE HAVEFORM NUMBER	07513
10017	5042	04C1		CLR	R1		07514
10018	5044	04C2		CLR	R2	MOVE 'OPW 0' TO USER STACK	07515
10019	5046	06A0	6962	BL	PSHREG		07516
10020	504A	0720	D972	SETJ	RODFLAG	SET ROJUT FLAG TO CHANGE ALL FLAGS	07517
10021	504E	0380		RTWP		RETURN TO CALLER	07518

10023				*				
10024				*	HANDLE SMOOTH KEY			
10025				*				
10026				*	LEVEL 1 ROUTINE			
10027				*				
10028				*	USE SQUARE CONVOLVE FUNCTION WITH WINDOW OF X DIVISIONS (0<X<=2)			
10029				*				
10030				*	N=NUMBER OF POINTS TO SMOOTH OVER (=10 * X)			
10031				*	N IS FORCED TO BE ODD TO ENFORCE SIMITY OF SMOOTHED WAVEFORM.			
10032				*	WINDOW GOES FROM -N/2 TO N/2 POINTS AROUND DESIRED POINT			
10033				*	ENDPOINTS ARE EXCESSIVLY WEIGHTED BY EXTENDING END POINTS			
10034				*	FOR FIXED WINDOW SIZE			
10035				*				
10036				*				
10037				*	NOTE ---			
10038				*	USES GNOBAS FOR DATA BUFFER			
10039				*				
10040				*	REGISTER USEAGE---			
10041				*	R0 BUFFER COUNTER			
10042				*	R1 BUFFER POINTER			
10043				*	R2 WD POINTER			
10044				*	R3 OPW POINTER			
10045				*	R4 RESOLV COUNTER			
10046				*	R5,R6 DP SUM OF N POINTS			
10047				*	R7,R8 SAME AS R5,R6 (SHIFTED HERE TO SP)			
10048				*	R12 SHIFT COUNT FOR WINDOW DIVIDE			
10049				*	R13 WINDOW "N"			
10050				*				
10051			5050		BAD18 EQU *			07520
10052	5050	06A0	6952		BL PSHREG			07521
10053	5054	C043			MOV R3,R1			07522
10054	5056	C084			MOV R4,R2			07523
10055	5058	04C0			CLR R0			07524
10056	505A	06A0	6962		BL PSHREG			07525
10057	505E	04E8	D94A		CLR FATAL			07526
10058	5062	0380			RTWP			07527
10059	5064	C020	33D4		NOGOOD7 MOV WSTK,R0			07528
10060	5068	06A0	6936		BL PSHSTK			07529
10061	506C	04E0	D94A		CLR FATAL			07530
10062	5070	0380			RTWP			07531
10063			5072		KEYSMOOTH EQU *			07532
10064	5072	0720	D972		SETO RDTFLAG			07533
10065	5076	06A0	690E		BL POPREG			07534
10066	507A	16F4			JNE NOGOOD7 MUST BE A CONSTANT			07535
10067	507C	C041			MOV R1,R1			07536
10068	507E	11F2			JLT NOGOOD7 MUST BE >=0			07537
10069	5080	0649			DECT SOFT			07538
10070	5082	C660	333C		MOV C2,*SOFT PUSH FP 2			07539
10071	5086	0649			DECT SOFT			07540
10072	5088	C660	3404		MOV FP1M,*SOFT			07541
10073	508C	0420	7524		BLWP FPCMPR			07542
10074	5090	15E9			JGT NOGOOD7 MUST BE <=2			07543
10075	5092	C0C1			MOV R1,R3			07544
10076	5094	C102			MOV R2,R4			07545
10077	5096	06A0	690E		BL POPREG			07546
10078	509A	13DA			JEQ BAD18 Y MUST BE A WFM			07547
10079	509C	C060	D970		MOV RESOLV,R1			07548
10080	50A0	06A0	757E		BL INT2FP FLOAT RESOLJTION			07549
10081	50A4	05C9			INCT SOFT POP FP 2 AND SET UP FOR PUSHING			07550

KEYSMOJTH

07519

10082	50A6	C642		MOV R2,*SOFT	PUSH RESOLUTION ONTO STACK	07551
10083	50A8	0649		DECT SOFT		07552
10084	50AA	C641		MOV R1,*SOFT		07553
10085	50AC	C060	33FE	MOV FP10M,R1		07554
10086	50B0	C0A0	3340	MOV FP10E,R2		07555
10087	50B4	0420	7050	BLWP FPDIV	RESOLV / 10	07556
10088	50B8	C043		MOV R3,R1	GET WIDTH AGAIN	07557
10089	50BA	C084		MOV R4,R2		07558
10090	50BC	0420	70AA	BLWP FMPY	NUMBER OF POINTS DESIRED (FLOATING POINT)	07559
10091	50C0	C079		MOV *SOFT+,R1		07560
10092	50C2	C0B9		MOV *SOFT+,R2		07561
10093	50C4	06A0	75A0	BL FP2INT	NUMBER OF POINTS REQUESTED (INTEGER)	07562
10094	50C8	0811		SRA R1,1		07563
10095	50CA	0A11		SLA R1,1		07564
10096	50CC	0581		INC R1	NUMBER OF POINTS TO USE (THIS IS AN ODD NUMBER)	07565
10097	50CE	0649		DECT SOFT	SAVE R13,R14,R15	07566
10098	50D0	C640		MOV R13,*SOFT		07567
10099	50D2	0649		DECT SOFT		07568
10100	50D4	C64E		MOV R14,*SOFT		07569
10101	50D6	0649		DECT SOFT		07570
10102	50D8	C64F		MOV R15,*SOFT		07571
10103	50DA	C341		MOV R1,R13	SAVE WINDOW WIDTH	07572
10104	50DC	0420	69A8	BLWP OPWH2W0	TRANSFER HEADER FROM OPW TO W0	07573
10105	50E0	0811		SRA R1,1		07574
10106	50E2	C381		MOV R1,R14	R14 HAS END-OVER COUNT	07575
10107	50E4	C128	D970	MOV RESOLV,R4		07576
10108	50E8	610E		S R14,R4	RESOLV COUNT ONLY INTERIOR POINTS	07577
10109	50EA	0604		DEC R4		07578
10110	50EC	C060	33E0	MOV GND0BAS,R1	GET POINTERS	07579
10111	50F0	C3E0	0A02	MOV W0ADD,R15		07580
10112	50F4	C0E0	D95C	MOV OPWFMD,R3		07581
10113	50F8	04C5		CLR R5		07582
10114	50FA	04C6		CLR R5		07583
10115				*		
10116				*	SET UP BUFFER	
10117				*		
10118	50FC	C00E		MOV R14,R0	1/2 BUFFER FILLED WITH OPW POINT 0	07584
10119	50FE	1606		JNE SMTHIT	DONE ALREADY IF LESS THAN TWO POINTS WIDE	07585
10120	5100	C0A0	D970	MOV RESOLV,R2		07586
10121	5104	CFF3		MOV *R3+,*R15+	MOVE OPWFMD TO WFM 0	07587
10122	5106	0602		DEC R2	(NO SMOOTH)	07588
10123	5108	15FD		JGT *-4		07589
10124	510A	1035		JMP SMTHQUT		07590
10125	510C	C453		SMTHIT MOV *R3,*R1		07591
10126	510E	06A0	5188	BL DPADD		07592
10127	5112	0600		DEC R0		07593
10128	5114	16FB		JNE *-8		07594
10129	5116	C00E		MOV R14,R0		07595
10130	5118	0580		INC R0		07596
10131	511A	C473		MOV *R3+,*R1	OTHER HALF WITH OPW DATA	07597
10132	511C	06A0	5188	BL DPADD		07598
10133	5120	0600		DEC R0		07599
10134	5122	16FB		JNE *-8		07600
10135				*		
10136				*	START SMOOTHING LOOP	
10137				*		
10138			5124	SMTHLOP EQU *		07601
10139	5124	0720	D986	SETJ DATA1		07602
10140	5128	C086		MOV R6,R2		07603

10141	512A	0045			MOV R5,R1		07604
10142	512C	1103			JLT *+8		07605
10143	512E	04E0	D986		CLR DATA1	DP SUM IS POSITIVE	07606
10144	5132	1005			JMP POSNOW		07607
10145	5134	0541			INV R1	DP SUM IS NEGATIVE --- NEGATE IT	07608
10146	5136	0542			INV R2		07609
10147	5138	0582			INC R2		07510
10148	513A	1701			JNC **4		07511
10149	513C	0581			INC R1		07512
10150			513E		PJSNOW EQU *		07513
10151	513E	3C4D			DIV R13,R1		07614
10152	5140	C820	D986	D986	MOV DATA1,DATA1		07615
10153	5146	1301			JEQ POSRSLT		07616
10154	5148	0501			NEG R1	IF DP SUM WAS NEGATIVE, SO IS RESULT	07617
10155			514A		POSRSLT EQU *		07618
10156	514A	CFC1			MOV R1,*R15+	SAVE SMOOTHED DATA	07619
10157	514C	C060	33E0		MOV GND0BAS,R1		07520
10158	5150	0511			NEG *R1		07521
10159	5152	06A0	5188		BL DPADD	SUBTRACT OUT FIRST POINT	07522
10160	5156	31C1			MOV R1,R7		07523
10161	5158	0641			DECT R1	REPOINT TO BUFFER START	07624
10162	515A	C00D			MOV R13,R0		07625
10163	515C	0600			DEC R0	ROLL (WINDOW - 1) DATA POINTS	07626
10164	515E	CC77			MOV *R7+,*R1+	ROLL BUFFER	07627
10165	5160	0600			DEC R0		07628
10166	5162	16FD			JNE *-4		07629
10167	5164	0604			DEC R4		07530
10168	5166	1104			JLT SMTHEND		07531
10169	5168	C473			MOV *R3+,*R1	ADD NEW DATA IN BUFFER	07632
10170	516A	06A0	5188		SMTHCNT BL DPADD	ADD NEW POINT TO DP SUM	07533
10171	516E	10DA			JMP SMTHLOP		07634
10172	5170	060E			SMTEND DEC R14		07635
10173	5172	15FB			JGT SMTHCNT		07636
10174	5174	13FA			JEQ SMTHCNT		07637
10175			5176		SMTQUIT EQU *		07638
10176	5176	C3F9			MOV *SOFT+,R15		07539
10177	5178	C389			MOV *SOFT+,R14		07640
10178	517A	C379			MOV *SOFT+,R13		07641
10179	517C	0200	0001		LI R0,1		07642
10180	5180	04C1			CLR R1		07643
10181	5182	06A0	6962		BL PSHREG	PUSH W0 ON USER STACK	07644
10182	5186	0380			RTW		07645
10183	5188	0707			DPADD SETC R7		07646
10184	518A	C231			MOV *R1+,R8		07647
10185	518C	1101			JLT **4		07648
10186	518E	04C7			CLR R7		07649
10187	5190	A147			A R7,R5		07550
10188	5192	A188			A R8,R6		07651
10189	5194	1701			JNC **4		07652
10190	5196	0585			INC R5		07553
10191	5198	045B			B *R11		07654

'HXPD' - HORIZONTAL EXPAND

07655

```

10193 *
10194 *   HANDLE HEXP KEY (HORIZONTAL EXPAND)
10195 *
10196 *   LEVEL 1 ROUTINE
10197 *
10198 *   ALGORITHM USED:
10199 *
10200 *   EACH POINT IN THE WAVEFORM IS MULTIPLIED BY THE EXPANSION
10201 *   RATIO. THE NEAREST INTEGER FUNCTION IS THEN APPLIED TO THIS
10202 *   PRODUCT TO GET THE NEW POINT NUMBER WHICH WILL BE ASSIGNED THE
10203 *   ORIGINAL POINT NUMBER'S VERTICAL VALUE
10204 *   ANY POINTS IN THE EXPANDED WAVEFORM THAT ARE UNDEFINED ARE
10205 *   ASSIGNED VALUES USING THE LINEAR INTERPOLATION FUNCTION
10206 *   BETWEEN DEFINED POINTS. THE POINT AT THE RIGHT EDGE OF THE
10207 *   SCREEN IS ALSO CALCULATED USING LINEAR INTERPOLATION.
10208 *
10209 519A 0450 0A9C BAD7 B NOGOOD2 07656
10210 519E 519E KEY<X>D EQU * 07657
10211 519E 8820 D954 333C C CURSOR,C2 07658
10212 51A4 1625 JNE POPHSC 07659
10213 51A6 06A0 690E BL POPREG POP X 07660
10214 51AA 13F7 JEQ BAD7 X MUST BE WFM 07661
10215 51AC C060 D958 MOV CURS2,R1 07662
10216 51B0 6050 D956 S CURS1,R1 DELTA POINT 07663
10217 51B4 06A0 69F2 BL HORDLTA GET HOR VALUES OF SPREAD 07664
10218 51B8 0649 DECT SOFT 07665
10219 51BA C642 MOV R2,*SOFT PUSH SCREEN TIME 07666
10220 51BC 0649 DECT SOFT 07667
10221 51BE C641 MOV R1,*SOFT 07668
10222 51C0 C060 D970 MOV RESOLV,R1 07669
10223 51C4 0601 DEC R1 07670
10224 51C6 06A0 757E BL INT2FP FLOAT RESOLV 07671
10225 51CA J420 7050 BLWP FPDIV HSCL / (R-1) 07672
10226 51CE C079 MOV *SOFT+,R1 POP NEW HEXP 07673
10227 51D0 C099 MOV *SOFT,R2 07674
10228 51D2 C160 D95E MOV OPWFMM,R5 07675
10229 51D6 A150 3340 A HEXP,R5 07676
10230 51DA 05C5 INCT R5 07677
10231 51DC C655 MOV *R5,*SOFT PUSH OLD HEXP 07678
10232 51DE 0645 DECT R5 07679
10233 51E0 0649 DECT SOFT 07680
10234 51E2 C655 MOV *R5,*SOFT 07681
10235 51E4 0420 7050 BLWP FPDIV OLD HEXP / NEW HEXP 07682
10236 51E8 C0F9 MOV *SOFT+,R3 07683
10237 51EA C119 MOV *SOFT,R4 07684
10238 51EC 0649 DECT SOFT REPUSH RATIO 07685
10239 51EE 1018 JMP GOTRATJ 07686
10240 51F0 06A0 690E POPHSC BL POPREG 0 OR 1 CURSOR ON 07687
10241 51F4 16D2 JNE BAD7 X IS WFM 07688
10242 51F6 C041 MOV R1,R1 07689
10243 51F8 13D0 JEQ BAD7 07690
10244 51FA 11CF JLT BAD7 X MUST BE >= 0 07691
10245 51FC 0649 DECT SOFT PUSH RATIO 07692
10246 51FE C642 MOV R2,*SOFT 07693
10247 5200 0649 DECT SOFT 07694
10248 5202 C641 MOV R1,*SOFT 07695
10249 5204 C0C1 MOV R1,R3 SAVE RATIO ALSO 07696
10250 5206 C102 MOV R2,R4 07697
10251 5208 06A0 690E BL POPREG POP Y 07698

```

'HKPD' - HORIZONTAL EXPAND

07655

10252	520C	1609		JNE GOTRATD	Y MUST BE WFM	07599
10253	520E	06A0	6962	BL PSHREG	RESTORE Y	07700
10254	5212	C043		MOV R3,R1		07701
10255	5214	3084		MOV R4,R2		07702
10256	5216	06A0	6962	BL PSHREG	RESTORE X	07703
10257	521A	04E0	D94A	CLR FATAL	SET ERROR FLAG	07704
10258	521E	0380		RTWP		07705
10259			5220	GOTRATD EQU *		07706
10260	5220	0420	69A8	BLWP OPWH2WD		07707
10261	5224	C1A0	DAD4	MOV W0HEAD,R5	SAVE OLD HEXP	07708
10262	5228	A1A0	3340	A HEXP,R6		07709
10263	522C	C076		MOV *R6+,R1		07710
10264	522E	0649		DECT SOFT		07711
10265	5230	C676		MOV *R6+,*SOFT		07712
10266	5232	0649		DECT SOFT		07713
10267	5234	C641		MOV R1,*SOFT		07714
10268	5236	C104		MOV R4,R4		07715
10269	5238	1502		JGT *+6		07716
10270	523A	04E0	D94C	CLR WARNING		07717
10271	523E	C084		MOV R4,R2	PUSH RATIO AGAIN	07718
10272	5240	C043		MOV R3,R1		07719
10273	5242	0420	7050	BLWP FPDIV	NEW HEXP = OLD HEXP / RATIO	07720
10274	5246	C1A0	DAD4	MOV W0HEAD,R5		07721
10275	524A	A1A0	3340	A HEXP,R6		07722
10276	524E	C0B9		MOV *SOFT+,*R6+		07723
10277	5250	C5B9		MOV *SOFT+,*R6		07724
10278	5252	C104		MOV R4,R4		07725
10279	5254	1502		JGT *+6		07726
10280	5256	0460	535A	B 4XPLT1	FOR EXPANSION RATIO LESS THAN 1, DIFFERENT ROUTINE	07727
10281	525A	C1A0	D970	MOV RESOLV,R6		07728
10282	525E	04C7		CLR R7		07729
10283	5260	04C1		CLR R1		07730
10284	5262	C0A0	D954	MOV CURSOR,R2	ARE CURSORS JP?	07731
10285	5266	1305		JEQ NOCRS	... IF NOT, XFER ENTIRE OPWFM TO WFM 0	07732
10286	5268	C050	D956	MOV CURS1,R1		07733
10287	526C	C1C1		MOV R1,R7	SAVE CURSOR POSITION	07734
10288	526E	6181		S R1,R6	SHIFT OPERATIONAL WFM ONLY	07735
10289	5270	0A11		SLA R1,1		07736
10290	5272	C146		NOCRS MOV R6,R5	R5 NOW CONTAINS 1ST BAD POINT #	07737
10291	5274	A060	D95C	A JPHFMD,R1		07738
10292	5278	C0A0	DAD2	MOV W0ADD,R2		07739
10293	527C	CCB1		MOV *R1+,*R2+	SHIFT WFM TO PLACE C1 ON LEFT OF SCREEN	07740
10294	527E	0606		DEC R6		07741
10295	5280	15FD		JGT *-4		07742
10296	5282	C187		MOV R7,R6	CURSOR POS. IS THE # OF BAD POINTS	07743
10297	5284	1304		JEQ NOBAD	NO BAD POINTS, CURS1 AT SCREEN EDGE	07744
10298	5286	0641		DECT R1	GET LAST GOOD POINT	07745
10299	5288	CC91		MOV *R1,*R2+	PROPAGATE LAST GOOD POINT	07746
10300	528A	0606		DEC R6	ACROSS SCREEN	07747
10301	528C	15FD		JGT *-4		07748
10302	528E	C060	D970	NOBAD MOV RESOLV,R1		07749
10303	5292	0601		DEC R1		07750
10304	5294	06A0	757E	BL INT2FP	CONVERT RESOLUTION TO FP	07751
10305	5298	0649		DECT SOFT		07752
10306	529A	C642		MOV R2,*SOFT	SET UP FOR FP DIVIDE	07753
10307	529C	0649		DECT SOFT		07754
10308	529E	C641		MOV R1,*SOFT		07755
10309	52A0	C043		MOV R3,R1	GET RATIO/	07756
10310	52A2	C084		MOV R4,R2		07757

'HXPD' - HORIZONTAL EXPAND

07655

10311	52A4	0420	7050	BLWP FPDIV	RESOLUTION DIVIDED BY RATIO	07758
10312	52A8	C079		MOV *SOFT+,R1	GIVES THE LAST POINT EXPANDED	07759
10313	52AA	C099		MOV *SOFT,R2		07760
10314	52AC	0649		DECT SOFT		07761
10315	52AE	06A0	75A6	BL FPTRNC		07762
10316	52B2	C001		MOV R1,R0	SAVE THE POINT NUMBER	07763
10317	52B4	8045		C R5,R1	FIRST BAD POINT # > LAST POINT EXPANDED?	07764
10318	52B6	1502		JGT *+6	YES, WFM EXPANDED OK (NO WARNING)	07765
10319	52B8	04E0	094C	CLR WARNING	IF EXTENDING BAD POINTS, GIVE WARNING	07766
10320	52BC	06A0	757E	BL INT2FP	COMPUTE DISTANCE FROM POINT	07767
10321	52C0	0420	6F60	BLWP FPSUB	TO SCREEN EDGE	07768
10322	52C4	C200		MOV R0,R0		07769
10323	52C6	0A18		SLA R0,1		07770
10324	52C8	A220	DAD2	A WJADD,R0		07771
10325	52CC	C180		MOV *R0+,R6	GET DATA AT LAST POINT	07772
10326	52CE	C108		MOV *R0,R7	GET DATA AT 1ST OFFSCREEN POINT	07773
10327	52D0	51C6		S R6,R7	COMPUTE THE DIFFERENCE	07774
10328	52D2	C047		MOV R7,R1		07775
10329	52D4	06A0	757E	BL INT2FP		07776
10330	52D8	0420	70AA	BLWP FPMPY		07777
10331	52DC	C046		MOV R6,R1		07778
10332	52DE	06A0	757E	BL INT2FP		07779
10333	52E2	0420	6F84	BLWP FPADD		07780
10334	52E6	C079		MOV *SOFT+,R1		07781
10335	52E8	C089		MOV *SOFT+,R2		07782
10336	52EA	06A0	75A0	BL FP2INT	CONVERT INTERPOLATED VALUE TO INTEGER	07783
10337	52EE	0A10		SLA R0,1		07784
10338	52F0	A020	DAD2	A WJADD,R0		07785
10339	52F4	C140		MOV R0,R5		07786
10340	52F6	C020	0970	MOV RESOLV,R0		07787
10341	52FA	0600		DEC R0		07788
10342	52FC	0A10		SLA R0,1		07789
10343	52FE	A020	DAD2	A WJADD,R0		07790
10344	5302	C401		MOV R1,*R0		07791
10345	5304	C045		SET4LOP MOV R5,R1	MAP ORIGIN ADDRESS	07792
10346	5306	6060	DAD2	S WJADD,R1		07793
10347	530A	0811		SRA R1,1	ORIGIN POINT #	07794
10348	530C	111A		JLT SETH00V	FINISHED WHEN < 0	07795
10349	530E	06A0	757E	BL INT2FP		07796
10350	5312	0649		DECT SOFT	PUSH RATIO	07797
10351	5314	C644		MOV R4,*SOFT		07798
10352	5316	0649		DECT SOFT		07799
10353	5318	C643		MOV R3,*SOFT		07800
10354	531A	0420	70AA	BLWP FPMPY	RATIO * PNT #	07801
10355	531E	C079		MOV *SOFT+,R1		07802
10356	5320	C089		MOV *SOFT+,R2		07803
10357	5322	06A0	75A0	BL FP2INT	POINT # TO MAP INTO	07804
10358	5326	0A11		SLA R1,1		07805
10359	5328	A060	DAD2	A WJADD,R1	NE POINTS ADDRESS	07806
10360	532C	C455		MOV *R5,*R1	MAP OPW INTO W0	07807
10361	532E	0640		HKI_PNT DECT R0	MOVE DOWN LAST ADDRESS MAPPED	07808
10362	5330	8001		C R1,R0	ANY UNMAPPED POINTS?	07809
10363	5332	1504		JGT HLOPING		07810
10364	5334	1303		JEQ HLOPING		07811
10365	5336	C420	3390	MOV BADPNT,*R0	WIPE ANY UNMAPPED POINTS	07812
10366	533A	10F9		JMP HKILPNT		07813
10367	533C	C001		HLOPING MOV R1,R0	SAVE THIS ADDRESS AS LAST ADDRESS MAPPED	07814
10368	533E	0645		DECT R5	MAP NEXT POINT	07815
10369	5340	10E1		JMP SETHLOP		07816

'HXPD' - HORIZONTAL EXPAND

07655

10370	5342	04C1		SETHDON	CLR R1		07817
10371	5344	C020	333A		MOV C1,R0		07818
10372	5348	06A0	6962		BL PSHREG	PUSH WFM 0 ONTO USERS STACK	07819
10373	534C	0649			DECT SOFT		07820
10374	534E	04D9			CLR *SOFT		07821
10375	5350	0420	69F8		BLWP FILLPNT	INTERPOLATE MISSING POINTS ON W0	07822
10376	5354	0720	D972		SET0 RDTFLAG		07823
10377			5358	HXPJON	EQU *		07824
10378	5358	0388			RTWP		07825
10379			535A	HXPLT1	EQU *		07826
10380	535A	C160	D95C		MOV OPWFMD,R5	START OF ORIGIN	07827
10381	535E	04C6			CLR R6	R6 IS POINT COUNTER	07828
10382	5360	C020	D970		MOV RESOLV,R0	R0 HAS NUMBER OF POINTS TO MAP	07829
10383	5364	C060	D954		MOV CURSOR,R1		07830
10384	5368	1305			JEQ LT1NOCRS		07831
10385	536A	C050	D956		MOV CURS1,R1	START ORIGIN AT C1, IF UP	07832
10386	536E	5001			S R1,R0		07833
10387	5370	0A11			SLA R1,1		07834
10388	5372	A141			A R1,R5		07835
10389			5374	LT1NOCRS	EQU *		07836
10390	5374	C046			MOV R6,R1		07837
10391	5376	06A0	757E		BL INT2FP	FP POINT # TO MAP	07838
10392	537A	0649			DECT SOFT	PUSH RATIO	07839
10393	537C	C644			MOV R4,*SOFT		07840
10394	537E	0649			DECT SOFT		07841
10395	5380	C643			MOV R3,*SOFT		07842
10396	5382	0420	70AA		BLWP FPMPY	RATIO * POINT #	07843
10397	5386	C079			MOV *SOFT+,R1		07844
10398	5388	C0B9			MOV *SOFT+,R2		07845
10399	538A	06A0	75A0		BL FP2INT	NEW PNT # TO MAP INTO	07846
10400	538E	C0B5			MOV *R5+,R2	VALUE TO MAP OVER	07847
10401	5390	8A11			SLA R1,1		07848
10402	5392	A060	DA02		A WJADD,R1	ABSOLUTE ADDRESS TO STORE DATA	07849
10403	5396	CC42			MOV R2,*R1+	STORE DATA	07850
10404	5398	0586			INC R6	INCREMENT POINT COUNTER	07851
10405	539A	0600			DEC R0		07852
10406	539C	16E8			JNE LT1NOCRS		07853
10407	539E	04E0	D94C		CLR WARNING	WARN USER OF MAP WITH EXTENDED POINTS	07854
10408	53A2	C220	D970		MOV RESOLV,R8		07855
10409	53A6	0608			DEC R8		07856
10410	53A8	0A18			SLA R8,1		07857
10411	53AA	A220	DA02		A WJADD,R8	LAST ADDRESS TO CLEAR OUT	07858
10412	53AE	8201			C R1,R8		07859
10413	53B0	15C8			JGT SETHDON		07860
10414	53B2	CC60	3390		MOV BADPNT,*R1+		07861
10415	53B6	10FB			JMP *-8		07862

```

10417 *****
10418 **
10419 ** HANDLER FOR 'ITRP' KEY
10420 **
10421 ** LEVEL 1 ROUTINE
10422 **
10423 ** INPUT: END POINTS FROM CURSORS AND USERSTACK
10424 ** OUTPUT: INTERPOLATED WFM IN W0
10425 **
10426 ** NOTE ---
10427 ** THE ENDPOINTS FOR THE INTERPOLATION DEPEND ON HOW
10428 ** MANY CURSORS ARE BEING DISPLAYED:
10429 **
10430 ** 0 - ENDPOINTS FROM X & Y OF USERSTACK
10431 **
10432 ** 1 - ENDPOINT FROM CURSOR #1 AND X OF USERSTACK
10433 **
10434 ** 2 - ENPOINTS FROM CURSORS
10435 **
10436 *****
    
```

10437	53B8	8820	D954	333A	KEYITRP C	CURSOR,C1	HOW MANY CURSORS ARE BEING DISPLAYED?	07864
10438	53BE	152B			JGT	POP3	2, USE CURSORS	07865
10439	53C0	130E			JEQ	POP1	1, USE CURSOR #1 & X OF USERSTACK	07866
10440	53C2	06A0	690E		POP2	BL	POPREG	0, USE X & Y OF USERSTACK
10441	53C6	C142			MOV	R2,R5		07868
10442	53C8	C101			MOV	R1,R4	SAVE DATA POPPED FROM X	07869
10443	53CA	C0C0			MOV	R0,R3		07870
10444	53CC	1661			JNE	BADPOP2	IF WFM # THEN ERROR	07871
10445	53CE	06A0	601E		BL	FP2ELE	CONVERT VALUE TO POINT NUMBER	07872
10446	53D2	C1A0	D94A		MOV	FATAL,R6	IS VALUE WITHIN HORIZONTAL RANGE?	07873
10447	53D6	155C			JGT	BADPOP2	NO, THIS IS AN ERROR	07874
10448	53D8	135B			JEQ	BADPOP2		07875
10449	53DA	C181			MOV	R1,R6	SAVE POINT NUMBER AS ENDPOINT	07876
10450	53DC	1002			JMP	POP1+4		07877
10451	53DE	C1A0	D956		POP1	MOV	CURS1,R5	CURSOR #1 IS FIRST VALUE
10452	53E2	06A0	690E		BL	POPREG	POP VALUE OFF USERSTACK	07879
10453	53E6	06A0	6962		BL	PSHREG	LEAVE VALUE ON STACK	07880
10454	53EA	C000			MOV	R0,R0		07881
10455	53EC	164C			JNE	BADPOP1	IF WFM # THEN ERROR	07882
10456	53EE	06A0	601E		BL	FP2ELE	CONVERT VALUE TO POINT NUMBER	07883
10457	53F2	C1E0	D94A		MOV	FATAL,R7	IS VALUE WITHIN HORIZONTAL RANGE?	07884
10458	53F6	1547			JGT	BADPOP1	NO, THIS IS AN ERROR	07885
10459	53F8	1346			JEQ	BADPOP1		07886
10460	53FA	C1C1			MOV	R1,R7	SAVE POINT NUMBER AS ENDPOINT	07887
10461	53FC	8820	D954	333A	C	CURSOR,C1	SHOULD ANOTHER VALUE BE RESTORED IN STACK?	07888
10462	5402	1101			JLT	*+4		07889
10463	5404	1012			JMP	ITRP		07890
10464	5406	C085			MOV	R5,R2		07891
10465	5408	C044			MOV	R4,R1	RESTORE VALUE	07892
10466	540A	C003			MOV	R3,R0		07893
10467	540C	C147			MOV	R7,R5		07894
10468	540E	06A0	6962		BL	PSHREG		07895
10469	5412	C1C5			MOV	R5,R7		07896
10470	5414	100A			JMP	ITRP		07897
10471	5416	06A0	690E		POP3	BL	POPREG	POP WAVEFORM NUMBER OFF USERSTACK
10472	541A	133D			JEQ	BADPOP3	IF CONSTANT, THEN ERROR	07899
10473	541C	04C1			CLR	R1	PUSH '0 WFM' ONTO USERSTACK	07900
10474	541E	06A0	6962		BL	PSHREG		07901
10475	5422	C1A0	D956		MOV	CURS1,R6	CURSOR #1 IS FIRST VALUE	07902

10476	5426	C1E0	D958	MOV	CURS2,R7	CURSOR #2 IS SECOND VALUE	07903	
10477								
10478				***** TRANSFER OPERATIONAL WAVEFORM TO WAVEFORM #0 *****				
10479								
10480	542A	0420	69A8	ITRP	BLWP	OPWH2W0	TRANSFER HEADER FROM OPW TO W0	
10481	542E	C060	DAD2	MOV	W0ADD,R1	ADDRESS OF W0'S DATA	07904	
10482	5432	C0A0	D95C	MOV	OPWFM0,R2	ADDRESS OF OPW'S DATA	07905	
10483	5436	C0E0	D970	MOV	RESOLV,R3	CURRENT WAVEFORM RESOLUTION	07906	
10484	543A	CC72		MOV	*R2+,*R1+	TRANSFER WAVEFORM DATA FROM OPW TO W0	07907	
10485	543C	0603		DEC	R3		07908	
10486	543E	15FD		JGT	*-4		07909	
10487								
10488				***** PUT MINIMUM ENDPOINT IS R6, MAXIMUM IN R7 *****				
10489								
10490	5440	81C6		C	R6,R7	COMPARE ENDPOINTS	07910	
10491	5442	1316		JEQ	NOITRP	IF EQUAL, NO INTERPOLATION IS NEEDED	07911	
10492	5444	1103		JLT	*+8		07912	
10493	5446	C206		MOV	R6,R8	SWAP R6 & R7 SO R6 GETS MINIMUM ENDPOINT	07913	
10494	5448	C187		MOV	R7,R6	R7 GET MAXIMUM ENDPOINT	07914	
10495	544A	C1C8		MOV	R8,R7		07915	
10496	544C	0586		INC	R6	INCREMENT FIRST ENDPOINT	07916	
10497	544E	81C6		C	R6,R7	COMPARE ENDPOINTS	07917	
10498	5450	130F		JEQ	NOITRP	IF EQUAL, NO INTERPOLATION IS NEEDED	07918	
10499								
10500				***** REPLACE POINT BETWEEN ENDPOINTS WITH UNDEFINED VALUE *****				
10501								
10502	5452	C220	3390	MOV	BADPNT,R8	VALUE FOR UNDEFINED DATA POINT	07919	
10503	5456	0A16		SLA	R6,1		07920	
10504	5458	0A17		SLA	R7,1	CONVERT ENDPOINTS TO ADDRESSES	07921	
10505	545A	A1A0	DAD2	A	W0ADD,R6		07922	
10506	545E	A1E0	DAD2	A	W0ADD,R7		07923	
10507	5462	CD88		MOV	R8,*R6+	MOVE UNDEFINED VALUE TO POINT	07924	
10508	5464	81C6		C	R6,R7	FILL UNTIL SECOND ENDPOINT	07925	
10509	5466	1AFD		JL	*-4		07926	
10510	5468	0649		DECT	SOFT		07927	
10511	546A	04D9		CLR	*SOFT		07928	
10512	546C	0420	69F8	BLWP	FILLPNT	SUBROUTINE TO INTERPOLATE UNDEFINED POINTS	07929	
10513	5470	0720	D972	NOITRP	SET0	RDTFLAG	UPDATE ENTIRE CALCULATOR READOUT	
10514	5474	0700		SET0	R0	FLAG INDICATING WFM #	07930	
10515	5476	04C1		CLR	R1	WFM #0	07931	
10516	5478	06A0	6962	BL	PSHREG	PUSH 0 WFM	07932	
10517	547C	0420	6C98	BLWP	OPWCHG	MAKE 0 NEW OPWFM	07933	
10518	5480	06A0	690E	BL	POPREG	POP 0 WFM	07934	
10519	5484	0388		RTWP			07935	
10520	5486	8820	D954	333A	BADPOP1	C	CURSOR,C1	IF NO CURSORS ANOTHER PUSH IS NEEDED
10521	548C	1101		JLT	BADPOP2			07936
10522	548E	1005		JMP	ERRPOP			07937
10523	5490	C085		BADPOP2	MOV	R5,R2	RESORE DATA	
10524	5492	C044		MOV	R4,R1		07938	
10525	5494	C003		MOV	R3,R0		07939	
10526	5496	06A0	6962	BADPOP3	BL	PSHREG	PUSH DATA BACK ONTO USER STACK	
10527	549A	04E0	D94A	ERRPOP	CLR	FATAL	SET ERROR FLAG	
10528	549E	0380		RTWP			07940	

KEYINTG

07947

```

10530 *****
10531 **
10532 ** HANDLER FOR 'INTG' KEY **
10533 **
10534 ** LEVEL 1 ROUTINE **
10535 **
10536 ** NOTE --- **
10537 ** THE INTEGRAL OF THE OPERATIONAL WAVEFORM IS CALCULATED **
10538 ** USING THE TRAPEZOIDAL RULE: **
10539 **
10540 ** INT(0) = 0 **
10541 **
10542 ** FOR N = 1 TO P/W-1: **
10543 **
10544 ** INT(N) = (Y(0)/2 + Y(1) + Y(2) + . . . + Y(N)/2) * 4EXP **
10545 **
10546 *****
10547 54A0 06A0 690E KEYINTG BL POPREG POP (X) OFF USERSTACK 07948
10548 54A4 1605 JNE INTGWFM 07949
10549 54A6 06A0 6962 BL PSHREG (X) MUST BE A WFM 07950
10550 54AA 04E0 094A CLR FATAL 07951
10551 54AE 0380 RTWP 07952
10552 54B0 0420 69A8 INTGWFM BLWP OPWH2W0 TRANSFER HEADER FROM OPW TO W0 07953
10553 54B4 04C3 CLR R3 07954
10554 54B6 04C4 CLR R4 MINIMUM IS IN R3,R4 07955
10555 54B8 04C5 CLR R5 07956
10556 54BA 04C6 CLR R6 MAXIMUM IS IN R5,R6 07957
10557 54BC C1E0 095C MOV OPWFMD,R7 ADDRESS OF OPWFM'S DATA 07958
10558 54C0 C220 0970 MOV RESOLV,R8 CURRENT WAVEFORM RESOLUTION 07959
10559 54C4 0608 DEC R8 07960
10560 54C6 C020 095E MOV OPWFMH,R0 ADDRESS OF OPWFM'S HEADER 07961
10561 54CA A020 3348 A VOFFAB,R0 ADDRESS OF VERTICAL OFFSET 07962
10562 54CE C010 MOV *R0,R0 VERTICAL OFFSET 07963
10563 54D0 C077 MOV *R7+,R1 VALUE OF FIRST POINT OF WAVEFORM 07964
10564 54D2 A048 A R0,R1 ADD VERTICAL OFFSET 07965
10565 54D4 C081 MOV R1,R2 07966
10566 54D6 08F1 SRA R1,15 CREATE SIGNED DOUBLE-WORD VALUE 07967
10567 54D8 0812 SRA R2,1 DIVIDE VALUE BY 2 07968
10568 54DA 1703 JNC SUMIPNT 07969
10569 54DC 0582 INC R2 ROUND RESULT 07970
10570 54DE 1701 JNC SUMIPNT 07971
10571 54E0 0581 INC R1 07972
10572 54E2 C2F7 SJMIPNT MOV *R7+,R11 VALUE OF NEXT POINT OF WAVEFORM 07973
10573 54E4 A2C0 A R0,R11 ADD VERTICAL OFFSET 07974
10574 54E6 C308 MOV R11,R12 07975
10575 54E8 08FB SRA R11,15 CREATE SIGNED DOUBLE-WORD VALUE 07976
10576 54EA 081C SRA R12,1 DIVIDE VALUE BY 2 07977
10577 54EC 1703 JNC *+8 ROUND RESULT 07978
10578 54EE 0582 INC R2 (NOTE - ROUND IS ADDED TO SUM SO 07979
10579 54F0 1701 JNC *+4 ADDITION OF SECOND HALF DOESN'T 07980
10580 54F2 0581 INC R1 RESULT IN ROUND BEING ADDED TWICE.) 07981
10581 54F4 A048 A R11,R1 07982
10582 54F6 A08C A R12,R2 ADD Y(N)/2 07983
10583 54F8 1701 JNC *+4 07984
10584 54FA 0581 INC R1 07985
10585
10586 ***** CHECK FOR NEW MINIMUM AND MAXIMUM *****
10587
10588 54FC 80C1 CHKMIN C R1,R3 COMPARE IF VALUE IS NEW MINIMUM 07986

```


10589	54FE	1103		JLT	NEWIMIN	YES, SAVE THIS VALUE AS NEW MINIMUM	07987
10590	5500	1505		JGT	CHKIMAX	NO, CHECK IF VALUE IS NEW MAXIMUM	07988
10591	5502	8102		C	R2,R4		07989
10592	5504	1803		J4	CHKIMAX		07990
10593	5506	C0C1		NEWIMIN	MOV R1,R3	SAVE NEW MINIMUM	07991
10594	5508	C102		MOV	R2,R4		07992
10595	550A	1007		JMP	NXTIPNT		07993
10596	550C	8141		CHKIMAX	C R1,R5	COMPARE IF VALUE IS NEW MAXIMUM	07994
10597	550E	1503		JGT	NEWIMAX	YES, SAVE THIS VALUE AS NEW MAXIMUM	07995
10598	5510	1104		JLT	NXTIPNT	NO, CHECK NEXT POINT OF WAVEFORM	07996
10599	5512	8182		C	R2,R5		07997
10600	5514	1A02		JL	NXTIPNT		07998
10601	5516	C141		NEWIMAX	MOV R1,R5	SAVE NEW MAXIMUM	07999
10602	5518	C182		MOV	R2,R6		08000
10603	551A	A04B		NXTIPNT	A R11,R1	ADD SECOND HALF OF POINT VALUE (SEE NOTE	08001
10604	551C	A08C		A	R12,R2	ABOVE) TO KEEP RUNNING SUM	08002
10605	551E	1701		JNC	*+4		08003
10606	5520	0581		INC	R1		08004
10607	5522	0608		DEC	R8	DECREMENT REMAINING POINT COUNT	08005
10608	5524	15DE		JGT	SUMIPNT	IF MORE POINTS, CONTINUE	08006
10609							
10610						***** CALCULATE AUTOSCALE VALUES *****	
10611							
10612	5526	C1E0	D95E	MOV	OPWFH,R7	ADDRESS OF OPWFH'S HEADER	08007
10613	552A	C087		MOV	R7,R2		08008
10614	552C	A0A0	3338	A	VEXP,R2	ADDRESS OF VERTICAL MULTIPLIER	08009
10615	5530	C072		MOV	*R2+,R1	VERTICAL MULTIPLIER	08010
10616	5532	C092		MOV	*R2,R2		08011
10617	5534	A1E0	3340	A	HEXP,R7	ADDRESS OF HORIZONTAL INCREMENT	08012
10618	5538	0229	FFFC	AI	SOFT,-4		08013
10619	553C	CE77		MOV	*R7+,*SOFT+	PUSH HORIZONTAL INCREMENT ONTO SOFTSTACK	08014
10620	553E	C657		MOV	*R7,*SOFT		08015
10621	5540	0649		DECT	SOFT		08016
10622	5542	0420	70AA	BLW ^D	FPMPY	VERTICAL MULTIPLIER * HORIZONTAL INCREMENT	08017
10623	5546	C079		MOV	*SOFT+,R1		08018
10624	5548	C0B9		MOV	*SOFT+,R2		08019
10625	554A	0229	FFF4	AI	SOFT,-12		08020
10626	554E	CE43		MOV	R3,*SOFT+	PUSH MINIMUM VALUE ONTO SOFTSTACK	08021
10627	5550	CE44		MOV	R4,*SOFT+		08022
10628	5552	CE60	3358	MOV	C16,*SOFT+		08023
10629	5556	CE45		MOV	R5,*SOFT+	PUSH MAXIMUM VALUE ONTO SOFTSTACK	08024
10630	5558	CE46		MOV	R6,*SOFT+		08025
10631	555A	CE50	3358	MOV	C16,*SOFT+		08026
10632	555E	0229	FFF4	AI	SOFT,-12		08027
10633	5562	06A0	713C	BL	NRMLIZ	CONVERT MINIMUM TO SINGLE PRECISION	08028
10634	5566	0420	70AA	BLW ^D	FPMPY	CREATE ACTUAL VALUE OF MINIMUM	08029
10635	556A	C939	D9AA	MOV	*SOFT+,MINART1	MOVE MINIMUM TO AUTOSCALE ARGUMENT	08030
10636	556E	C839	D9AC	MOV	*SOFT+,MINART2		08031
10637	5572	06A0	713C	BL	NRMLIZ	CONVERT MAXIMUM TO SINGLE PRECISION	08032
10638	5576	0420	70AA	BLW ^D	FPMPY	CREATE ACTUAL VALUE OF MAXIMUM	08033
10639	557A	C839	D9AE	MOV	*SOFT+,MAXART1	MOVE MAXIMUM TO AUTOSCALE ARGUMENT	08034
10640	557E	C839	D980	MOV	*SOFT+,MAXART2		08035
10641	5582	0420	78E4	BLW ^D	SCALE	GET AUTOSCALE VALUES	08036
10642	5586	0649		DECT	SOFT		08037
10643	5588	C660	D980	MOV	MAXART2,*SOFT	PUSH SCALING MULTIPLIER ONTO SOFTSTACK	08038
10644	558C	0649		DECT	SOFT		08039
10645	558E	C650	D9AE	MOV	MAXART1,*SOFT		08040
10646	5592	0420	70AA	BLW ^D	FPMPY	VEXP * HEXP * SCALING MULTIPLIER	08041
10647	5596	C079		MOV	*SOFT+,R1		08042

KEYINTS

07947

10648	5598	C0B9		MOV	*SOFT+,R2		08043
10649	559A	C800	D9AC	MOV	R0,MINART2	OLD VERTICAL SCALE	08044
10650	559E	04E0	D9AE	CLR	MAXART1	RUNNING SUM IN MAXART1,MAXART2	08045
10651	55A2	04E0	D9B0	CLR	MAXART2		08046
10652	55A6	C160	D95C	MOV	OPWFM,R5	ADDRESS OF OPWFM DATA	08047
10653	55AA	C1A0	DAD2	MOV	WOADD,R6	ADDRESS OF W0 DATA	08048
10654	55AE	C020	D970	MOV	RESOLV,R0	CURRENT WAVEFORM RESOLUTION	08049
10655	55B2	C0F5		MOV	*R5+,R3	LOAD FIRST POINT OF WAVEFORM	08050
10656	55B4	A0E0	D9AC	A	MINART2,R3	ADD OLD VERTICAL OFFSET	08051
10657	55B8	C103		MOV	R3,R4		08052
10658	55BA	08F3		SRA	R3,15	CREATE SIGNED DOUBLE-WORD VALUE	08053
10659	55BC	0814		SRA	R4,1	DIVIDE VALUE BY 2	08054
10660	55BE	1717		JNC	FSTIPNT		08055
10661	55C0	0584		INC	R4	ROUND RESULT	08056
10662	55C2	1715		JNC	FSTIPNT		08057
10663	55C4	0583		INC	R3		08058
10664	55C6	1013		JMP	FSTIPNT		08059
10665							
10666					***** INTEGRATE THE WAVEFORM *****		
10667							
10668	55C8	C8F5		INTGPNT MOV	*R5+,R3	GET NEXT POINT OF WAVEFORM	08060
10669	55CA	A0E0	D9AC	A	MINART2,R3	ADD VERTICAL OFFSET	08061
10670	55CE	C103		MOV	R3,R4		08062
10671	55D0	08F3		SRA	R3,15	CREATE SIGNED DOUBLE-WORD VALUE	08063
10672	55D2	0814		SRA	R4,1	DIVIDE VALUE BY 2	08064
10673	55D4	1705		JNC	*+12	ROUND RESULT	08065
10674	55D6	05A0	D9B0	INC	MAXART2	(NOTE - ROUND IS ADDED TO SUM SO	08066
10675	55DA	1702		JNC	*+6	ADDITION OF SECOND HALF DOESN'T	08067
10676	55DC	05A0	D9AE	INC	MAXART1	RESULT IN ROUND BEING ADDED TWICE.)	08068
10677	55E0	A803	D9AE	A	R3,MAXART1		08069
10678	55E4	A804	D9B0	A	R4,MAXART2	ADD Y(N)/2	08070
10679	55E8	1702		JNC	FSTIPNT		08071
10680	55EA	05A0	D9AE	INC	MAXART1		08072
10681	55EE	0229	FFFA	FSTIPNT AI	SOFT,-6		08073
10682	55F2	CE60	D9AE	MOV	MAXART1,*SOFT+	PUSH VALUE ONTO SOFTSTACK	08074
10683	55F6	CE60	D9B0	MOV	MAXART2,*SOFT+		08075
10684	55FA	CE60	3358	MOV	C16,*SOFT+		08076
10685	55FE	0229	FFFA	AI	SOFT,-6		08077
10686	5602	06A0	713C	BL	NRMLIZ	CONVERT VALUE TO SINGLE PRECISION	08078
10687	5606	0420	70A0	BLW ²	FPMPYZ	CALCULATE INTEGRAL OF POINT	08079
10688	560A	C5B9		MOV	*SOFT+,*R6	MOVE VALUE TO W0	08080
10689	560C	05C9		INCT	SOFT		08081
10690	560E	60A0	D9AA	S	MINART1,*R6+	SUBTRACT NEW VERTICAL OFFSET	08082
10691	5612	A803	D9AE	A	R3,MAXART1	ADD SECOND HALF OF POINT VALUE (SEE NOTE	08083
10692	5616	A804	D9B0	A	R4,MAXART2	ABOVE) TO RUNNING SUM	08084
10693	561A	1702		JNC	*+6		08085
10694	561C	05A0	D9AE	INC	MAXART1		08086
10695	5620	0600		DEC	R0	DECREMENT REMAINING POINT COUNT	08087
10696	5622	15D2		JGT	INTGPNT	CONTINUE IF MORE POINTS	08088
10697	5624	C020	DAD4	MOV	WOHEAD,R0	ADDRESS TO W0'S HEADER	08089
10698	5628	A020	334C	A	VSCA.D,R0	ADDRESS OF VERTICAL UNIT(S)	08090
10699	562C	C420	3438	MOV	SPACES,*R0	UNITS ARE SPACES AFTER INTEGRATION	08091
10700	5630	0200	0001	LI	R0,1	FLAG INDICATING WAVEFORM NUMBER	08092
10701	5634	04C1		CLR	R1	WAVEFORM 0	08093
10702	5636	04C2		CLR	R2		08094
10703	5638	06A0	6962	BL	PSHREG	PUSH WFM #0 ONTO USERSTACK	08095
10704	563C	0720	D972	SETJ	RDTFLAG	UPDATE ENTIRE CALCULATOR READOUT	08096
10705	5640	0380		RTWP			08097

10707	5642	0706			KEYHP_FT SET0 R6	SET FLAG INDICATING LEFT SHIFT	08099
10708	5644	1001			JMP *+4		08100
10709	5646	04C6			KEYHPRGT CLR R6	SET FLAG INDICATING RIGHT SHIFT	08101
10710	5648	8820	D954	333A	C CURSOR,C1	HOW MANY CURSORS ARE ON?	08102
10711	564E	1102			JLT HP0	0	08103
10712	5650	1303			JEQ HP1	1	08104
10713	5652	150E			JGT HP2	2	08105
10714	5654	04C3			HP0 CLR R3	DEFAULT LEFT POINT TO 0	08106
10715	5656	1002			JMP *+6		08107
10716	5658	C0E0	D956		HP1 MOV CURS1,R3	CURSOR #1 IS LEFT POINT	08108
10717	565C	06A0	690E		BL POPREG	POP (X) OFF USERSTACK	08109
10718	5660	1610			JNE HPERR	WFM # IN X IS ERROR	08110
10719	5662	06A0	6D1E		BL FP2LE	CONVERT HORIZONTAL VALUE TO POINT #	08111
10720	5666	C0A0	D94A		MOV FATAL,R2	IS THIS A VALID POINT #?	08112
10721	566A	1318			JEQ HPERR	NO, THIS IS AN ERROR	08113
10722	566C	C083			MOV R3,R2	RETRIEVE FIRST DATA POINT #	08114
10723	566E	1004			JMP HPSHFT		08115
10724	5670	C060	D956		HP2 MOV CURS1,R1	GET CURSOR #1 AS FIRST DATA POINT #	08116
10725	5674	C0A0	D958		MOV CURS2,R2	GET CURSOR #2 AS SECOND DATA POINT #	08117
10726	5678	6081			HPSHFT S R1,R2	DELTA SHIFT VALUE	08118
10727	567A	0742			ABS R2		08119
10728	567C	C186			MOV R6,R5	CHECK SHIFT TYPE (LEFT OR RIGHT)	08120
10729	567E	1101			JLT *+4		08121
10730	5680	0502			NEG R2	IF RIGHT, LAST POINT IS FROM END	08122
10731	5682	0602			DEC R2		08123
10732	5684	1503			JGT *+8	GET LAST POINT NUMBER	08124
10733	5686	1302			JEQ *+6		08125
10734	5688	A0A0	D970		A RESOLV,R2		08126
10735	568C	0A12			SLA R2,1	DISPLACEMENT OF POINT IN WAVEFORM	08127
10736	568E	C060	D95C		MOV OPWFMD,R1	ADDRESS OF OPERATIONAL WAVEFORM	08128
10737	5692	0428	56AA		BLWP ROTATE	ROTATE WAVEFORM	08129
10738	5696	0720	D972		SET0 ROTFLAG	UPDATE ENTIRE READOUT	08130
10739	569A	0380			RTWP		08131
10740							
10741	569C	C020	33D4		HPERR MOV WSTK,R0	PUSH LAST VALUE POPPED BACK ONTO USERSTACK	08132
10742	56A0	06A0	6936		BL PSHSTK		08133
10743	56A4	04E0	D94A		CLR FATAL	SET ERROR FLAG	08134
10744	56A8	0380			RTWP		08135

ROTATE WAVEFORM RIGHT

08136

```

10746 *****
10747 **
10748 ** ROTATE WAVEFORM **
10749 ** **
10750 ** LEVEL 4 ROUTINE **
10751 ** **
10752 ** INPUT - R1 - ADDRESS OF WAVEFORM TO ROTATE **
10753 ** R2 - DISPLACEMENT OF LAST POINT **
10754 **
10755 *****
10756 56AA DBC0 ROTATE WORD WPLVL4 LEVEL 4 ROUTINE 08137
10757 56AC 56AE WORD *+2 08138
10758 56AE C06D 0002 MOV 2(R13),R1 GET ADDRESS OF WAVEFORM 08139
10759 56B2 C0AD 0004 MOV 4(R13),R2 GET DISPLACEMENT OF NEW LAST POINT 08140
10760 56B6 20A0 333C COC C2,R2 IS THIS AN ODD OR EVEN POINT NUMBER 08141
10761 56BA 1611 JNE FSTSHFT IF ODD, THEN FAST SHIFT 08142
10762 56BC 0642 DECT R2 IF EVEN, A PRESHIFT IS NEEDED 08143
10763 56BE 1505 JGT SLWSHFT 08144
10764 56C0 1304 JEQ SLWSHFT 08145
10765 56C2 A0A0 D970 A RESOLV,R2 FIRST POINT MOVED TO LAST POINT 08146
10766 56C6 A0A0 D970 A RESOLV,R2 08147
10767 56CA C101 SLWSHFT MOV R1,R4 ADDRESS OF WAVEFORM 08148
10768 56CC C160 D970 MOV RESOLV,R5 WAVEFORM RESOLUTION 08149
10769 56D0 0605 DEC R5 08150
10770 56D2 C184 MOV R4,R5 08151
10771 56D4 C1F4 MOV *R4+,*R7 SAVE FIRST POINT 08152
10772 56D6 C0B4 MOV *R4+,*R6+ SHIFT RESOLUTION-1 POINTS TO THE LEFT 08153
10773 56D8 0605 DEC R5 08154
10774 56DA 15F0 JGT *-4 SHIFT RESOLUTION-1 WORDS 08155
10775 56DC C587 MOV R7,*R6 MOVE FIRST POINT TO LAST POINT 08156
10776 56DE C0C1 FSTSHFT MOV R1,R3 ADDRESS OF WAVEFORM 08157
10777 56E0 C103 MOV R3,R4 WAVEFORMS BEGINNING ADDRESS 08158
10778 56E2 C160 D970 MOV RESOLV,R5 WAVEFORM RESOLUTION 08159
10779 56E6 0A15 SLA R5,1 RESOLUTION * 2 08160
10780 56E8 A105 A R5,R4 START OF NEXT WAVEFORM'S DATA 08161
10781 56EA A0C2 A R2,R3 ADDRESS OF WAVEFORMS LAST DATA POINT 08162
10782 56EC 05C2 INCT R2 ADDRESS INCREMENT 08163
10783 56EE C183 MOV R3,R6 08164
10784 56F0 A182 A R2,R6 NEXT POINT TO MOVE 08165
10785 56F2 8106 C R6,R4 ADD IS MODULO-RESOLUTION 08166
10786 56F4 1A01 JL *+4 08167
10787 56F6 5185 S R5,R5 SUBTRACT RESOLV*2 TO KEEP ADDRESS IN RANGE 08168
10788 56F8 C220 D970 MOV RESOLV,R8 08169
10789 56FC 0608 DEC R8 MOVE RESOLUTION-1 POINTS 08170
10790 56FE C103 MOV *R3,*R7 SAVE LAST POINT 08171
10791 5700 C4D6 MOV *R6,*R3 MOVE POINT TO ITS CORRECT POSITION 08172
10792 5702 C0C6 MOV R6,R3 THIS POINT IS NOW AVAILABLE TO BE FILLED 08173
10793 5704 A182 A R2,R5 ADD DISPLACEMENT TO GET NEXT POINT 08174
10794 5706 8106 C R6,R4 ADD IS MODULO-RESOLUTION 08175
10795 5708 1A01 JL *+4 08176
10796 570A 6185 S R5,R5 SUBTRACT RESOLV*2 TO KEEP ADDRESS IN RANGE 08177
10797 570C 0608 DEC R8 MORE POINTS TO MOVE? 08178
10798 570E 15F8 JGT MOVPT YES, CONTINUE 08179
10799 5710 C4C7 MOV R7,*R3 MOVE LAST POINT BACK INTO WFM 08180
10800 5712 0380 RTWP 08181

```

```

10802 *****
10803 **
10804 ** HANDLER FOR 'DIFF' KEY **
10805 ** ** **
10806 ** LEVEL 1 ROUTINE **
10807 ** ** **
10808 ** NOTE --- **
10809 ** THE DERIVATIVE OF THE OPERATIONAL WAVEFORM IS CALCULATED **
10810 ** USING THE MEAN SLOPE OF ADJACENT POINTS: **
10811 ** ** **
10812 ** DIF(0) = (Y(1) - Y(0)) / HORIZONTAL INCREMENT **
10813 ** ** **
10814 ** FOR N = 1 TO P/W - 2: **
10815 ** ** **
10816 ** DIF(N) = ((Y(N+1) - Y(N-1)) / 2) / HORIZONTAL INCREMENT **
10817 ** ** **
10818 ** DIF(P/W-1) = (Y(P/W-1) - Y(P/W-2)) / HORIZONTAL INCREMENT **
10819 ** ** **

```

Address	Hex	Hex	Hex	Label	Comment	Address
10820					*****	
10821	5714	06A0	690E	KEYDIFF BL	POPREG POP (X) OFF USERSTACK	08183
10822	5718	1605		JNE	DIFFWFM	08184
10823	571A	06A0	6962	BL	PSHREG (X) MUST BE A WFM	08185
10824	571E	04E0	094A	CLR	FATAL	08186
10825	5722	0380		RTWP		08187
10826	5724	0420	69A8	DIFFWFM BLWP	OPWH2W0 TRANSFER HEADER FROM OPW TO W0	08188
10827	5728	C020	0970	MOV	RESOLV,R0 GET CURRENT WAVEFORM RESOLUTION	08189
10828	572C	0640		DECT	R0	08190
10829	572E	C060	D95C	MOV	OPWFMD,R1 ADDRESS OF OPWFM'S WAVEFORM DATA	08191
10830	5732	C081		MOV	R1,R2	08192
10831	5734	05C2		INCT	R2	08193
10832	5736	C0F2		MOV	*R2+,R3 GET Y(1)	08194
10833	5738	60D1		S	*R1,R3 Y(1) - Y(0)	08195
10834	573A	C103		MOV	R3,R4 R3 IS MIN, R4 IS MAX	08196
10835	573C	C172		DMINMAX MOV	*R2+,R5 GET Y(N+1)	08197
10836	573E	6171		S	*R1+,R5 Y(N+1) - Y(N-1)	08198
10837	5740	0815		SRA	R5,1 (Y(N+1) - Y(N-1)) / 2	08199
10838	5742	1701		JNC	CHKDMIN	08200
10839	5744	0585		INC	R5 ROUND DIVISION BY 2	08201
10840	5746	80C5		CHKDMIN C	R5,R3 IS THIS A NEW MINIMUM?	08202
10841	5748	1501		JGT	CHKDMAX NO, CHECK FOR NEW MAXIMUM	08203
10842	574A	C0C5		MOV	R5,R3 YES, SAVE THIS AS NEW MINIMUM	08204
10843	574C	8185		CHKDMAX C	R5,R4 IS THIS A NEW MAXIMUM?	08205
10844	574E	1101		JLT	*+4 NO, CHECK NEXT POINT	08206
10845	5750	C105		MOV	R5,R4 YES, SAVE THIS AS NEW MAXIMUM	08207
10846	5752	0608		DEC	R0 DECREMENT REMAINING POINT COUNT	08208
10847	5754	15F3		JGT	DMINMAX IF MORE POINTS CONTINUE	08209
10848	5756	1104		JLT	SCLDIFF	08210
10849	5758	0642		DECT	R2 HANDLE ENDPOINT SPECIAL	08211
10850	575A	C152		MOV	*R2,R5 GET Y(M)	08212
10851	575C	6151		S	*R1,R5 Y(M) - Y(M-1)	08213
10852	575E	10F3		JMP	CHKDMIN	08214
10853	5760	C020	D95E	SCLDIFF MOV	OPWFMD,R0 ADDRESS OF OPWFM'S HEADER	08215
10854	5764	C080		MOV	R0,R2	08216
10855	5766	A020	3338	A	VEXP,R0 ADDRESS OF VERTICAL MULTIPLIER	08217
10856	576A	0229	FFFC	AI	SOFT,-4	08218
10857	576E	CE70		MOV	*R0+,*SOFT+ PUSH VERTICAL MULTIPLIER ONTO SOFTSTACK	08219
10858	5770	C650		MOV	*R0,*SOFT	08220
10859	5772	0649		DECT	SOFT	08221
10860	5774	A0A0	3340	A	HEXP,R2 ADDRESS OF HORIZONTAL INCREMENT	08222

KEYDIFF

08182

10861	5778	C072		MOV	*R2+,R1	MOVE HORIZONTAL INCREMENT TO R1,R2	08223
10862	577A	C092		MOV	*R2,R2		08224
10863	577C	0420	7050	BLWP	FPDIV	VEXP / HEXP	08225
10864	5780	C079		MOV	*SOFT+,R1		08226
10865	5782	C099		MOV	*SOFT,R2		08227
10866	5784	0439		CLR	*SOFT	PUSH MINIMUM ONTO SOFTSTACK	08228
10867	5786	0649		DECT	SOFT		08229
10868	5788	C643		MOV	R3,*SOFT		08230
10869	578A	0420	70AA	BLWP	FPMPY	CREATE ACTUAL VALUE OF MINIMUM	08231
10870	578E	C839	D9AA	MOV	*SOFT+,MINART1	MOVE MINIMUM TO AUTOSCALE ARGUMENT	08232
10871	5792	C819	D9AC	MOV	*SOFT,MINART2		08233
10872	5796	04D9		CLR	*SOFT	PUSH MAXIMUM ONTO SOFTSTACK	08234
10873	5798	0649		DECT	SOFT		08235
10874	579A	C644		MOV	R4,*SOFT		08236
10875	579C	0420	70AA	BLWP	FPMPY	CREATE ACTUAL VALUE OF MAXIMUM	08237
10876	57A0	C839	D9AE	MOV	*SOFT+,MAXART1	MOVE MAXIMUM TO AUTOSCALE ARGUMENT	08238
10877	57A4	C839	D980	MOV	*SOFT+,MAXART2		08239
10878	57A8	0420	78E4	BLWP	SCALE	GET AUTOSCALE VALUES	08240
10879	57AC	0649		DECT	SOFT		08241
10880	57AE	C660	D980	MOV	MAXART2,*SOFT	PUSH SCALING MULTIPLIER ONTO SOFTSTACK	08242
10881	57B2	0649		DECT	SOFT		08243
10882	57B4	C660	D9AE	MOV	MAXART1,*SOFT		08244
10883	57B8	0420	70AA	BLWP	FPMPY	(VEXP / HEXP) * SCALING MULTIPLIER	08245
10884	57BC	C079		MOV	*SOFT+,R1		08246
10885	57BE	C039		MOV	*SOFT+,R2		08247
10886	57C0	C020	D970	MOV	RESOLV,R0	GET CURRENT WAVEFORM RESOLUTION	08248
10887	57C4	0600		DEC	R0		08249
10888	57C6	C0E0	D95C	MOV	OPWFMD,R3	GET ADDRESS OF OPWFMD	08250
10889	57CA	C103		MOV	R3,R4		08251
10890	57CC	05C4		INCF	R4		08252
10891	57CE	C160	DAD2	MOV	WOADD,R5	ADDRESS OF W0	08253
10892	57D2	C184		MOV	*R4+,R6	GET Y(1)	08254
10893	57D4	C1F3		MOV	*R3+,R7	GET Y(0)	08255
10894	57D6	6187		S	R7,R5	Y(1) - Y(0)	08256
10895	57D8	0649		DECT	SOFT		08257
10896	57DA	04D9		CLR	*SOFT		08258
10897	57DC	1005		JMP	DIFFOUT		08259
10898	57DE	C134		DIFFPNT MOV	*R4+,R5	GET Y(N+1)	08260
10899	57E0	6187		S	R7,R5	Y(N+1) - Y(N-1)	08261
10900	57E2	C1F3		MOV	*R3+,R7	SAVE NEXT Y(N-1)	08262
10901	57E4	0649		DECT	SOFT		08263
10902	57E6	0719		SETJ	*SOFT	PUSH (Y(N+1) - Y(N-1)) / 2 ONTO SOFTSTACK	08264
10903	57E8	0649		DIFFOUT DECT	SOFT		08265
10904	57EA	C646		MOV	R6,*SOFT		08266
10905	57EC	0420	70A0	BLWP	FPMPYZ	CALCULATE DERIVATIVE OF POINT	08267
10906	57F0	C579		MOV	*SOFT+,*R5	MOVE RESULT TO W0	08268
10907	57F2	05C9		INCF	SOFT		08269
10908	57F4	5060	D9AA	S	MINART1,*R5+	SUBTRACT NEW VERTICAL OFFSET	08270
10909	57F8	0600		DEC	R0	DECREMENT REMAINING POINT COUNT	08271
10910	57FA	15F1		JGT	DIFFPNT	IF MORE POINTS CONTINUE	08272
10911	57FC	1105		JLT	PSHDIFF	HANDLE ENDPOINT SPECIAL	08273
10912	57FE	C193		MOV	*R3,R6	GET Y(N)	08274
10913	5800	6187		S	R7,R5	Y(N) - Y(N-1)	08275
10914	5802	0649		DECT	SOFT		08276
10915	5804	04D9		CLR	*SOFT		08277
10916	5806	10F0		JMP	DIFFOUT		08278
10917	5808	C020	DAD4	PSHDIFF MOV	WOHEAD,R0	ADDRESS TO W0'S HEADER	08279
10918	580C	A020	334C	A	VSCALD,R0	ADDRESS OF VERTICAL UNIT(S)	08280
10919	5810	C420	3430	MOV	SPACES,*R0	UNITS ARE SPACES AFTER INTEGRATION	08281

10920	5814	0200	0001	LI	R0,1	FLAG INDICATING WAVEFORM NUMBER	08202
10921	5818	04C1		CLR	R1	WAVEFORM 0	08203
10922	581A	04C2		CLR	R2		08204
10923	581C	06A0	6962	BL	PSHREG	PUSH WFM #0 ONTO USERSTACK	08205
10924	5820	0720	0972	SET0	ROTFLAG	UPDATE ENTIRE CALCULATOR READOUT	08206
10925	5824	0300		RTWP			08207

10927					*****			
10928					*			
10929					*	HANDLE MIN, MAX, MID, P-P KEYS		
10930					*			
10931					*	LEVEL 1 ROUTINE		
10932					*			
10933	5826	04C1			KEYMAX	CLR R1	R1 FLAGS WHO CALLED	08289
10934	5828	1008				JMP RUNSET		08290
10935	582A	0201	0001		KEYMIN	LI R1,1		08291
10936	582E	1005				JMP RUNSET		08292
10937	5830	0201	0002		KEYMID	LI R1,2		08293
10938	5834	1002				JMP RUNSET		08294
10939	5836	0201	0003		KEYP2P	LI R1,3		08295
10940			583A			RUNSET EQU *		08296
10941	583A	06A0	5A62		RJNALL	BL GETENDS		08297
10942	583E	C2C0				MOV R0,R11	R11 HAS WINDOW WIDTH NOW	08298
10943	5840	C080				MOV R0,R2	R2=COUNTER	08299
10944	5842	C001				MOV R1,R0	R0 FLAGS WHO CALLED NOW	08300
10945	5844	C046				MOV R6,R1		08301
10946	5846	A060	D95C			A OPWFMD,R1	R1=DATA POINTER	08302
10947	584A	C0D1				MOV *R1,R3	R3=MAX	08303
10948	584C	C103				MOV R3,R4	R4=MIN	08304
10949	584E	E820	3348	D972		SJC CLINE15,ROTFLAG		08305
10950	5854	C1F1			BIGLOP	MOV *R1+,R7	R7=DATA	08306
10951	5856	80C7				C R7,R3		08307
10952	5858	1101				JLT **4		08308
10953	585A	C0C7				MOV R7,R3	NEW MAX	08309
10954	585C	81C4				C R4,R7		08310
10955	585E	1161				JLT **4		08311
10956	5860	C107				MOV R7,R4	NEW MIN	08312
10957	5862	0602				DEC R2		08313
10958	5864	16F7				JNE BIGLOP		08314
10959	5866	C228	D95E			MOV OPWFMD,R8		08315
10960	586A	A220	3348			A VOFFAB,R8		08316
10961	586E	A008				A *R8,R3	ADD OFFSET TO MAX AND MIN	08317
10962	5870	A118				A *R8,R4		08318
10963	5872	C800				MOV R8,R0		08319
10964	5874	130A				JEQ MAX		08320
10965	5876	0640				DECT R0		08321
10966	5878	111B				JLT MIN		08322
10967	587A	131C				JEQ MID		08323
10968	587C	60C4				S R4,R3	PEAK-PEAK	08324
10969	587E	0743				ABS R3		08325
10970	5880	0283	8000			CI R3,\$8000	MAXIMUM POSSIBLE NUMBER MUST BE REDUCED	08326
10971	5884	1602				JNE **6		08327
10972	5886	0203	7FFF			LI R3,\$7FFF		08328
10973	588A	C043			MAX	MOV R3,R1		08329
10974	588C	04C2				CLR R2		08330
10975	588E	C120	D95E		FINISH	MOV OPWFMD,R4		08331
10976	5892	A120	3338			A VEXP,R4		08332
10977	5896	0649				DECT SOFT		08333
10978	5898	C174				MOV *R4+,R5		08334
10979	589A	C654				MOV *R4,*SOFT		08335
10980	589C	0649				DECT SOFT		08336
10981	589E	C645				MOV R5,*SOFT		08337
10982	58A0	0420	70AA			BLWP FPMPY	GET NUMERIC RESULT	08338
10983	58A4	C079				MOV *SOFT+,R1	PUSH RESULT	08339
10984	58A6	C089				MOV *SOFT+,R2		08340
10985	58A8	04C0				CLR R0		08341

10986	58AA	06A0	6962		BL PSHREG	08342
10987	58AE	0380			RTWP	08343
10988	58B0	C0C4		MIN	MOV R4,R3	08344
10989	58B2	10EB			JMP MAX	08345
10990			58B4	MID	EQJ *	08346
10991	58B4	0814			SRA R4,1	08347
10992	58B6	0813			SRA R3,1	08348
10993	58B8	A0C4			A R4,R3	08349
10994	58BA	10E7			JMP MAX	08350

KEYMEAN / KEYRMS

00351

```

10996 *****
10997 **
10998 ** KEY HANDLER FOR 'MEAN' AND 'RMS' **
10999 **
11000 ** LEVEL 1 ROUTINE **
11001 **
11002 *****
11003 58BC 0420 6054 KEYMEAN BLWP SUM CALCULATE SUM OF POINTS IN QPWFH 00352
11004 58C0 0706 SETJ R6 FLAG INDICATING 'MEAN' KEY 00353
11005 58C2 1003 JMP MEANRMS 00354
11006 58C4 0420 60E0 KEYRMS BLWP SUMSQR CALCULATE SUM OF SQUARES OF POINTS IN QPWFH 00355
11007 58C8 04C6 CLR R6 FLAG INDICATING 'RMS' KEY 00356
11008 58CA C040 MEANRMS MOV R0,R1 WINDOW WIDTH FROM 'SUM' & 'SUMSQR' 00357
11009 58CC 0601 DEC R1 00358
11010 58CE 1304 JEQ NODIV CURSORS TOGETHER, DON'T DIVIDE 00359
11011 58D0 06A0 757E BL INT2FP FLOATING POINT (M-N) 00360
11012 58D4 0420 7050 BLWP FPDIV SUM DIVIDED BY NUMBER OF POINTS 00361
11013 58D8 C186 NODIV MOV R6,R5 IS THIS KEY 'RMS'? 00362
11014 58DA 1602 JNE **6 00363
11015 58DC 0420 748A BLWP FPSQRT YES, FIND SQUARE ROOT OF VALUE 00364
11016 58E0 04C0 CLR R0 FLAG INDICATING FLOATING POINT NUMBER 00365
11017 58E2 C079 MOV *SOFT+,R1 POP VALUE OFF SOFTSTACK 00366
11018 58E4 C0B9 MOV *SOFT+,R2 00367
11019 58E6 06A0 6962 BL PSHREG PUSH VALUE ONTO USERSTACK 00368
11020 58EA E820 3348 0972 SOC CLINE16,RDTFLAG UPDATE LINE #16 OF READOUT 00369
11021 58F0 0380 RTWP 00370

```

GETMID

08371

```

11023 *
11024 * GET THE 50 PERCENT LEVELS OF A WFM
11025 * LEVEL 2 ROUTINE
11026 *
11027 * OUTPUT- FATAL FLAG FOR FIRST SEARCH IN R1
11028 * 50 PERCENT VALUES ON THE STACK (FP NOTATION)
11029 *
11030 * REGISTER USAGE -
11031 * R6 - DATA AT START OF WINDOW
11032 * R7 - DATA AT END OF WINDOW
11033 * R8 - WINDOW WIDTH OF SEARCH
11034 *
11035 58F2 DB80 GETMID WORD WPLVL2 LEVEL 2 ROUTINE 08372
11036 58F4 58F5 WORD *+2 08373
11037 58F6 C260 0012 MOV 18(R13),SOFT GET CURRENT SOFTSTACK POINTER 08374
11038 58FA 06A0 5A62 BL GETENDS 08375
11039 58FE C200 MOV R0,R8 SAVE WINDOW WIDTH 08376
11040 5900 A1A0 095C A OPWFMD,R6 DATA AT START OF WINDOW 08377
11041 5904 A1E0 095C A OPWFMD,R7 DATA AT END OF WINDOW 08378
11042 5908 05D6 C *R6,*R7 ERROR IF LEVELS EQUAL 08379
11043 590A 1603 JNE *+8 08380
11044 590C 04E0 094A CLR FATAL 08381
11045 5910 0380 RTWP 08382
11046 5912 C084 MOV R4,R2 FIND 50 PERCENT LEVEL 08383
11047 5914 8820 0954 333C C CURSOR,C2 08384
11048 591A 1610 JNE NOFWD5 2 CURSORS NEEDED FOR 2ND SEARCH 08385
11049 591C C007 MOV R7,R8 08386
11050 591E C060 0970 MOV RESOLV,R1 08387
11051 5922 6060 0958 S CURS2,R1 WINDOW IS C2 TO END OF WFM 08388
11052 5926 3820 333A DABE MOV C1,CRJS=LAG 08389
11053 592C 04E0 DAC0 CLR CROSNJM 08390
11054 5930 0420 6A1C BLWP FWDGRJS 08391
11055 5934 C860 094A 0002 MOV FATAL,2(R13) 08392
11056 593A 1002 JMP *+6 08393
11057 593C 0649 NOFWD5 DECT SOFT DECREMENT STACK 08394
11058 593E 0649 DECT SOFT 08395
11059 5940 C048 MOV R8,R1 WINDOW WIDTH OF SEARCH 08396
11060 5942 C007 MOV R7,R0 START SEARCH AT C2 08397
11061 5944 C820 333A DABE MOV C1,CRJS=LAG START SEARCH AT CURRENT POINT 08398
11062 594A 04E0 DAC0 CLR CROSNJM 08399
11063 594E 0720 094A SETJ FATAL 08400
11064 5952 0420 6A28 BLWP BAKGRJS 08401
11065 5956 C849 0012 MOV SOFT,18(R13) 08402
11066 595A 0380 RTWP 08403

```

KEYRISE / KEYFALL

08404

```

11068 *
11069 * HANDLE RISE AND FALL KEYS
11070 *
11071 * LEVEL 1 ROUTINE
11072 *
11073 * REGISTER USAGE -
11074 * R0 - END OF WINDOW (POINT NUMBER)
11075 * R3 - 10 PERCENT VERTICAL VALUE (90 FOR FALL)
11076 * R5 - 90 PERCENT VERTICAL VALUE (10 FOR FALL)
11077 * R6 - START OF WINDOW (POINT NUMBER)
11078 *
11079 * NOTE:
11080 * WINDOW WIDTH OF SEARCH IS FROM CURSOR 1 TO CURSOR 2 (OR
11081 * RIGHT EDGE OF SCREEN IF CURSOR 2 NOT ON) FOR RISE.
11082 * WINDOW IS FROM CURSOR 2 TO RIGHT EDGE OF SCREEN FOR FALL.
11083 *
11084 595C 8820 0954 333C KEYFALL EQU * 08405
11085 595C 8820 0954 333C C CURSOR,C2 CURSOR 2 MUST BE ON FOR FALL 08406
11086 5962 1643 JNE BAD17 08407
11087 5964 0720 DAC4 SETD FALFLG SET FALL FLAG 08408
11088 5968 1802 JMP **6 08409
11089 596A KEYRISE EQU * 08410
11090 596A 04E0 DAC4 CLR FALFLG CALCULATE RISE TIME 08411
11091 596E 0420 58F2 BLWP GETMID 08412
11092 5972 C020 094A MOV FATAL,R0 ERROR IF VERTICAL LEVELS OF CURSORS EQUAL 08413
11093 5976 1339 JEQ BAD17 08414
11094 5978 06A0 5A62 BL GETENDS 08415
11095 597C C007 MOV R7,R0 SAVE THE END OF THE WINDOW OFFSET 08416
11096 597E 0816 SRA R6,1 CHANGE OFFSET TO ACTUAL POINT 08417
11097 5980 0810 SRA R0,1 08418
11098 5982 C2E0 DAC4 MOV FALFLG,R11 CALCULATE FALL TIME OR RISE TIME? 08419
11099 5986 1308 JEQ NOTFAL 08420
11100 5988 C041 MOV R1,R1 IF 2ND 50 PERCENT LEVEL NOT FOUND ... 08421
11101 598A 1352 JEQ BAD16 ...ERROR 08422
11102 598C C103 MOV R3,R4 EXCHANGE 10 AND 90 PERCENT LEVELS FOR FALL 08423
11103 598E C0C5 MOV R5,R3 08424
11104 5990 C144 MOV R4,R5 08425
11105 5992 C180 MOV R0,R5 CURSOR 2 START OF WINDOW FOR FALL 08426
11106 5994 C020 0970 MOV RESOLV,R0 GET WINDOW FOR FALL 08427
11107 5998 0600 DEC R0 08428
11108 599A 05C9 INCT SOFT GET 2ND 50 PERCENT VALUE 08429
11109 599C 05C9 INCT SOFT 08430
11110 599E NOTFAL EQU * 08431
11111 599E C079 MOV *SOFT+,R1 08432
11112 59A0 C0B9 MOV *SOFT+,R2 08433
11113 59A2 06A0 75A6 BL FPTRNC BACK UP TO POINT PRIOR TO 50 PERCENT LEVEL 08434
11114 59A6 C220 095C MOV OPWFMD,R8 08435
11115 59AA C1C0 MOV R0,R7 RESTORE END OF WINDOW POINT 08436
11116 59AC C101 MOV R1,R4 SAVE 50 PERCENT POINT NUMBER 08437
11117 59AE 0A11 SLA R1,1 08438
11118 59B0 61C4 S R4,R7 GET WINDOW WIDTH - 50 PERCENT POINT 08439
11119 59B2 0587 INC R7 TO END OF WINDOW 08440
11120 59B4 A281 A R1,R8 08441
11121 59B6 C008 MOV R8,R0 ADDRESS TO START SEARCH 08442
11122 59B8 C047 MOV R7,R1 WINDOW WIDTH OF SEARCH 08443
11123 59BA C085 MOV R5,R2 FIND 90 PERCENT VALUE (10 FOR FALL) 08444
11124 59BC C820 333A DABE MOV C1,CROSFLAG START SEARCH AT 50 PERCENT POINT 08445
11125 59C2 04E0 DAC0 CLR CROSNJM 08446
11126 59C6 0420 6A1C BLWP FWDCROS 08447

```

KEYRISE / KEYFALL

08404

11127	59CA	05C8			INCT R8	START AT FIRST INTEGRAL POINT PAST 50 PERCENT LEV	08448
11128	59CC	0008			MOV R8,R0		08449
11129	59CE	6106			S R6,R4	WINDOW FROM START OF WINDOW TO 50 PERCENT POI	08450
11130	59D0	05C4			INCT R4		08451
11131	59D2	C044			MOV R4,R1	WINDOW WIDTH OF SEARCH	08452
11132	59D4	C083			MOV R3,R2	FIND 10 PERCENT VALUE (90 FOR FALL)	08453
11133	59D6	C820	333A	DABE	MOV C1,CROSSFLAG		08454
11134	59DC	04E0	DAC0		CLR CROSNJM		08455
11135	59E0	0420	6A28		BLWP BAKCROS		08456
11136	59E4	C020	D94A		MOV FATAL,R0		08457
11137	59E8	1102			JLT *+6		08458
11138	59EA	0460	5A30	BAD17	B BAD16		08459
11139	59EE	C079		GHTIME	MOV *SOFT+,R1		08460
11140	59F0	C0B9			MOV *SOFT+,R2		08461
11141	59F2	0420	6F60		BLWP FPSUB	GET TIME DIFFERENCE	08462
11142	59F6	C0E0	D95E		MOV DPWFMM,R3		08463
11143	59FA	A0E0	334J		A HEXP,R3		08464
11144	59FE	C073			MOV *R3+,R1	MULTIPLY BY HSGL TO GET TRUE	08465
11145	5A00	C093			MOV *R3,R2	HORIZONTAL TIME VALUE	08466
11146	5A02	0420	70AA		BLWP FPMPY		08467
11147	5A06	C079			MOV *SOFT+,R1		08468
11148	5A08	C0B9			MOV *SOFT+,R2		08469
11149	5A0A	0741			ABS R1	GET POSITIVE TIME ONLY	08470
11150	5A0C	04C0			CLR R0		08471
11151	5A0E	E820	3348	D972	SOC CLINE16,RDTFLAG		08472
11152	5A14	06A0	6962		BL PSHREG		08473
11153	5A18	0380			RTWP		08474

```

11155 *
11156 * HANDLE WID KEY
11157 *
11158 * LEVEL 1 ROUTINE
11159 *
11160 5A1A KEYWIDTH EQU * 08476
11161 5A1A 8820 D954 333C C CURSOR,C2 08477
11162 5A20 1607 JNE BAD16 C2 MUST BE DN 08478
11163 5A22 0420 58F2 BLWP GETMID GET 50 PERCENT TIME VALUES 08479
11164 5A26 C041 MOV R1,R1 CHECK FOR FATAL ON 1ST 50 PERCENT SEARCH 08480
11165 5A28 1303 JEQ BAD16 08481
11166 5A2A C020 D94A MOV FATAL,R0 08482
11167 5A2E 1103 JLT *+8 08483
11168 5A30 04E0 D94A BAD15 CLR FATAL 08484
11169 5A34 0380 RTWP 08485
11170 5A36 0460 59EE B GHTIME 08486
    
```

11172				*				
11173				*	HANDLE DEL KEY			
11174				*				
11175				*	LEVEL 1			
11176				*				
11177	5A3A	0420	58F2		KEYDEL.AY	3LWP	GETMID	08488
11178	5A3E	04C1			CLR	R1		08489
11179	5A40	8820	0954	333A	C	CURSOR,21		08490
11180	5A46	1102			JLT	*+6		08491
11181	5A48	C050	0956		MOV	CURS1,R1	GET C1 POSITION	08492
11182	5A4C	C020	094A		MOV	FATAL,R0		08493
11183	5A50	1101			JLT	*+4		08494
11184	5A52	10EE			JMP	BAD16		08495
11185	5A54	0649			DECT	SOFT		08496
11186	5A56	C660	3356		MOV	C15,*SOFT		08497
11187	5A5A	0649			DECT	SOFT		08498
11188	5A5C	C641			MOV	R1,*SOFT	PUSH CURSOR NUMBER ON STACK	08499
11189	5A5E	0460	59EE		B	GHTIME		08500

GETENDS --- GET END POINTS OF WINDOWS FOR FUNCTIONS WITH CURSOR09501

11191				*			
11192				*	ASCERTAIN ENDPOINTS OF WINDOW AND VALUES FOR THEM		
11193				*			
11194				*	LEVEL 5 (BASTARD FORM)		
11195				*			
11196				*	OUTPUT ---		
11197				*	R3 10 PER CENT VALUE MANTISSA		
11198				*	R4 50 PER CENT VALUE MANTISSA		
11199				*	R5 90 PER CENT VALUE MANTISSA		
11200				*	R6 START OF WINDOW (OFFSET)		
11201				*	R7 END OF WINDOW (OFFSET)		
11202				*	R0 WINDOW WIDTH		
11203				*			
11204	5A62	0649			GETENDS DECT SOFT		08502
11205	5A64	C64B			MOV R11,*SOFT		08503
11206	5A66	04C6			CLR R6	DEFAULT IS FULL SCREEN	08504
11207	5A68	C1E0	D970		MOV RESOLV,R7		08505
11208	5A6C	0607			DEC R7		08506
11209	5A6E	C020	D954		MOV CURSOR,R0		08507
11210	5A72	1306			JEQ ENDVALS		08508
11211	5A74	C1A0	D956		MOV CURS1,R6	C1 IS UP	08509
11212	5A78	0600			DEC R0		08510
11213	5A7A	1302			JEQ ENDVALS		08511
11214	5A7C	C1E0	D958		MOV CURS2,R7	C2 IS UP	08512
11215	5A80	C020	D95C		ENDVALS MOV OPWFND,R0		08513
11216	5A84	0A16			SLA R6,1		08514
11217	5A86	0A17			SLA R7,1		08515
11218	5A88	C2C0			MOV R0,R11		08516
11219	5A8A	A006			A R5,R0		08517
11220	5A8C	C010			MOV *R0,R0	DATA OF C1	08518
11221	5A8E	A2C7			A R7,R11		08519
11222	5A90	C31B			MOV *R11,R12	DATA OF C2	08520
11223	5A92	C10C			MOV R12,R4		08521
11224	5A94	C08C			MOV R12,R2		08522
11225	5A96	6100			S R0,R4	DELTA VALUE	08523
11226	5A98	04CB			CLR R11		08524
11227	5A9A	04C8			CLR R8		08525
11228	5A9C	C304			MOV R4,R12		08526
11229	5A9E	1502			JGT *+6		08527
11230	5AA0	050C			NEG R12		08528
11231	5AA2	0588			INC R8	DIV IS UNSIGNED ARITHMATIC	08529
11232	5AA4	3EE0	334C		DIV C10,R11	DELTA / 10	08530
11233	5AA8	0A1C			SLA R12,1		08531
11234	5AAA	880C	334C		C R12,C10		08532
11235	5AAE	1101			JLT *+4		08533
11236	5AB0	0588			INC R11		08534
11237	5AB2	C208			MOV R8,R8		08535
11238	5AB4	1301			JEQ *+4		08536
11239	5AB6	0508			NEG R11	CORRECT SIGN	08537
11240	5AB8	C0C0			MOV R0,R3		08538
11241	5ABA	A0C8			A R11,R3	10 PERCENT VALUE	08539
11242	5ABC	C142			MOV R2,R5		08540
11243	5ABE	6148			S R11,R5	90 PERCENT VALUE	08541
11244	5AC0	0814			SRA R4,1	DELTA / 2	08542
11245	5AC2	1701			JNC *+4		08543
11246	5AC4	1584			INC R4	ROUND UP 50 PERCENT VALUE	08544
11247	5AC6	A100			A R0,R4	50 PERCENT VALUE	08545
11248	5AC8	C007			MOV R7,R0		08546
11249	5ACA	6006			S R5,R0	WINDOW SIZE	08547

GETENDS --- GET END POINTS OF WINDOWS FOR FUNCTIONS WITH CURSOR08501

11250	5ACC	0810	SRA R0,1	08548
11251	5ACE	0500	INC R0	08549
11252	5AD0	C2F9	MOV *SOFT+,R11	08550
11253	5AD2	045B	B *R11	08551

KEYAREA / KEYENERGY

08552

11255						*****		
11256						**		**
11257						**	KEY HANGLER FOR 'AREA' AND 'ENERGY'	**
11258						**		**
11259						**	LEVEL 1 ROUTINE	**
11260						**		**
11261						*****		
11262	5AD4	0420	6054		KEYAREA	BLWP	SUM	CALCULATE SUM OF POINTS IN OPWFH 08553
11263	5AD8	1002				JMP	*+6	08554
11264	5ADA	0420	6DE0		KEYENERGY	BLWP	SUMSQR	CALCULATE SUM OF SQUARES OF POINTS IN OPWFH 08555
11265	5ADE	C0A0	D95E			MOV	OPWFH,R2	ADDRESS OF OPW'S HEADER 08556
11266	5AE2	A0A0	334J			A	HEXP,R2	ADDRESS OF HORIZONTAL INCREMENT 08557
11267	5AE6	C072				MOV	*R2+,R1	08558
11268	5AE8	C092				MOV	*R2,R2	LOAD HORIZONTAL INCREMENT 08559
11269	5AEA	0420	70AA			BLWP	FPMPY	MULTIPLY SUM BY HORIZONTAL INCREMENT 08560
11270	5AEE	04C0				CLR	R0	FLAG INDICATING FLOATING POINT VALUE 08561
11271	5AF0	C079				MOV	*SOFT+,R1	POP VALJE OFF SOFTSTACK 08562
11272	5AF2	C0B9				MOV	*SOFT+,R2	08563
11273	5AF4	C0E0	D954			MOV	CURS0R,R3	ARE CURSORS ON? 08564
11274	5AF8	1306				JEQ	VALOK	NO, VALUE IS OK 08565
11275	5AFA	8820	D956	D958		C	CURS1,CURS2	ARE CURSORS TOGETHER? 08566
11276	5B00	1602				JNE	VALOK	NO, VALUES IS OK 08567
11277	5B02	04C1				CLR	R1	VALUE IS 0 WHEN CURSORS ARE TOGETHER 08568
11278	5B04	04C2				CLR	R2	08569
11279	5B06	06A0	6952		VALOK	BL	PSHREG	PUSH VALUE ONTO USERSTACK 08570
11280	5B0A	E820	3348	D972		SJC	CLINE16,ROTFLLAG	UPDATE LINE #16 OF READOUT 08571
11281	5B10	0380				RTWP		08572

KEYPER

08573

11283

*

11284

*

HANDLE PER KEY

11285

*

11286

*

LEVEL 1 ROUTINE

11287

*

11288

5812

KEYPER EQU *

08574

11289

5812

0420

5842

BLWP GETPER

08575

11290

5816

0380

RTWP

08576

11292				*				
11293				*	HANDLE FREQ KEY			
11294				*				
11295				*	LEVEL 1 ROUTINE			
11296				*				
11297			5818		KEYFREQ EQU *			08578
11298	5818	0420	5842		BLWP GETPER			08579
11299	581C	C0E0	D94A		MOV FATAL,R3			08580
11300	5820	1101			JLT **4			08581
11301	5822	100E			JMP PSHFREQ			08582
11302	5824	06A0	690E		BL POPREG	GET VALUES OFF USER STACK		08583
11303	5828	0649			DECT SOFT			08584
11304	582A	C660	333A		MOV FP1E,*SOFT			08585
11305	582E	0649			DECT SOFT			08586
11306	5830	C660	3404		MOV FP1M,*SOFT			08587
11307	5834	0420	7050		BLWP FPDIV	1/PERIOD		08588
11308	5838	C079			MOV *SOFT+,R1			08589
11309	583A	C0B9			MOV *SOFT+,R2			08590
11310	583C	06A0	6962		BL PSHREG	PUSH ANSWER BACK ONTO STACK		08591
11311			5840		PSHFREQ EQU *			08592
11312	5840	0380			RTWP			08593

11314				*				
11315				*	FIND PERIOD OF OPERATIONAL WFM USING (X) FROM STACK			
11316				*				
11317				*	LEVEL 2 ROUTINE			
11318				*				
11319	5B42		0380		GETPER WORD WPLV.2			08595
11320	5B44		5B46		WORD *+2			08596
11321	5B46	C26D	0012		MOV 18(R13),SOFT			08597
11322	5B4A	06A0	690E		BL POPREG GET (X)			08598
11323	5B4E	1621			JNE PERBAD (X) MUST BE A CONSTANT			08599
11324	5B50	0420	6368		BLWP FP2WF4 GET MANTISSA CORRESPONDING TO (X)			08600
11325	5B54	06A0	5A62		BL GETENDS			08601
11326	5B58	C081			MOV R1,R2 SEARCH FOR (X)			08602
11327	5B5A	C040			MOV R0,R1 FULL WINDOW			08603
11328	5B5C	C006			MOV R6,R0			08604
11329	5B5E	A020	095C		A J>WFMD,R0 START SEARCH AT C1			08605
11330	5B62	C2C0			MOV R0,R11			08606
11331	5B64	C301			MOV R1,R12 SAVE R0,R1 FOR RESTART LATER			08607
11332	5B66	C820	333A	DABE	MOV C1,CROSSFLAG			08608
11333	5B6C	04E0	DAC0		CLR CROSNUM			08609
11334	5B70	0420	6A1C		BLW> FWDCROS GET CROSS #1			08610
11335	5B74	C820	333A	DABE	MOV C1,CROSSFLAG NORMAL SEARCH AGAIN			08611
11336	5B7A	C820	333E	DAC0	MOV C3,CROSNUM FIND THIRD CROSSING THIS TIME			08612
11337	5B80	C00B			MOV R11,R0 RESTART AT OLD SPOT			08613
11338	5B82	C04C			MOV R12,R1			08614
11339	5B84	0420	6A1C		BLWP FWDCROS			08615
11340	5B88	C0E0	094A		MOV FATAL,R3 WERE ALL CROSSINGS FOUND PROPERLY?			08616
11341	5B8C	1302			JEQ PERBAD			08617
11342	5B8E	0460	59EE		B GHTIME			08618
11343	5B92	C0E0	33D4	PERBAD	MOV WSTK,R3 IF (X) IS WFM OR 3 CROSSINGS NOT FOUND ...			08619
11344	5B96	04E0	094A		CLR FATAL			08620
11345	5B9A	C033			MOV *R3+,R0 PUT BACK OLD (X)			08621
11346	5B9C	C073			MOV *R3+,R1			08622
11347	5B9E	C0B3			MOV *R3+,R2			08623
11348	5BA0	06A0	6962		BL PSHREG			08624
11349	5BA4	0380			RTWP			08625

KEYORD

08626

```

11351 *****
11352 **
11353 ** 'ORD' KEY HANDLER **
11354 ** ** **
11355 ** LEVEL 1 ROUTINE **
11356 ** ** **
11357 ** STACK OPERATIONS - **
11358 ** SOFTSTACK - USED BUT NO EFFECT TO CALLER **
11359 ** USERSTACK - POPS 1 PUSHES 1 **
11360 ** ** **
11361 *****
11362 53A6 06A0 690E KEYORD BL POPREG POP ARGUMENT OFF USER STACK 08627
11363 58AA 1645 JNE ERR.JR3+4 IF WFM # THEN ERROR 08628
11364 58AC 0649 DECT SOFT 08629
11365 58AE C642 MOV R2,*SOFT PUSH ARGUMENT ONTO SOFTSTACK 08630
11366 58B0 0649 DECT SOFT 08631
11367 58B2 C641 MOV R1,*SOFT 08632
11368 58B4 C0A0 D95E MOV OPWFMH,R2 ADDRESS OF OPWFH'S HEADER 08633
11369 58B8 A0A0 3340 A HEXP,R2 ADDRESS OF OPWFH'S HORIZONTAL INCREMENT 08634
11370 58BC C072 MOV *R2+,R1 HORIZONTAL INCREMENT 08635
11371 58BE C092 MOV *R2,R2 08636
11372 58C0 0420 7050 BLWP FPDIV CALCULATE CORRESPONDING POINT # 08637
11373 58C4 C079 MOV *SOFT+,R1 POP POINT NUMBER OFF STACK 08638
11374 58C6 C099 MOV *SOFT,R2 08639
11375 58C8 0649 DECT SOFT LEAVE POINT NUMBER ON STACK 08640
11376 58CA 06A0 75A5 BL FPTRNC TRUNCATE POINT NUMBER TO INTEGER 08641
11377 58CE 1901 JND **4 08642
11378 58D0 1030 JMP ERR.ORD OVERFLOW IS ERROR 08643
11379 58D2 112F JLT ERR.JR3 NEGATIVE IS ERROR 08644
11380 58D4 C0E0 D970 MOV RESOLV,R3 08645
11381 58D8 0603 DEC R3 MAXIMUM IF RESOLUTION-1 08646
11382 58DA 80C1 C R1,R3 08647
11383 58DC 152A JGT ERR.JR3 08648
11384 58DE 1104 JLT ORDOK 08649
11385 58E0 04F9 CLR *SOFT+ NO DELTA HORIZONTAL IF LAST POINT 08650
11386 58E2 0409 CLR *SOFT 08651
11387 58E4 0649 DECT SOFT 08652
11388 58E6 1005 JMP ORDL 08653
11389 58E8 C0C1 ORDOK MOV R1,R3 SAVE THIS POINT NUMBER 08654
11390 58EA 06A0 757E BL INT2FP CREATE FLOATING POINT TRUNCATED POINT # 08655
11391 58EE 0420 6F60 BLWP FPSU3 GET DELTA DISTANCE FOR INTERPOLATION 08656
11392 58F2 0A13 ORDL SLA R3,1 08657
11393 58F4 A0E0 D95C A OPWFMD,R3 ADDRESS OF LEFT BOUNDARY POINT 08658
11394 58F8 C133 MOV *R3+,R4 LEFT BOUNDARY POINT 08659
11395 58FA C053 MOV *R3,R1 RIGHT BOUNDARY POINT 08660
11396 58FC 6044 S R4,R1 VERTICAL DISTANCE BETWEEN POINTS 08661
11397 58FE 04C2 CLR R2 ADD ZERO EXPONENT 08662
11398 5C00 0420 70A0 BLWP FPMPYZ MULTIPLY DELTA DISTANCE TO GET ADDON VALUE 08663
11399 5C04 A644 A R4,*SOFT ADD VERTICAL VALUE OF LEFT BOUNDARY POINT 08664
11400 5C06 C0A0 D95E MOV OPWFMH,R2 ADDRESS OF OPWFH'S HEADER 08665
11401 5C0A A0A0 3348 A VOFFAB,R2 ADDRESS OF VERTICAL OFFSET 08666
11402 5C0E A652 A *R2,*SOFT ADD VERTICAL OFFSET 08667
11403 5C10 C0A0 D95E MOV OPWFMH,R2 ADDRESS OF OPWFH'S HEADER 08668
11404 5C14 A0A0 3338 A VEXP,R2 ADDRESS OF VERTICAL MULTIPLIER 08669
11405 5C18 C072 MOV *R2+,R1 VERTICAL MULTIPLIER 08670
11406 5C1A C092 MOV *R2,R2 08671
11407 5C1C 0420 70AA BLWP FPMPY COMPUTE ACTUAL VERTICAL VALUE 08672
11408 5C20 04C0 CLR R0 FLAG INDICATING FLOATING POINT VALUE 08673
11409 5C22 C079 MOV *SOFT+,R1 VERTICAL VALUE 08674

```

KEYORD

08626

11410	5C24	C0B9			MOV	*SOFT+,R2		08675
11411	5C26	06A0	6962		BL	PSHREG	PUSH VALUE ONTO USER STACK	08676
11412	5C2A	E828	3348	D972	SOC	CLINE15,ROTFLAG	UPDATE LINE #16 OF READOUT	08677
11413	5C30	0380			RTWP			08678
11414	5C32	0229	0004	ERR.JRD	AI	SOFT,4	POP ARGUMENT OFF SOFTSTACK	08679
11415	5C36	C020	33D4		MOV	WSTK,RJ		08680
11416	5C3A	06A0	6936		BL	PSHSTK	PUSH ARGUMENT BACK ONTO USER STACK	08681
11417	5C3E	04E0	D94A		CLR	FATAL	SET ERROR FLAG	08682
11418	5C42	0380			RTWP			08683

KEY2ORD

08684

```

11420 *****
11421 **
11422 ** '>ORD' KEY HANDLER **
11423 ** ** **
11424 ** LEVEL 1 ROUTINE **
11425 ** ** **
11426 ** STACK OPERATIONS - **
11427 ** SOFTSTACK - USED BUT NO EFFECT TO CALLER **
11428 ** USERSTACK - POPS 1 **
11429 ** ** **
11430 *****
11431 5C44 06A0 690E KEY2ORD BL POPREG POP HORIZONTAL LOCATION OFF USERSTACK 08685
11432 5C48 C142 MOV R2,R5 08686
11433 5C4A C101 MOV R1,R4 SAVE (X) IN CASE OF ERROR 08687
11434 5C4C C0C8 MOV R0,R3 08688
11435 5C4E 1611 JNE ERRSTEP IF X IS WFM # THEN ERROR 08689
11436 5C50 0649 DECT SOFT 08690
11437 5C52 C642 MOV R2,*SOFT PUSH (X) ONTO SOFTSTACK 08691
11438 5C54 0649 DECT SOFT 08692
11439 5C56 C641 MOV R1,*SOFT 08693
11440 5C58 C0A0 095E MOV OPWFMH,R2 ADDRESS OF OPWFM'S HEADER 08694
11441 5C5C A0A0 3340 A HEXP,R2 ADDRESS OF OPWFM'S HORIZONTAL INCREMENT 08695
11442 5C60 C072 MOV *R2+,R1 HORIZONTAL INCREMENT 08696
11443 5C62 C092 MOV *R2,R2 08697
11444 5C64 0420 7050 BLWP FPDIV CALCULATE CORRESPONDING POINT # 08698
11445 5C68 C079 MOV *SOFT+,R1 POP POINT NUMBER OFF SOFTSTACK 08699
11446 5C6A C0B9 MOV *SOFT+,R2 08700
11447 5C6C 06A0 75A0 BL FP2INT CONVERT FLOATING POINT # TO NEAREST INTEGER 08701
11448 5C70 1902 JND *+6 OVERFLOW IS ERROR 08702
11449 5C72 0460 50B8 ERRSTEP B ERR2ORD+4 08703
11450 5C76 11FD JLT ERRSTEP NEGATIVE IS ERROR 08704
11451 5C78 8801 0970 C R1,RESJLV 08705
11452 5C7C 15FA JGT ERRSTEP RESOLUTION-1 IS UPPER LIMIT 08706
11453 5C7E 13F9 JEQ ERRSTEP 08707
11454 5C80 C181 MOV R1,R6 SAVE THIS POINT # 08708
11455 5C82 06A0 690E BL POPREG POP (Y) OFF USERSTACK 08709
11456 5C86 1302 JEQ *+6 08710
11457 5C88 0460 50B4 B ERR2ORD IF (Y) IS WFM # THEN ERROR 08711
11458 5C8C C003 MOV R3,R3 RESTORE (X) 08712
11459 5C8E C0C1 MOV R1,R3 08713
11460 5C90 C044 MOV R4,R1 08714
11461 5C92 C102 MOV R2,R4 08715
11462 5C94 C085 MOV R5,R2 08716
11463 5C96 06A0 6962 BL PSHREG 08717
11464 5C9A C1C3 MOV R3,R7 SAVE >ORD VALUE 08718
11465 5C9C C204 MOV R4,R8 08719
11466 5C9E C0E0 095C MOV OPWFMD,R3 ADDRESS OF OPWFM'S DATA 08720
11467 5CA2 C020 0970 MOV RESOLV,R0 CURRENT RESOLUTION 08721
11468 5CA6 C113 MOV *R3,R4 INITIALIZE MAXIMUM 08722
11469 5CA8 C173 MOV *R3+,R5 INITIALIZE MINIMUM 08723
11470 5CAA 0600 DEC R0 08724
11471 5CAC 8113 OMINMAX C *R3,R4 FIND OPWFM'S MINIMUM AND MAXIMUM 08725
11472 5CAE 1506 JGT NEWOMAX 08726
11473 5CB0 8153 C *R3,R5 08727
11474 5CB2 1105 JLT NEWOMIN 08728
11475 5CB4 05C3 NXTOPNT INCT R3 POINT TO NEXT POINT 08729
11476 5CB6 0600 DEC R0 CHECK OFF POINT IN WAVEFORM 08730
11477 5CB8 15F9 JGT OMINMAX 08731
11478 5CBA 1004 JMP CHKOPNT 08732

```

11479	5CBC	C113		NEWMAX	MOV	*R3,R4	SAVE NEW MAXIMUM	08733
11480	5CBE	10FA			JMP	NXTOPNT		08734
11481	5CC0	C153		NEWMIN	MOV	*R3,R5	SAVE NEW MINIMUM	08735
11482	5CC2	10F8			JMP	NXTOPNT		08736
11483	5CC4	C2A0	D95E	CHKOPNT	MOV	OPWFMH,R10	ADDRESS OF OPWFH'S HEADER	08737
11484	5CC8	A2A0	3348		A	VOFFAB,R10	ADDRESS OF OPWFH'S VERTICAL OFFSET	08738
11485	5CCC	C29A			MOV	*R10,R10	VERTICAL OFFSET	08739
11486	5CCE	A10A			A	R10,R4	ADJUST MAXIMUM	08740
11487	5CD0	A14A			A	R10,R5	ADJUST MINIMUM	08741
11488	5CD2	C0A0	D95E		MOV	OPWFMH,R2	ADDRESS OF OPWFH'S HEADER	08742
11489	5CD6	A0A0	3338		A	VEXP,R2	ADDRESS OF OPWFH'S VERTICAL INCREMENT	08743
11490	5CDA	C072			MOV	*R2+,R1	VERTICAL INCREMENT	08744
11491	5CDC	C092			MOV	*R2,R2		08745
11492	5CDE	0649			DECT	SOFT		08746
11493	5CE0	0409			CLR	*SOFT	PUSH MINIMUM POINT ONTO SOFTSTACK	08747
11494	5CE2	0649			DECT	SOFT		08748
11495	5CE4	C645			MOV	R5,*SOFT		08749
11496	5CE6	0420	70AA		BLWP	FPMPY	CALCULATE ACTUAL VALUE OF MINIMUM	08750
11497	5CEA	0649			DECT	SOFT		08751
11498	5CEC	0409			CLR	*SOFT	PUSH MAXIMUM POINT ONTO SOFTSTACK	08752
11499	5CEE	0649			DECT	SOFT		08753
11500	5CF0	C644			MOV	R4,*SOFT		08754
11501	5CF2	0420	70AA		BLWP	FPMPY	CALCULATE ACTUAL VALUE OF MAXIMUM	08755
11502	5CF6	C047			MOV	R7,R1		08756
11503	5CF8	C088			MOV	R8,R2	RESTORE >ORD VALUE	08757
11504	5CFA	0420	7524	CMPOMAX	BLWP	FPCMPR	CHECK IF >ORD VALUE IS NEW MAXIMUM	08758
11505	5CFE	1505			JGT	ISOMAX		08759
11506	5D00	C839	D9AE		MOV	*SOFT+,MAXART1	IF NOT, USE CURRENT MAXIMUM	08760
11507	5D04	C839	D9B0		MOV	*SOFT+,MAXART2		08761
11508	5D08	1005			JMP	CMPOMIN		08762
11509	5D0A	C801	D9AE	ISOMAX	MOV	R1,MAXART1	IF SO, USE >ORD VALUE	08763
11510	5D0E	C802	D9B0		MOV	R2,MAXART2		08764
11511	5D12	0229	0004		AI	SOFT,4	POP OLD MAXIMUM OFF SOFTSTACK	08765
11512	5D16	0420	7524	CMPOMIN	BLWP	FPCMPR	CHECK IF >ORD VALUE IS NEW MINIMUM	08766
11513	5D1A	1105			JLT	ISOMIN		08767
11514	5D1C	C839	D9AA		MOV	*SOFT+,MINART1	IF NOT, USE CURRENT MINIMUM	08768
11515	5D20	C839	D9AC		MOV	*SOFT+,MINART2		08769
11516	5D24	1005			JMP	AUTOJRD		08770
11517	5D26	C801	D9AA	ISOMIN	MOV	R1,MINART1	IF SO, USE >ORD VALUE	08771
11518	5D2A	C802	D9AC		MOV	R2,MINART2		08772
11519	5D2E	0229	0004		AI	SOFT,4	POP OLD MINIMUM OFF SOFTSTACK	08773
11520	5D32	C320	D95E	AJTJRD	MOV	OPWFMH,R12	OPWFH'S HEADER ADDRESS	08774
11521	5D36	A320	3338		A	VEXP,R12	OPWFH'S VERTICAL INCREMENT ADDRESS	08775
11522	5D3A	C2FC			MOV	*R12+,R11	VERTICAL INCREMENT	08776
11523	5D3C	C31C			MOV	*R12,R12		08777
11524	5D3E	0420	78E4		BLWP	SCALE	AUTOSCALE RESULTING WAVEFORM TO TEK 1-2-5	08778
11525	5D42	C120	D95E		MOV	OPWFMH,R4	OPWFH'S HEADER ADDRESS	08779
11526	5D46	C160	DA04		MOV	WOHEAD,R5	WO'S HEADER ADDRESS	08780
11527	5D4A	A120	334J		A	HEXP,R4	OPWFH'S HORIZONTAL INCREMENT ADDRESS	08781
11528	5D4E	A160	3340		A	HEXP,R5	WO'S HORIZONTAL INCREMENT ADDRESS	08782
11529	5D52	C074			MOV	*R4+,*R5+		08783
11530	5D54	C554			MOV	*R4,*R5	TRANSFER HORIZONTAL INCREMENT, OPWFH TO WO	08784
11531	5D56	0400			CLR	R0	POINT COUNTER STARTS AT 0	08785
11532	5D58	C120	D95C		MOV	OPWFHD,R4	OPWFH'S DATA ADDRESS	08786
11533	5D5C	C160	DA02		MOV	WOADD,R5	WO'S DATA ADDRESS	08787
11534	5D60	C048		ORDWFH	MOV	R11,R1	RESTORE ORIGINAL VERTICAL INCREMENT	08788
11535	5D62	C08C			MOV	R12,R2		08789
11536	5D64	8180			C	R0,R5	IS THIS THE >ORD POINT?	08790
11537	5D66	1606			JNE	NOTOPNT	NO, CONTINUE SCALING ORIGINAL POINTS	08791

11538	5D68	0649		DECT	SOFT		08792	
11539	5D6A	C648		MOV	R8,*SOFT	YES, SUBSTITUTE >ORD VALUE	08793	
11540	5D6C	0649		DECT	SOFT		08794	
11541	5D6E	C647		MOV	R7,*SOFT		08795	
11542	5D70	05C4		INCT	R4	SKIP ORIGINAL POINT	08796	
11543	5D72	1007		JMP	ISOPNT		08797	
11544	5D74	0649		NOTOPNT	DECT	SOFT	08798	
11545	5D76	04D9		CLR	*SOFT	ZERO EXPONENT	08799	
11546	5D78	0649		DECT	SOFT		08800	
11547	5D7A	C674		MOV	*R4+,*SOFT	PUSH NEXT MANTISSA ONTO SOFTSTACK	08801	
11548	5D7C	A64A		A	R10,*SOFT	ADD VERTICAL OFFSET	08802	
11549	5D7E	0420	70AA	BLWP	FPMPY	CALCULATE ACTUAL VALUE OF POINT	08803	
11550	5D82	C060	D9AE	ISOPNT	MOV	MAXART1,R1	08804	
11551	5D86	C0A0	D980		MOV	MAXART2,R2	LOAD SCALING VALUE	08805
11552	5D8A	0420	70A0	BLWP	FPMPYZ	SCALE POINT ON SCREEN	08806	
11553	5D8E	0579		MOV	*SOFT+,*R5	MOVE NEW POINT TO HQ	08807	
11554	5D90	05C9		INCT	SOFT		08808	
11555	5D92	6D60	D9AA	S	MINART1,*R5+	SUBTRACT NEW VERTICAL OFFSET	08809	
11556	5D96	0580		INC	R0	INCREMENT TO NEXT POINT	08810	
11557	5D98	8800	D970	C	R0,RESOLV	AUTOSCALE ALL POINTS	08811	
11558	5D9C	11E1		JLT	ORDWFM		08812	
11559	5D9E	0720	D972	SETO	ROTFLAG	UPDATE ENTIRE READOUT	08813	
11560	5DA2	0700		SETO	R0	FLAG INDICATING WFM #	08814	
11561	5DA4	04C1		CLR	R1	WFM #0	08815	
11562	5DA6	06A0	6962	BL	PSHREG	PUSH 0 WFM	08816	
11563	5DAA	0420	6C98	BLWP	OPWCHG	MAKE 0 NEW OPWFM	08817	
11564	5DAE	06A0	690E	BL	POPREG	POP 0 WFM	08818	
11565	5DB2	0380		RTWP			08819	
11566	5DB4	06A0	6962	ERR2ORD	BL	PSHREG	PUSH (Y) BACK ONTO USERSTACK	08820
11567	5DB8	C085		MOV	R5,R2		08821	
11568	5DBA	C044		MOV	R4,R1		08822	
11569	5DBC	C083		MOV	R3,R0		08823	
11570	5DBE	J6A0	6952	BL	PSHREG	PUSH (X) BACK ONTO USERSTACK	08824	
11571	5DC2	04E0	D94A	CLR	FATAL	SET ERROR FLAG	08825	
11572	5DC6	0380		RTWP			08826	

KEYCRS1 / KEYCRS2.1 / KEYOFF

08827

11574					*				
11575					*	HANDLE 'CRS1'			
11576					*				
11577					*	LEVEL 1			
11578					*				
11579	5DC8	C820	333A	D954	KEYCRS1	MOV C1,CURSOR			08828
11580	5DCE	1003				JMP ++8			08829
11581	5DD0	C820	333C	D954	KEYCRS2.1	MOV C2,CURSOR			08830
11582	5DD6	8820	D958	D956		C CURS2,CURS1			08831
11583	5DDC	1503				JGT ++8			08832
11584	5DDE	C820	D956	D958		MOV CURS1,CURS2			08833
11585	5DE4	E820	3340	D972		SJC CLINE15,RDTFLAG			08834
11586	5DEA	0380				RTWP			08835
11587	5DEC	04E0	D954		KEYOFF	CLR CURSOR			08836
11588	5DF0	E820	3340	D972		SJC CLINE15,RDTFLAG			08837
11589	5DF6	0380				RTWP			08838

```

11591 *
11592 * HANDLE VCRD AND HCRD KEYS
11593 *
11594 * LEVEL 1 ROUTINE
11595 *
11596 * KEYVCRD --- VERT VALUE
11597 * KEYHCRD --- HORZ VALUE
11598 *
11599 * INPUT CURSOR
11600 * OUTPUT ON USER STACK
11601 *
11602 * INPUT NO REG
11603 * OUTPUT ON USER STACK
11604 * DESTROYS R0,R1,R2,R3,R4,R5
11605 *
11606 5DF8 C020 D954 KEYHCRD MOV CURSOR,R0 CURSOR MUST BE UP TO GET HORIZONTAL TIME 08840
11607 5DFC 1309 JEQ NOPHER 08841
11608 5DFE E820 3348 D972 SOC CLINE16,RDTFLAG 08842
11609 5E04 0420 68E0 BLWP HCRPOS 08843
11610 5E08 04C0 PSHIT2 CLR R0 08844
11611 5E0A 06A0 6952 BL PSHREG 08845
11612 5E0E 0380 RTWP 08846
11613 NJP4ER EQU * 08847
11614 5E10 04E0 D94A CLR FATAL SET ERROR FOR RESTRICTION VIOLATION 08848
11615 5E14 0380 RTWP 08849
11616 5E16 C020 D954 KEYVCRD MOV CURSOR,R0 CURSOR MUST BE UP TO GET VERTICAL VALUE 08850
11617 5E1A 13FA JEQ NOPHER 08851
11618 5E1C E820 3348 D972 SOC CLINE15,RDTFLAG 08852
11619 5E22 0420 6880 BLWP VCRPOS 08853
11620 5E26 10F0 JMP PSHIT2 08854
11621 5E28 0420 67E6 BLWP ZROREF 08855
11622 5E2C 10E0 JMP PSHIT2 08856
    
```

```

11524 *
11525 * HANDLE >VC KEY
11526 *
11527 * LEVEL 1 ROUTINE
11528 *
11529 * LOOK FOR FIRST CROSSING OF VALUE AND SET VERTICAL
11530 * CURSOR POINT FOLLOWING THAT CROSSOVER (TO AVOID
11531 * TRAP IN SUCCESSIVE >VC SEARCHES)
11532 * IF POINT FOLLOWING CURSOR IS EQUAL TO SEARCH POINT,
11533 * IGNORE ALL DATA POINTS UP TILL END OF PLATEAU.
11534 * BEGIN NORMAL CURSOR SEARCH AFTER THAT PLATEAU.
11535 * IF NO CURSORS ARE SET, ERROR RETURN WITH NO CURSOR CHANGE
11536 * IF ONE CURSOR ON, SET CURSOR 1 TO VALUE
11537 * IF TWO CURSORS SET, SET CURSOR 2 TO DELTA VALUE
11538 * CURSOR ONE BEGINS SEARCH FROM ITS POINT TO RIGHT, THEN
11539 * WRAPS AROUND TO LEFT OF SCREEN TO CONTINUE SEARCH.
11540 * CURSOR TWO ALSO BEGINS FROM ITS POINT TO RIGHT, BUT
11541 * WRAPS AROUND TO CURSOR ONE POSITION TO CONTINUE
11542 * SEARCH.
11543 * IF VALUE NOT FOUND WITHIN THE APPROPRIATE CURSORS RANGE,
11544 * AN ERROR RETURN IS MADE WITH NO CURSOR MOVES.
11545 *
11546 * REGISTER USAGE ---
11547 * R0 = SEARCH START ADDRESS
11548 * R1 = WINDOW WIDTH
11549 * R2 = VALUE TO FIND
11550 *
11551 * KEY2VCRD EQU *
11552 5E2E 0720 D972 SETO RDTFLAG 08858
11553 5E32 06A0 5A62 BL GETENDS 08859
11554 5E36 C107 MOV R7,R4 SAVE R7 FOR FUTURE USE 08861
11555 5E38 06A0 690E BL POPREG GET SEARCH VALUE 08862
11556 5E3C 06A0 6952 BL >SHREG LEAVE VALUE ON USERSTACK 08863
11557 5E40 C000 MOV R0,R0 08864
11558 5E42 1651 JNE NOGOOD6 MUST BE CONSTANT 08865
11559 5E44 0420 6B58 BLW> FP2WFM MANTISSA VALUE TO FIND 08866
11560 5E48 C0A0 D94C MOV WARNING,R2 IS CONVERSION SUCCESSFUL? 08867
11561 5E4C 134C JEQ NOGOOD6 NO, THEN ERROR 08868
11562 5E4E C081 MOV R1,R2 08869
11563 5E50 C006 MOV R5,R0 START SEARCH AT C1 08870
11564 5E52 A020 D95C A OPWFMD,R0 ABSOLUTE ADDRESS TO START SEARCH 08871
11565 5E56 C060 D970 MOV RESOLV,R1 08872
11566 5E5A C220 D954 MOV CURSOR,R8 08873
11567 5E5E 1343 JEQ NOGOOD6 MUST HAVE CURSOR DISPLAYED 08874
11568 5E60 0608 DEC R8 08875
11569 5E62 130C JEQ VSRCH 08876
11570 *
11571 * 2 CURSORS ARE ON
11572 *
11573 5E64 A090 A *R0,R2 SEARCH DELTA VALUE 08877
11574 5E66 C220 D95E MOV OPWFMD,R8 08878
11575 5E6A A220 3348 A VOFFAB,R8 ADD IN VERTICAL OFFSET 08879
11576 5E6E A098 A *R8,R2 08880
11577 5E70 C004 MOV R4,R0 08881
11578 5E72 A020 D95C A OPWFMD,R0 ABSOLUTE ADDRESS TO START SEARCH 08882
11579 5E76 5050 D958 S CURS2,R1 08883
11580 5E7A 1002 JMP *+6 08884
11581 5E7C 5E7C VSRCH EQU * 08885
11582 5E7C 6060 D956 S CURS1,R1 WINDOW WIDTH 08886
    
```

11683	5E80	04E0	DABE	CLR CROSFLAG		08887
11684	5E84	04E0	DAC0	CLR CROSNUM		08888
11685	5E88	0420	6A1C	BLWP FWDGROS	GET CROSSPOINT	08889
11686	5E8C	C060	D94A	MOV FATAL,R1	WAS A HIT MADE?	08890
11687	5E90	161C		JNE HITLVL		08891
11688	5E92	05C9		INGT SOFT	NO --- CLEAR -1 OFF STACK	08892
11689	5E94	C020	D95C	MOV OPWFMD,R0		08893
11690	5E98	C120	D956	MOV CURS1,R4	START SEARCH AGAIN AT LEFT OF WIDER WINDOW	08894
11691	5E9C	0A14		SLA R4,1		08895
11692	5E9E	C060	D970	MOV RESOLV,R1		08896
11693	5EA2	C0E0	D954	MOV CURSOR,R3		08897
11694	5EA6	0603		DEC R3		08898
11695	5EA8	1303		JEQ C1UP		08899
11696	5EAA	A004		A R4,R0	2 CURSORS UP --- RESTART AT C1	08900
11697	5EAC	5068	D956	S CURS1,R1		08901
11698		5E80		C1UP EQU *		08902
11699	5EB0	0720	D94A	SETJ FATAL		08903
11700	5EB4	04E0	DABE	CLR CROSFLAG		08904
11701	5EB8	04E0	DAC0	CLR CROSNUM		08905
11702	5EBC	0420	6A12	BLWP CNTGROS	GET CROSSING AGAIN	08906
11703	5EC0	C060	D94A	MOV FATAL,R1	DID WE GET A STRIKE THIS TIME?	08907
11704	5EC4	1602		JNE HITLVL		08908
11705	5EC6	05C9		INGT SOFT		08909
11706	5EC8	100E		JMP NOGOOD5		08910
11707			5ECA	HITLVL EQU *		08911
11708	5ECA	C050	D95C	MOV OPWFMD,R1		08912
11709	5ECE	6081		S R1,R0	RETURNED ABSOLUTE VALUE --- CONVERT TO CURSOR COO	08913
11710	5ED0	0810		SRA R0,1		08914
11711	5ED2	C060	D954	MOV CURSOR,R1		08915
11712	5ED6	0601		DEC R1		08916
11713	5ED8	1603		JNE C2UP		08917
11714	5EDA	C800	D956	MOV R0,CURS1		08918
11715	5EDE	0380		RTWP		08919
11716		5EE0		C2UP EQU *		08920
11717	5EE0	C800	D958	MOV R0,CURS2		08921
11718	5EE4	0380		RTWP		08922
11719	5EE6	04E0	D94A	NOGOOD6 CLR FATAL		08923
11720	5EEA	0380		RTWP		08924

11722				*					
11723				*	HANDLE >HC KEY				
11724				*					
11725				*	LEVEL 1 ROUTINE				
11726				*					
11727				*	IF NO CURSORS ON, NOP				
11728				*	IF 1 CURSOR ON, MOV C1 TO (X)				
11729				*	IF 2 CURSORS ON, MOV DELTA C TO (X)				
11730				*					
11731				*	HSCL = (R-1) * HEXP / 10				
11732				*					
11733				*	PNT = (REQUEST * (R-1)) / (HSCL * 10)				
11734				*					
11735				*	THEREFORE ---				
11736				*	PNT = REQUEST / HEXP				
11737				*					
11738	5EEC	C020	D954		KEY2HCRD MOV CURSOR,R0				08926
11739	5EF0	1603			JNE **8				08927
11740	5EF2	04E0	D94A		CLR FATAL				08928
11741	5EF6	0380			RTWP		NOP IF NO CURSORS UP		08929
11742	5EF8	06A0	690E		BL POPREG				08930
11743	5EFC	06A0	6962		BL PSHREG				08931
11744	5F00	C000			MOV R0,R0		ERROR IF X IS WFM		08932
11745	5F02	1303			JEQ HCRSPOP				08933
11746	5F04	04E0	D94A		NOHCRS CLR FATAL				08934
11747	5F08	0380			RTWP				08935
11748			5F0A		HCRSPOP EQU *				08936
11749	5F0A	8820	0978	3338	C XYWFM,C0		ERROR IF WFM IN VS MODE		08937
11750	5F10	16F9			JNE NOHCRS				08938
11751	5F12	C041			MOV R1,R1				08939
11752	5F14	11F7			JLT NOHCRS		NEG # IS ERROR		08940
11753	5F16	E820	3340	D972	SDC CLINE15,RDTFLAG				08941
11754	5F1C	E820	3348	D972	SDC CLINE15,RDTFLAG				08942
11755	5F22	C020	D95E		MOV OPWFMH,R0		PUSH REQUEST		08943
11756	5F26	A020	3340		A HEXP,R0				08944
11757	5F2A	0649			DECT SOFT				08945
11758	5F2C	C642			MOV R2,*SOFT				08946
11759	5F2E	0649			DECT SOFT				08947
11760	5F30	C641			MOV R1,*SOFT				08948
11761	5F32	C070			MOV *R0+,R1				08949
11762	5F34	C090			MOV *R0,R2				08950
11763	5F36	0420	7050		BLWP FPDIV		REQUEST / 4EXP		08951
11764	5F3A	C079			MOV *SOFT+,R1				08952
11765	5F3C	C039			MOV *SOFT+,R2				08953
11766	5F3E	06A0	75A0		BL FP2INT		PNT #		08954
11767	5F42	C120	D970		MOV RESOLV,R4				08955
11768	5F46	0604			DEC R4				08956
11769	5F48	8044			C R4,R1				08957
11770	5F4A	1403			JHE **8				08958
11771	5F4C	04E0	D94C		CLR WARNING		NUMBER PAST RIGHT OF SCREEN		08959
11772	5F50	C844			MOV R4,R1				08960
11773	5F52	C020	D954		MOV CURSOR,RJ				08961
11774	5F56	0600			DEC R0				08962
11775	5F58	160A			JNE HCRS2ON				08963
11776	5F5A	C801	D956		MOV R1,CURS1				08964
11777	5F5E	8820	D956	D958	C CURS1,CURS2				08965
11778	5F64	1103			JLT **8				08966
11779	5F66	C820	D956	D958	MOV CURS1,CURS2		SLIDE CURSOR-2 OVER		08967
11780	5F6C	0380			RTWP				08968

11781	5F6E	A050	D956	HCRS20N A CURS1,R1	2 CURSORS ON ==> SET DELTA C2	08969
11782	5F72	0101		C R1,R4		08970
11783	5F74	1503		JGT C2T00BIG		08971
11784	5F76	C801	D958	MOV R1,CURS2		08972
11785	5F7A	0380		RTWP		08973
11786			5F7C	C2T00BIG EQU *		08974
11787	5F7C	C804	D958	MOV R4,CURS2	C2 <= RESOLV	08975
11788	5F80	04E0	D94C	CLR WARNING		08976
11789	5F84	0380		RTWP		08977

KEYCRS1LFT / KEYCRS1RGT / KEYCRS2LFT / KEYCRS2RGT --- CURSOR 08978

```

11791 *****
11792 **
11793 ** CURSOR SLEW KEYS- 'CRS1<', 'CRS1>'
11794 ** 'CRS2<', 'CRS2>'
11795 **
11796 ** LEVEL 1 ROUTINE
11797 **
11798 ** INPUT- NONE
11799 ** OUTPUT- NONE
11800 **
11801 ** STACK OPERATIONS-
11802 ** NONE
11803 **
11804 ** NOTE ---
11805 ** THE CURSOR SLEW KEYS PROVIDE THE USER WITH CONTINUOUS
11806 ** MOTION CURSOR CONTROL. FOUR BUTTONS CONTROL THE MOVEMENT
11807 ** OF TWO CURSORS. EACH CURSOR USES ONE BUTTON FOR LEFT TO
11808 ** RIGHT MOTION AND ONE BUTTON FOR RIGHT TO LEFT MOTION. THE
11809 ** CURSORS HAVE CERTAIN RANGE LIMITATIONS IN WHICH THEY ARE
11810 ** ALLOWED TO MOVE. CURSOR #1 IS LIMITED BY THE LEFT AND
11811 ** RIGHT EDGES OF THE CRT SCREEN. CURSOR #2 IS LIMITED BY
11812 ** CURSOR #1 ON THE LEFT AND EDGE OF THE CRT SCREEN ON THE
11813 ** RIGHT. CURSOR #1 IS MORE POWERFUL THAN CURSOR #2 IN THAT
11814 ** CURSOR #1 CAN PUSH CURSOR #2 RIGHT BUT CURSOR #2 CANNOT
11815 ** PUSH CURSOR #1 LEFT.
11816 ** THE SPEED OF THE CURSORS IS PROPORTIONAL TO THE CRT
11817 ** GRATICULE NOT TO THE RESOLUTION OF THE WAVEFORMS. THERE IS
11818 ** AN ACCELERATION OF THE CURSORS INITIALLY AFTER PUSHING THE
11819 ** BUTTON. THIS INCREASES THE USERS ABILITY TO MOVE THE
11820 ** CURSORS SHORT DISTANCES AS WELL AS LONG DISTANCES.
11821 ** THE FACT THE SPEED OF THE CURSORS IS PROPORTIONAL TO
11822 ** THE CRT GRATICULE REQUIRES THE MOTION OF THE CURSORS TO
11823 ** VARY DEPENDING UPON THE RESOLUTION OF THE WAVEFORMS. THE
11824 ** SMALLER THE RESOLUTION THE SLOWER THE CURSORS CAN JUMP
11825 ** FROM POINT TO POINT ON THE WAVEFORM. THIS IS ACCOMPLISHED
11826 ** BY EXECUTING A LOOP THE SAME NUMBER OF TIMES REGARDLESS OF
11827 ** THE RESOLUTION AND ONLY MOVING THE CURSORS IN PRESELECTED
11828 ** PASSES THROUGH THE LOOP. THE ACCELERATION IS CREATED BY
11829 ** SHORTENING THE DELAY BETWEEN MOVEMENTS.
11830 **

```

```

11831 *****
11832          5F85      KEYCRS1LFT EQU *                                08979
11833 5F86 0701          SET0 R1          MOVE CURSOR #1 LEFT          08980
11834 5F88 04C2          CLR R2          DON'T MOVE CURSOR #2          08981
11835 5F8A 1003          JMP CRS1KEY
11836          5F8C      KEYCRS1RGT EQU *                                08983
11837 5F8C 0201 0001     LI R1,1          MOVE CURSOR #1 RIGHT          08984
11838 5F90 04C2          CLR R2          DON'T MOVE CURSOR #2          08985
11839 5F92 8820 D954 3338 CRS1KEY C     CURSOR,C0     IS CURSOR #1 BEING DISPLAYED? 08986
11840 5F98 1510          JGT CRSKEY          YES, START MOVING CURSOR #1 08987
11841 5F9A 04E0 D94A     CLR FATAL
11842 5F9E 0388          RTWP          NO, THIS KEY IS A NOOP          08989
11843          5FA0      KEYCRS2LFT EQU *                                08990
11844 5FA0 04C1          CLR R1          DON'T MOVE CURSOR #1          08991
11845 5FA2 0702          SET0 R2          MOVE CURSOR #2 LEFT          08992
11846 5FA4 1003          JMP CRS2KEY
11847          5FA6      KEYCRS2RGT EQU *                                08994
11848 5FA6 04C1          CLR R1          DON'T MOVE CURSOR #1          08995
11849 5FA8 0202 0001     LI R2,1          MOVE CURSOR #2 RIGHT          08996

```


KEYCRS1LFT / KEYCRS1RGT / KEYCRS2LFT / KEYCRS2RGT --- CURSOR 08978

11850	5FAC	8820	D954	333A	CRS2KEY	C	CURSOR,C1	IS CURSOR #2 BEING DISPLAYED?	08997
11851	5FB2	1503			JGT	CRSKEY		YES, START MOVING CURSOR #2	08998
11852	5FB4	04E0	D94A		CLR	FATAL			08999
11853	5FB8	0380			RTWP			NO, THIS KEY IS A NOOP	09000
11854					*				
11855					*	R1	:= INCREMENT FOR CURSOR #1		
11856					*	R2	:= INCREMENT FOR CURSOR #2		
11857					*				
11858	5FBA	0700			CRSKEY	SETD	R0	LOAD KEY # OF \$FF	09001
11859	5FBC	8820	DAA0	3336	C	PROGRAM,CN1		IS SYSTEM IN PROGRAM EXECUTE MODE?	09002
11860	5FC2	1505			JGT	PROGCRS		YES, USE KEY # OF \$FF FOR SINGLE STEP ONLY	09003
11861	5FC4	C320	2042		MOV	KEYID,R12		SET CRU BIT TO READ KEYCODE	09004
11862	5FC8	1E00			SBZ	0			09005
11863	5FCA	0020	E00F		MOV3	KBCODE+1,R0		READ KEYCODE OF KEY CURRENTLY DEPRESSED	09006
11864					*				
11865					*	R0	:= KEYCODE OF KEY CURRENTLY DEPRESSED		
11866					*				
11867	5FCE	04C3			PROGCRS	CLR	R3	CREATE DOUBLE-WORD MAXIMUM RESOLUTION	09007
11868	5FD0	0204	0400		LI	R4,1024			09008
11869	5FD4	3CE0	0970		DIV	RESOLV,R3		CALCULATE DELAY LOOP COUNT	09009
11870	5FD8	C220	0970		MOV	RESOLV,R8			09010
11871	5FDC	0978			SRL	R8,7		NUMBER OF POINTS TO MOVE BEFORE READOUT	09011
11872	5FDE	C2C8			MOV	R8,R11		IS TO BE UPDATED	09012
11873	5FE0	0204	0400		LI	R4,\$0430		INITIAL CURSOR MOVEMENT DELAY	09013
11874					*				
11875					*	R3	:= COUNT OF DELAY LOOP EXECUTIONS		
11876					*	R4	:= DELAY COUNT BETWEEN CURSOR MOVEMENTS		
11877					*	R8	:= R11 := DELAY BEFORE READOUT IS UPDATED		
11878					*				
11879	5FE4	020C	3000		LI	R12,\$3000		DELAY COUNT FOR SINGLE STEP	09014
11880	5FE8	9020	E00F		CRSTEP	CB	KBCODE+1,R0	IF KEY IS STILL DOWN, DELAY TO ALLOW	09015
11881	5FEC	1502			JNE	MOVCRS		CURSOR SINGLE STEPPING	09016
11882	5FEE	060C			DEC	R12			09017
11883	5FF0	15FB			JGT	CRSTEP			09018
11884	5FF2	C141			MOVCRS	MOV	R1,R5	SETUP INITIAL MOVEMENT VALUES	09019
11885	5FF4	C182			MOV	R2,R6			09020
11886	5FF6	C1C3			MOV	R3,R7			09021
11887					*				
11888					*	R5	:= INCREMENT FOR CURSOR #1		
11889					*	R6	:= INCREMENT FOR CURSOR #2		
11890					*	R7	:= COUNT OF DELAY LOOP EXECUTIONS		
11891					*				
11892	5FF8	A805	D956		CRS1	A	R5,CURS1	MOVE CURSOR #1	09022
11893	5FFC	1505			JGT	CRS1R		CURSOR #1 RANGE IS 0<=CRS 1<=RESOLUTION-1	09023
11894	5FFE	1304			JEQ	CRS1R			09024
11895	6000	04E0	D956		CLR	CURS1		SET CURSOR #1 TO 0	09025
11896	6004	C145			MOV	R5,R5		STOP MOVING ONLY IF CURSOR #1 IS MOVING	09026
11897	6006	1141			JLT	CRSWRN		STOP CURSOR MOVEMENT WHEN RANGE LIMIT IS MET	09027
11898	6008	8820	D970	D956	CRS1R	C	RESOLV,CURS1		09028
11899	600E	1506			JGT	CRS2			09029
11900	6010	C820	D970	D956	MOV	RESOLV,CURS1		SET CURSOR #1 TO RESOLUTION-1	09030
11901	6016	0620	D956		DEC	CURS1			09031
11902	601A	1037			JMP	CRSWRN		STOP CURSOR MOVEMENT WHEN RANGE LIMIT IS MET	09032
11903	601C	A806	D958		CRS2	A	R6,CURS2	MOVE CURSOR #2	09033
11904	6020	8820	D958	D956	C	CURS2,CURS1		CURSOR #2 RANGE IS CRS 1<=CRS 2<=RESOLUTION-1	09034
11905	6026	1506			JGT	CRS2R			09035
11906	6028	1305			JEQ	CRS2R			09036
11907	602A	C820	D956	D958	MOV	CURS1,CURS2		SET CURSOR #2 TO CURSOR #1	09037
11908	6030	C186			MOV	R6,R5		CURSOR #1 CAN PUSH CURSOR #2	09038

KEYCRS1LFT / KEYCRS1RGT / KEYCRS2LFT / KEYCRS2RGT --- CURSOR 09978

11909	6032	112B				JLT	CRSWRN	STOP CURSOR MOVEMENT WHEN RANGE LIMIT IS MET	09039
11910	6034	8820	D970	D958	CRS2R	C	RESOLV,CURS2		09040
11911	603A	1506				JGT	DLYCRS		09041
11912	603C	C820	D970	D958		MOV	RESOLV,CURS2	SET CURSOR #2 TO RESOLUTION-1	09042
11913	6042	0620	D958			DEC	CURS2		09043
11914	6046	1021				JMP	CRSWRN	STOP CURSOR MOVEMENT WHEN RANGE LIMIT IS MET	09044
11915	6048	9020	E00F		DLYCRS	CB	KBCODE*1,R0	IS KEY STILL DEPRESSED?	09045
11916	604C	1628				JNE	CRSR0	NO, THIS KEY IS DONE	09046
11917	604E	C304				MOV	R4,R12	DELAY BETWEEN MOVEMENTS	09047
11918	6050	060C				DEC	R12		09048
11919	6052	15FE				JGT	*-2		09049
11920	6054	04C5				CLR	R5	SET INCREMENTS TO ZERO	09050
11921	6056	04C6				CLR	R6	DON'T MOVE CURSORS IN DELAY LOOPS	09051
11922	6058	0607				DEC	R7	KEEP SPEED CONSTANT REGARDLESS OF RESOLUTION	09052
11923	605A	15CE				JGT	CRS1		09053
11924	605C	0608				DEC	R11	CHECK IF READOUT SHOULD BE UPDATED	09054
11925	605E	1511				JGT	CHKKEY		09055
11926	6060	C304				MOV	R4,R12	DECREASE DELAY TO CREATE ACCELERATION	09056
11927	6062	093C				SRL	R12,3		09057
11928	6064	610C				S	R12,R4	SUBTRACT 1/8 OF REMAINING DELAY EACH TIME	09058
11929	6066	C2C8				MOV	R8,R11		09059
11930	6068	E820	3340	D972		SOC	CLINE15,ROTFLAG	UPDATE LINE 15 OF READOUT	09060
11931	606E	0420	1430			BLW ^P	FRDOUT		09061
11932	6072	8820	D956	D958		C	CURS1,CURS2	IF BOTH CURSORS ARE ON SAME POINT, ADD	09062
11933	6078	1604				JNE	CHKKEY	AN ADDITIONAL DELAY AS CONVERSIONS TO ASCII	09063
11934	607A	020C	0100			LI	R12,\$100	OF ZERO ARE FASTER THAN CONVERSIONS OF	09064
11935	607E	060C				DEC	R12	NON-ZERO NUMBERS. THIS IS TO KEEP THE	09065
11936	6080	15FE				JGT	*-2	SPEED CONSTANT WHEN CURSORS ARE TOGETHER.	09066
11937	6082	9020	E00F		CHKKEY	CB	KBCODE*1,R0		09067
11938	6086	13B5				JEQ	MOVCRS	YES, CONTINUE MOVING CURSOR	09068
11939	6088	1002				JMP	CRSR0	NO, STOP CURSOR MOVEMENT	09069
11940	608A	04E0	D94C		CRSWRN	CLR	WARNING	SET WARNING TO TELL USER	09070
11941	608E	E820	3340	D972	CRSR0	SOC	CLINE15,ROTFLAG	UPDATE LINE 15 OF READOUT	09071
11942	6094	0380				RTW ^P			09072

KEYVPDN / KEYVPUP --- VERTICAL POSITION SLEW KEYS

09073

```

11944 *****
11945 **
11946 ** VERTICAL POSITION SLEW KEYS- 'VPUP', 'VPDN' **
11947 **
11948 ** LEVEL 1 ROUTINE **
11949 **
11950 ** INPUT- NONE **
11951 ** OUTPUT- NONE **
11952 **
11953 ** STACK OPERATIONS- **
11954 ** SOFTSTACK- USED BUT NO EFFECT TO CALLER **
11955 **
11956 ** NOTE --- **
11957 ** THE VERTICAL POSITION KEYS GIVE THE USER CONTINUOUS **
11958 ** MOTION CONTROL OF THE VERTICAL POSITION OF THE CURRENT **
11959 ** WAVEFORM. TWO BUTTONS CONTROL THE MOVEMENT OF THE CURRENT **
11960 ** WAVEFORM. ONE BUTTON FOR UPWARD MOVEMENT AND ONE BUTTON **
11961 ** FOR DOWNWARD MOVEMENT. **
11962 ** THE SPEED OF THE VERTICAL POSITIONING IS PROPORTIONAL **
11963 ** TO THE CRT GRATICULE NOT TO THE RESOLUTION OF THE WAVEFORMS. **
11964 ** THERE IS AN INITIAL DELAY AFTER FIRST DEPRESSING THE BUTTON **
11965 ** TO ALLOW THE USER TO SINGLE STEP THE WAVEFORM UP OR DOWN. **
11966 ** THE CONSTANT SPEED REGARDLESS OF THE RESOLUTION IS **
11967 ** ACCOMPLISHED BY EXECUTING A LOOP THE SAME NUMBER OF TIMES **
11968 ** REGARDLESS OF THE RESOLUTION BUT ONLY MOVING THE WAVEFORM **
11969 ** IN PRESELECTED PASSES THROUGH THE LOOP. **
11970 **
11971 *****
11972 6096 0201 FFF0 KEYVPDN LI R1,-16 MOVE CWF DOWN 1 INCREMENT INITIALLY 09074
11973 609A 020B 8001 LI R11,8001 NEGATIVE CLIPPING VALUE 09075
11974 609E 1004 JMP UPDOWN 09076
11975 60A0 0201 0010 KEYVPJP LI R1,16 MOVE CWF UP 1 INCREMENT INITIALLY 09077
11976 60A4 020B 7FFF LI R11,7FFF POSITIVE CLIPPING VALUE 09078
11977 60A8 0700 UPDOWN SETJ R0 USE KEYCODE OF $FF IF SYSTEM IS IN 09079
11978 60AA 8820 DAA0 3336 C PROGRAM,CN1 PROGRAM EXECUTE MODE SO RESULT IS 09080
11979 60B3 1505 JGT PROGVP SINGLE STEP ONLY 09081
11980 60B2 C320 2042 MOV KEYID,R12 READ CURRENT KEY CODE 09082
11981 60B6 1E00 SBZ 0 09083
11982 60B8 0020 E00F MOV3 KBCODE+1,R0 09084
11983 60BC 04C2 PROGVP CLR R2 CREATE DOUBLE-WORD OF MAXIMUM RESOLUTION 09085
11984 60BE 0203 0400 LI R3,1024 09086
11985 60C2 3CA0 0970 DIV RESOLV,R2 DIVIDE BY CURRENT RESOLUTION TO GET LOOP COUNT 09087
11986 60C6 C0E0 095E MOV OPWF4H,R3 09088
11987 60CA A0E0 3348 A VOFFAB,R3 CWF'S VERTICAL OFFSET 09089
11988 60CE 0708 SETJ R0 09090
11989 60D0 C101 MOVVPOS MOV R1,R4 FIRST TIME THROUGH LOOP MOVE WAVEFORM 09091
11990 60D2 C142 MOV R2,R5 SETUP NEW LOOP COUNT 09092
11991 60D4 C1A0 0970 DLYVPOS MOV RESOLV,R6 COUNT OF POINTS TO CHANGE 09093
11992 60D8 C1E0 095C MOV OPWF4H,R7 STARTING ADDRESS OF CWF DATA 09094
11993 60DC 64C4 S R4,*R3 ADJUST VERTICAL OFFSET 09095
11994 60DE 1907 JND VPOSNT 09096
11995 60E0 A113 A *R3,R4 GET LAST VERTICAL OFFSET 09097
11996 60E2 A10B A R11,R4 MAXIMUM VALUE FOR LAST MOVEMENT 09098
11997 60E4 C4CB MOV R11,*R3 ON OVERFLOW LOAD CLIPPING VALUE 09099
11998 60E6 0513 NEG *R3 VERTICAL OFFSET IS STORED AS NEGATIVE ACTUAL 09100
11999 60E8 04E0 094A CLR FATAL SET ERROR FLAG TO TELL USER 09101
12000 60EC 0700 SETJ R0 SET KEYCODE TO AN INVALID CODE 09102
12001 60EE ADC4 VPOSNT A R4,*R7+ INCREMENT/DECREMENT POINT 09103
12002 60F0 1908 JND VPOSNT 09104

```

12003	60F2	8820	D95A	3338	C	OPWF4,30	IS THIS WFM 0?	09185
12004	60F3	1618			JNE	VPST0P	NO, THEN CANNOT CLIP	09186
12005	60FA	34E0	D94C		CLR	WARNING	SET WARNING FLAG TO TELL USER ABOUT CLIPPING	09187
12006	60FE	0647			DECT	R7		09188
12007	6100	30C8			MOV	R11,*R7+	ON OVERFLOW LOAD CLIPPING VALUE	09189
12008	6102	0606			VPOSXKT DEC	R6	CHECK IF POSITIONING LOOP IS DONE	09110
12009	6104	15F4			JGT	VPOSNT	CONTINUE IF NOT	09111
12010	6106	04C4			CLR	R4	SET INCREMENT/DECREMENT TO ZERO	09112
12011	6108	0605			DEC	R5	CHECK IF DELAY LOOPS SHOULD BE RUN FOR	09113
12012	610A	15E4			JGT	DLYVPOS	SMALLER RESOLUTIONS TO KEEP SPEED CONSTANT	09114
12013	610C	0420	186C		BLWP	VZRR0	UPDATE VERTICAL ZERO READOUT DISPLAY	09115
12014	6110	0588			INC	R8	SHOULD SINGLE-STEP DELAY BE ADDED?	09116
12015	6112	1808			JH	CHKVKEY	NO, MOVE WFM AT CONSTANT SPEED	09117
12016	6114	0A21			SLA	R1,2	MULTIPLY SPEED BY FOUR	09118
12017	6116	020C	3000		LI	R12,\$3000	SINGLE-STEP DELAY COUNT	09119
12018	611A	9020	E00F		VSTEP CB	KBCODE+1,R0	IS THIS KEY STILL DOWN?	09120
12019	611E	1602			JNE	CHKVKEY	NO, STOP PROCESSING	09121
12020	6120	060C			DEC	R12		09122
12021	6122	15FB			JGT	VPSTEP		09123
12022	6124	9020	E00F		CHKVKEY CB	KBCODE+1,R0	CHECK IF KEY CODE IS STILL SAME	09124
12023	6128	1303			JEQ	MOVVPOS	IF KEY IS STILL DOWN CONTINUE	09125
12024	612A	0720	D972		SETJ	RDTFLAG	UPDATE READOUT LINE #1	09126
12025	612E	0380			RTWP			09127
12026	6130	A4C4			VSTOP A	R4,*R3	SET VERTICAL OFFSET BACK	09128
12027	6132	0647			DECT	R7	POINT BACK TO POINT THAT OVERFLOWED	09129
12028	6134	C177			MOV	*R7+,R5	SAVE THE OVERFLOWED VALUE	09130
12029	6136	0647			VBACK DECT	R7	POINT BACK TO LAST POINT	09131
12030	6138	65C4			S	R4,*R7	SET POINT BACK TO LAST LEVEL	09132
12031	613A	0586			INC	R6	INCREMENT POINT COUNT	09133
12032	613C	8806	D970		C	R6,RESOLV	HAVE ALL POINTS BEEN SET BACK?	09134
12033	6140	11FA			JLT	VPBACK	NO, CONTINUE	09135
12034	6142	13F9			JEQ	VPBACK		09136
12035	6144	6105			S	R5,R4	CALCULATE LAST INCREMENT/DECREMENT	09137
12036	6146	A108			A	R11,R4		09138
12037	6148	1307			JEQ	VPDONE	IF ZERO, VERTICAL POSITIONING IS DONE	09139
12038	614A	64C4			S	R4,*R3	ADJUST VERTICAL ZERO	09140
12039	614C	0606			DEC	R6		09141
12040	614E	ADC4			A	R4,*R7+	INCREMENT/DECREMENT WAVEFORM LAST TIME	09142
12041	6150	1901			JND	*+4		09143
12042	6152	18EE			JMP	VPST0P	CHECK FOR FURTHER OVERFLOW CONDITIONS	09144
12043	6154	3606			DEC	R6		09145
12044	6156	15FB			JGT	*-8		09146
12045	6158	0720	D972		VDONE SETJ	RDTFLAG	UPDATE READOUT	09147
12046	615C	04E0	D94A		CLR	FATAL	SET ERROR FLAG	09148
12047	6160	0380			RTWP			09149

12049				*				
12050				*	HANDLE WFM KEY			
12051				*				
12052				*	LEVEL 1 ROUTINE			
12053				*				
12054	6162	06A0	690E	KEYWFM	BL POPREG	POP X		09151
12055	6166	1607			JNE NOGOOD4	ERROR IF WFM		09152
12056	6168	06A0	75A0		BL FP2INT	CONVERT X TO INT		09153
12057	616C	1104			JLT NOGOOD4	MUST BE IN RANGE 0<=X<=WFMAXN		09154
12058	616E	8801	D974		C R1,WFMAXN			09155
12059	6172	1501			JGT NOGOOD4			09156
12060	6174	1007			JMP GOOD			09157
12061	6176	04E0	D94A	NOGOOD4	CLR FATAL	SET ERROR FLAG		09158
12062	617A	C020	33D4		MOV WSTK,R0			09159
12063	617E	C6A0	693E		BL PSHSTK	PUSH DATA BACK ON STACK		09160
12064	6182	0380			RTWP			09161
12065	6184	0200	0001	GOOD	LI R0,1	FLAG STACK AS WFM		09162
12066	6188	06A0	6962		BL PSHREG	PUSH WFM ON STACK		09163
12067	618C	E020	3348	D972	SOC CLINE15,RTFLAG	FLAG DISPLAY CHANGE		09164
12068	6192	0380		RECDON	RTWP			09165

KEY2WFM

09166

```

12070 *
12071 * HANDLE >WF KEY
12072 *
12073 * LEVEL 1 ROUTINE
12074 *
12075 6194 06A0 690E KEY2WFM BL POPREG POP X 09167
12076 6198 16EE JNE NOGOOD4 X MUST BE CONSTANT 09168
12077 619A 06A0 75A0 BL FP2INT 09169
12078 619E 11EB JLT NOGOOD4 09170
12079 61A0 8801 0974 C R1,WFMAXN 09171
12080 61A4 15E8 JGT NOGOOD4 09172
12081 61A6 06A0 6984 3L ADRWFM GET DATA AND HEADER ADDRESS FOR X 09173
12082 61AA C03A MOV *USER,R3 09174
12083 61AC 13E4 JEQ NOGOOD4 09175
12084 61AE C141 MOV R1,R5 SAVE R1 FROM X 09176
12085 61B0 06A0 690E BL POPREG POP Y 09177
12086 61B4 C045 MOV R5,R1 RESTORE R1 BEFORE PUSHING X ON STACK 09178
12087 61B6 0200 0001 LI R0,1 09179
12088 61BA 06A0 6962 BL PSHREG PUSH WFM X 09180
12089 61BE C641 MOV R1,*SOFT 09181
12090 61C0 0649 DECT SOFT 09182
12091 61C2 C650 095A MOV OPWFM,*SOFT 09183
12092 61C6 0420 698C BLWP XFRHEAD 09184
12093 61CA C039 MOV *SOFT+,R0 DATA POINTER 09185
12094 61CC C050 095C MOV OPWFM,R1 09186
12095 61D0 C0A0 0970 MOV RESOLV,R2 09187
12096 61D4 CC31 MOVDATA MOV *R1+,*R0* 09188
12097 61D6 0602 DEC R2 09189
12098 61D8 16FD JNE MOVDATA 09190
12099 61DA 0720 0972 SETD RDTFLAG 09191
12100 61DE 0380 RTWP 09192

```

12102				*				
12103				*	HANDLE CNS KEY			
12104				*				
12105				*	LEVEL 1 ROUTINE			
12106				*				
12107	61E0	06A0	6218	KEYCNS	BL CONCOM			09194
12108	61E4	C075			MOV *R5+,R1			09195
12109	61E6	C095			MOV *R5,R2			09196
12110	61E8	04C0			CLR R0			09197
12111	61EA	06A0	6962		BL PSHREG	PUSH CONSTANT VALUE		09198
12112	61EE	E820	3348	0972	SOC CLINE15, RDTFLAG			09199
12113	61F4	0380			RTWP			09200

12115				*				
12116				*	HANDLE	>CN	KEY	
12117				*				
12118				*	LEVEL	1	ROUTINE	
12119				*				
12120	61F6	06A0	6218		KEY2CNS	BL	CONCOM	09202
12121	61FA	06A0	690E			BL	POPREG	09203
12122	61FE	1302				JEQ	PSHCON	09204
12123	6200	06A0	757E			BL	INT2FP	09205
12124	6204	0D41			PSHCON	MOV	R1,*R5+	09206
12125	6206	0542				MOV	R2,*R5	09207
12126	6208	0020	33D4			MOV	WSTK,R0	09208
12127	620C	06A0	6936				PUT CONSTANT BACK ON STACK	09208
12127	620C	06A0	6936			BL	PSHSTK	09209
12128	6210	E820	3348	0972		SOC	CLINE16,RDTFLAG	09210
12129	6216	0380				RTWP		09211

CONCOM

09212

12131				*				
12132				*	DO COMMON PORTION OF KEYCNS AND KEY2CNS			
12133				*				
12134				*	OBTAIN CONSTANT #, VERIFY CORRECTNESS, AND COMPUTE			
12135				*	CONSTANT ADDRESS			
12136				*				
12137	6218	C188		CONCOM	MOV R11,R6			09213
12138	621A	06A0	690E		BL POPREG			09214
12139	621E	160C			JNE NOG00D5			09215
12140	6220	06A0	75A0		BL FP2INT			09216
12141	6224	1109			JLT NOG00D5	CONST # < 0		09217
12142	6226	0060	DADC		C MAXCNS,R1			09218
12143	622A	1306			JEQ NOG00D5			09219
12144	622C	1105			JLT NOG00D5	CONS # TOO HIGH		09220
12145	622E	0A21			SLA R1,2			09221
12146	6230	A050	DADA		A CONSTR,R1	CONSTANT ADDRESS		09222
12147	6234	C141			MOV R1,R5			09223
12148	6236	0456			B *R6			09224
12149	6238	0450	6175	N33335	B NOG00D4			09225

KEYPNT

09226

12151						*****		
12152						**		**
12153					**	'PNT' KEY HANDLER		**
12154					**			**
12155					**	LEVEL 1 ROUTINE		**
12156					**			**
12157					**	STACK OPERATIONS:		**
12158					**	SOFTSTACK - USED BUT NO AFFECT TO CALLER		**
12159					**	USERSTACK - POPS 1 PUSHES 1		**
12160					**			**
12161						*****		
12162	623C	06A0	690E		KEYPNT	BL POPREG	POP X OFF USERSTACK	09227
12163	6240	1606				JNE ERRPNT	ERROR IF X IS A WFM #	09228
12164	6242	06A0	75A0			BL FP2INT	CONVERT FLOATING POINT NUMBER TO INTEGER	09229
12165	6246	1103				JLT ERRPNT	RANGE IS: 0<=X<RESOLV	09230
12166	6248	8801	0970			C R1,RESOLV		09231
12167	624C	1107				JLT OKPNT		09232
12168	624E	C020	3304		ERRPNT	MOV WSTK,R0	PUSH X BACK ONTO USERSTACK	09233
12169	6252	06A0	6936			BL PSHSTK		09234
12170	6256	04E0	094A			CLR FATAL	SET ERROR FLAG	09235
12171	625A	0380				RTW ²		09236
12172	625C	0420	68A6		OKPNT	BLW ² PNT2FP	FIND VALUE OF THIS POINT	09237
12173	6260	04C0				CLR R0	FLAG INDICATING FP NUMBER	09238
12174	6262	06A0	6962			BL PSHREG	PUSH VALUE ONTO USERSTACK	09239
12175	6266	E820	3348	0972		SOC CLINE15,RDTFLAG	UPDATE LINE 16 OF READOUT	09240
12176	626C	0380				RTW ²		09241

12178				*****			
12179				**			**
12180				**	'>PNT' KEY HANDLER		**
12181				**			**
12182				**	LEVEL 1 ROUTINE		**
12183				**			**
12184				**	STACK OPERATIONS:		**
12185				**	SOFTSTACK - USED BUT NO AFFECT TO CALLER		**
12186				**	USERSTACK - POPS 1		**
12187				**			**
12188				*****			
12189	626E	D6A0	690E	KEY2PNT	BL	POPREG	POP X OFF USERSTACK (POINT NUMBER) 09243
12190	6272	C142			MOV	R2,R5	09244
12191	6274	C101			MOV	R1,R4	SAVE DATA POPPED FROM X 09245
12192	6276	C0C8			MOV	R0,R3	09246
12193	6278	162D			JNE	ERR2PNT	ERROR IF X IS A WFM # 09247
12194	627A	D6A0	75A0		BL	FP2INT	CONVERT FLOATING POINT NUMBER TO INTEGER 09248
12195	627E	112A			JLT	ERR2PNT	RANGE IS: 0<=X<RESOLV 09249
12196	6280	8060	D970		C	RESOLV,R1	09250
12197	6284	1127			JLT	ERR2PNT	09251
12198	6286	1326			JEQ	ERR2PNT	09252
12199	6288	C181			MOV	R1,R5	SAVE DATA POINT NUMBER 09253
12200	628A	D6A0	690E		BL	POPREG	GET Y OFF USERSTACK (POINT VALUE) 09254
12201	628E	D6A0	6962		BL	PSHREG	LEAVE Y ON USERSTACK 09255
12202	6292	C000			MOV	R0,R0	09256
12203	6294	161F			JNE	ERR2PNT	ERROR IF Y IS A WFM # 09257
12204	6296	D420	6868		BLWP	FP2WFM	CONVERT Y TO WAVEFORM DATA VALUE 09258
12205	629A	C0A0	D94C		MOV	WARNING,R2	WAS THIS CONVERSION SUCCESSFUL? 09259
12206	629E	151A			JGT	ERR2PNT	NO, THIS IS AN ERROR 09260
12207	62A0	1319			JEQ	ERR2PNT	09261
12208	62A2	C1E0	D970		MOV	RESOLV,R7	GET CURRENT WAVEFORM RESOLUTION 09262
12209	62A6	C220	D95C		MOV	OPWFM,R8	GET ADDRESS OF OPWFM 09263
12210	62AA	C2C1			MOV	R1,R11	MOVE NEW POINT TO MIN 09264
12211	62AC	C301			MOV	R1,R12	MOVE NEW POINT TO MAX 09265
12212	62AE	82D8		M42PNT	C	*R8,R11	FIND NEW MAX AND MIN 09266
12213	62B0	1101			JLT	*+4	09267
12214	62B2	C2D8			MOV	*R8,R11	KEEP CURRENT MAX IN R11 09268
12215	62B4	8318			C	*R8,R12	09269
12216	62B6	1501			JGT	*+4	09270
12217	62B8	C318			MOV	*R8,R12	KEEP CURRENT MIN IN R12 09271
12218	62BA	D5C8			INCT	R8	09272
12219	62BC	D607			DEC	R7	09273
12220	62BE	15F7			JGT	M42PNT	SCAN ENTIRE WAVEFORM FOR MAX AND MIN 09274
12221	62C0	62CC			S	R12,R11	IS P-P WITHIN 20 DIVISIONS (DATA RANGE)? 09275
12222	62C2	1901			JND	*+4	09276
12223	62C4	1007			JMP	ERR2PNT	09277
12224	62C6	0A16			SLA	R6,1	CALCULATE ADDRESS IN OPWFM OF POINT NUMBER 09278
12225	62C8	A1A0	D95C		A	OPWFM,R6	09279
12226	62CC	C581			MOV	R1,*R6	MOVE POINT VALUE TO POINT 09280
12227	62CE	D720	D972		SETO	RDTFLAG	UPDATE ENTIRE READOUT 09281
12228	62D2	0380			RTWP		09282
12229	62D4	C085		ERR2PNT	MOV	R5,R2	09283
12230	62D6	C044			MOV	R4,R1	RESTORE X DATA 09284
12231	62D8	C003			MOV	R3,R0	09285
12232	62DA	D6A0	6962		BL	PSHREG	PUSH X BACK ONTO USERSTACK 09286
12233	62DE	D4E0	D94A		CLR	FATAL	SET ERROR FLAG 09287
12234	62E2	0380			RTWP		09288

12236				*					
12237				*	HANDLE >PW KEY				
12238				*					
12239				*	LEVEL 1 ROUTINE				
12240				*					
12241	62E4	06A0	690E		KEY2P.W BL POPREG				09290
12242	62E8	160E			JNE NOG00D3	MUST BE A CONSTANT			09291
12243	62EA	06A0	75A0		BL FP2INT				09292
12244	62EE	0281	0080		CI R1,128				09293
12245	62F2	1310			JEQ SET128				09294
12246	62F4	0281	0100		CI R1,256				09295
12247	62F8	1319			JEQ SET256				09296
12248	62FA	0281	0200		CI R1,512				09297
12249	62FE	1322			JEQ SET512				09298
12250	6300	0281	0400		CI R1,1024				09299
12251	6304	132B			JEQ SET1024				09300
12252	6306	04E0	094A		NOG00D3 CLR FATAL				09301
12253	630A	C020	33D4		MOV WSTK,R0				09302
12254	630E	06A0	6936		BL PSHSTK				09303
12255	6312	0380			RTWP				09304
12256	6314	C060	3376		SET128 MOV C128,R1				09305
12257	6318	C820	3398	D97A	MOV CHE008,YTWM	X PRELOAD FOR 128 P/W			09306
12258	631E	C820	33A0	D97C	MOV CHE008,YTWM+2	INTERLACE FOR 128 P/W			09307
12259	6324	0203	0007		LI R3,7				09308
12260	6328	0460	6370		B SETRES				09309
12261	632C	C060	3378		SET256 MOV C256,R1				09310
12262	6330	C820	3396	D97A	MOV CHE004,YTWM	X PRELOAD FOR 256 P/W			09311
12263	6336	C820	339E	D97C	MOV CHE044,YTWM+2	INTERLACE FOR 256 P/W			09312
12264	633C	0203	0006		LI R3,6				09313
12265	6340	0460	6370		B SETRES				09314
12266	6344	C060	337A		SET512 MOV C512,R1				09315
12267	6348	C820	3394	D97A	MOV CHE002,YTWM	X PRELOAD FOR 512 P/W			09316
12268	634E	C820	339C	D97C	MOV CHE022,YTWM+2	INTERLACE FOR 512 P/W			09317
12269	6354	0203	0005		LI R3,5				09318
12270	6358	0460	6370		B SETRES				09319
12271	635C	C060	337C		SET1024 MOV C1024,R1				09320
12272	6360	C820	3392	D97A	MOV CHE001,YTWM	X PRELOAD FOR 1024 P/W			09321
12273	6366	C820	339A	D97C	MOV CHE011,YTWM+2	INTERLACE FOR 1024 P/W			09322
12274	636C	0203	0004		LI R3,4				09323
12275	6370	0202	0007		SETRES LI R2,7				09324
12276				*	SZC R2,DSPLT				09325 DEL
12277				*	SZC R2,DSWFM				09326 DEL
12278				*	SZC R2,DSPCRS				09327 DEL
12279				*	SOC R3,DSPLT				09328 DEL
12280				*	SOC R3,DSWFM				09329 DEL
12281				*	SOC R3,DSPCRS				09330 DEL
12282				*					00027PATCH
12283				*	PROBLEM #21 - PATCH #27				00027PATCH
12284				*					00027PATCH
12285				*	CORRECTS PROBLEM THAT DISPLAY DONE INTERRUPT IS NOT ALWAYS GENERATED				00027PATCH
12286				*	WHEN P/W IS CHANGED				00027PATCH
12287				*					00027PATCH
12288	6374	0460	97FA		B PATCH27	BRANCH TO PATCH 27			00027PATCH
12289			6378		BACK27 EQU *	DEFINE REENTRY POINT			00027PATCH
12290				*					00027PATCH
12291				*	END OF PATCH #27, PROBLEM #21				00027PATCH
12292	6378	1009			JMP *+020				00027PATCH
12293	637A	1008			JMP *+018				00027PATCH
12294	637C	1007			JMP *+016				00027PATCH

12295	637E	1006		JMP *+014	00027PATCH
12296	6380	1005		JMP *+012	00027PATCH
12297	6382	1004		JMP *+010	00027PATCH
12298	6384	1003		JMP *+008	00027PATCH
12299	6386	1002		JMP *+006	00027PATCH
12300	6388	1001		JMP *+004	00027PATCH
12301	638A	1000		JMP *+002	00027PATCH
12302	638C	C020	D970	MOV RESOLV,R0	09331
12303	6390	C801	D970	MOV R1,RESOLV	09332
12304	6394	06A0	757E	BL INT2FP	09333
12305	6398	04C0		CLR R0	09334
12306	639A	C820	D970 D958	MOV RESOLV,CURS2	09335
12307	63A0	0620	D958	DEC CURS2	09336
12308	63A4	04E0	D956	CLR CURS1	09337
12309	63A8	C020	DAE4	MOV RAMOPT,R3	09338
12310	63AC	8830	D970	C *R0+,RESOLV	09339
12311	63B0	1302		JEQ *+6	09340
12312	63B2	05C0		INCT R0	09341
12313	63B4	10FB		JMP *-8	09342
12314	63B6	C050		MOV *R0,R1	09343
12315	63B8	13A6		JEQ NOG00D3	09344
12316	63BA	0601		DEC R1	09345
12317	63BC	C801	D974	MOV R1,WFMAXN	09346
12318			63C0	CLRDS EQU *	09347
12319	63C0	0601		DEC R1	09348
12320	63C2	1108		JLT CLRSTK	09349
12321	63C4	06A0	6984	BL ADRWFM	09350
12322	63C8	C0B9		MOV *SOFT+,R2	09351
12323	63CA	05C9		INCT SOFT	09352
12324	63CC	A0A0	3354	A DISPLA,R2	09353
12325	63D0	0402		CLR *R2	09354
12326	63D2	10F6		JMP CLRDSP	09355
12327			63D4	CLRSTK EQU *	09356
12328	63D4	0200	0001	LI R0,1	09357
12329	63D8	0201	0001	LI R1,1	09358
12330	63DC	06A0	6962	BL PSHREG	09359
12331	63E0	0420	6C98	BLWP OPWCHS	09360
12332	63E4	04C1		CLR R1	09361
12333	63E6	06A0	6962	BL PSHREG	09362
12334	63EA	0420	6C98	BLWP OPWCHS	09363
12335	63EE	04C0		CLR R0	09364
12336	63F0	04C1		CLR R1	09365
12337	63F2	04C2		CLR R2	09366
12338	63F4	06A0	6962	BL PSHREG	09367
12339	63F8	06A0	6962	BL PSHREG	09368
12340	63FC	06A0	6962	BL PSHREG	09369
12341	6400	06A0	6962	BL PSHREG	09370
12342	6404	06A0	6962	BL PSHREG	09371
12343	6408	06A0	6962	BL PSHREG	09372
12344	640C	0720	D972	SET0 RDTFLAG	09373
12345	6410	04E0	D978	CLR XYWFM SET TO X VS. TIME DISPLAY	09374
12346	6414	4820	3362 D9CE	SZC MWXVSY,DSPWFM	09375
12347	641A	0380		RTWP	09376

12350				*				
12351				*	HANDLE P/W KEY			
12352				*				
12353				*	LEVEL 1 ROUTINE			
12354				*				
12355				*	RETURNS RESOLUTION OF WAVEFORM INTO X			
12356				*				
12357	641C	C060	D970		KEYP.W MOV RESOLV,R1	GET RESOLUTION OF WAVEFORM		09379
12358	6420	06A0	757E		BL INT2FP	CONVERT TO FLOATING POINT		09380
12359	6424	04C0			CLR R0			09381
12360	6426	06A0	6962		BL PSHREG	RETURN WAVEFORM RESOLUTION		09382
12361	642A	E820	3348	D972	SOC CLINE15,RDTFLAG	UPDATE LAST LINE OF DISPLAY		09383
12362	6430	0380			RTWP			09384

KEYCLX / KEYCLR

09385

12364				*					
12365				*	HANDLE CLX AND CLS KEY				
12366				*					
12367				*	LEVEL 1 ROUTINE				
12368				*					
12369				*	INPUT NONE				
12370				*	OUTPUT	STACK USER ADJUSTED			
12371				*					
12372				*	INPUT NO REG				
12373				*	OUTPUT THROUGH USER				
12374				*	DESTROYS NONE				
12375				*					
12376	6432	04C0			KEYCLR CLR R0			09386	
12377	6434	04C1			CLR R1			09387	
12378	6436	04C2			CLR R2			09388	
12379	6438	0203	0007		LI R3,7			09389	
12380	643C	06A0	6962		BL PSHREG	PUSH ZERO ONTO STACK (IN R1,R2)		09390	
12381	6440	0603			DEC R3	CLEAR ENTIRE STACK		09391	
12382	6442	16FC			JNE *-6			09392	
12383	6444	E820	3348 0972		SOC GLINE15,RDTFLAG	UPDATE LINE 16 OF DISPLAY		09393	
12384	644A	0380			RTMP			09394	
12385	644C	06A0	6918		KEYCLX BL POPSTK	CLX - POP OFF X VALUE		09395	
12386	6450	E820	3348 0972		SOC GLINE15,RDTFLAG	REFLECT CHANGE IN X DISPLAY		09396	
12387	6456	0380			RTMP			09397	

12389				*				
12390				*	HANDLE 'RDN' KEY			
12391				*				
12392				*	LEVEL 1 ROUTINE			
12393				*				
12394	6458	06A0	690E		KEYROLL BL POPREG			09399
12395	645C	C120	330C		MOV TSTK,R4			09400
12396	6460	C000			MOV R0,*R4+			09401
12397	6462	C001			MOV R1,*R4+			09402
12398	6464	C502			MOV R2,*R4			09403
12399	6466	E020	3348	0972	SOC CLINE16, RDTFLAG			09404
12400	646C	0380			RTWP			09405

KEYX2Y

09405

```

12402 *
12403 *      HANDLE X<>Y KEY
12404 *
12405 *      LEVEL 1 ROUTINE
12406 *
12407 *      INPUT          SOFT IS SOFTWARE STACK POINTER
12408 *                  USER IS USER STACK POINTER
12409 *
12410 *      OUTPUT        USER S(0) AND S(1) ARE SWITCHED
12411 *
12412 *      INPUT USER STACK
12413 *      OUTPUT USER STACK
12414 *      DESTROYS R0,R1,R2,R3,R4
12415 *

```

12416	646E	06A8	690E	KEYX2Y	3L POPREG	09407
12417	6472	06A0	6918		BL POPSTK	09408
12418	6476	06A0	6962		BL PSHREG	09409
12419	647A	C020	33D4		MOV WSTK,R0	09410
12420	647E	06A0	6936		BL PSHSTK	09411
12421	6482	E820	3348	D972	SDC CLINE15,ROTFLAG	09412
12422	6488	0380			RTWP	09413

12424				*				
12425				*	HANDLE ENTER KEY			
12426				*				
12427				*				
12428				*	LEVEL 1 ROUTINE			
12429				*				
12430				*				
12431				*	COPY S(0) INTO S(1) OF USER AND CLEAR NTR FLAG			
12432				*				
12433	648A	06A0	690E		KEYENTER 3L POPREG			09415
12434	648E	06A0	6962		BL PSHREG			09416
12435	6492	06A0	6962		BL PSHREG			09417
12436	6496	E820	3348	0972	SOC CLINE15, RDTFLAG			09418
12437	649C	0380			RTWP			09419

DIGIT

09420

```

12439 *
12440 * PLACE ENTERED NUMBERS IN XSTACK AREA --- PROGRAM NUMBER
12441 *
12442 * SCAN XSTACK NUMBER AND DECODE IT, PLACING ONTO STACK --- PROGRAM SC
12443 *
12444 * R3 = EEX COUNTER
12445 * R2 = DECIMAL POINT COUNTER
12446 *
12447 * LEVEL 1 ROUTINES
12448 *
12449 649E KEY0 EQU * 09421
12450 649E KEY1 EQU * 09422
12451 649E KEY2 EQU * 09423
12452 649E KEY3 EQU * 09424
12453 649E KEY4 EQU * 09425
12454 649E KEY5 EQU * 09426
12455 649E KEY6 EQU * 09427
12456 649E KEY7 EQU * 09428
12457 649E KEY8 EQU * 09429
12458 649E KEY9 EQU * 09430
12459 649E KEY. EQU * 09431
12460 649E KEYDEC EQU * 09432
12461 649E KEYEEX EQU * 09433
12462 649E KEYCHS EQU * 09434
12463 649E E820 3348 D972 SOC CLINE16, RDTFLAG MAKE SURE BOTTOM LINE IS CHANGED 09435
12464 64A4 C26D 0012 MOV 18(R13), SOFT GET STACK POINTER 09436
12465 64A8 0649 DECT SOFT 09437
12466 64AA C650 3358 MOV C16, *SOFT 09438
12467 64AE 0649 DECT SOFT 09439
12468 64B0 C060 33F4 MOV CHARSTART, R1 09440
12469 64B4 A060 D98A A PROGRS, R1 09441
12470 64B8 C641 MOV R1, *SOFT 09442
12471 64BA C320 D98A MOV PROGRS, R12 IS THIS THE FIRST NUMERIC KEY? 09443
12472 64BE 162C JNE MORNUM NO, NUMERIC ENTRY IS ALREADY IN PROGRESS 09444
12473 64C0 8020 D955 382A C KEY, CHSKEY IS FIRST NUMERIC KEY 'CHS'? 09445
12474 64C6 160F JNE FRSTDIGT NO, THEN NUMERIC ENTRY IS STARTING 09446
12475 64C8 06A0 690E BL POPREG POP (X) OFF JSERSTACK 09447
12476 64CC 1303 JEQ CHSCNS (X) MUST BE A CONSTANT 09448
12477 64CE 04E0 D94A CLR FATAL SET ERROR FLAG IF (X) IS A WFM 09449
12478 64D2 1004 JMP CHSWFM 09450
12479 64D4 0501 CHSCNS NEG R1 NEGATE CONSTANT 09451
12480 64D6 1902 JND CHSWFM CHECK FOR $0000 09452
12481 64D8 0911 SRL R1, 1 SHIFT $0000 TO $4000 09453
12482 64DA 0582 INC R2 INCREMENT EXPONENT TO OFFSET MANTISSA/2 09454
12483 64DC 06A0 6962 CHSWFM BL PSHREG PUSH RESULT BACK ONTO USERSTACK 09455
12484 64E0 04E0 D93A CLR PROGRS CLEAR NUMERIC ENTRY IN PROGRESS FLAG 09456
12485 64E4 0300 RTWP 09457
12486 64E6 FRSTDIGT EQU * 09458
12487 64E6 0202 DA62 LI R2, TXTBUFR 09459
12488 64EA 06A0 6936 BL PSHSTK PUSH DUMMY ON STACK TO VOID X 09460
12489 64EE 0200 0018 LI R0, 24 09461
12490 64F2 C0C0 MOV R0, R3 SAVE BUFFER LENGTH 09462
12491 64F4 0659 DECT *SOFT 09463
12492 64F6 0649 SPACOUT DECT SOFT 09464
12493 64F8 C650 3432 MOV SPACE, *SOFT 09465
12494 64FC DCA0 3433 MOV3 SPACBYT, *R2+ 09466
12495 6500 0600 DEC R0 09467
12496 6502 16F9 JNE SPACOUT 09468
12497 6504 0649 DECT SOFT 09469

```

12498	6505	C643			MOV R3,*SOFT		09470
12499	6508	06A0	1362		BL TEXTS		09471
12500	650C	C660	33F4		MOV CHARSTART,*SOFT		09472
12501	6510	0659			DECT *SOFT		09473
12502	6512	04C2			CLR R2	DEC FLAG	09474
12503	6514	04C3			CLR R3	EEX FLAG	09475
12504	6516	04C0			CLR R0	EEX SIZE COUNT	09476
12505			6518	MORNUM	EQU *		09477
12506	6518	05A0	D93A		INC PROGRS	COJNT KEY	09478
12507	651C	8820	D966	3826	C KEY,EEXKEY		09479
12508	6522	132C			JEQ EEXCHK		09480
12509	6524	8820	D955	382A	C KEY,CHSKEY		09481
12510	652A	134A			JEQ CHSCHK		09482
12511	652C	C0C3			MOV R3,R3	EEX YET?	09483
12512	652E	1604			JNE *+10	IF SO,DONT CHECK SIZE	09484
12513	6530	8820	D93A	3350	C PROGRS,C12		09485
12514	6536	1518			JGT TOOMNY		09486
12515	6538	8820	D955	3828	C KEY,DECKEY	DECIMAL POINT?	09487
12516	653E	133A			JEQ DECCHK		09488
12517	6540	C0C3			MOV R3,R3	EEX YET?	09489
12518	6542	1318			JEQ STUFF	IF NOT --- INSERT NUMBER	09490
12519	6544	06A0	67D2		BL GETEEX		09491
12520	6548	C139			MOV *SOFT+,R4	EEX ADDRESS	09492
12521	654A	05C4			INCT R4		09493
12522	654C	C144			MOV R4,R5		09494
12523	654E	0585			INC R5		09495
12524	6550	D515			MOVB *R5,*R4		09496
12525			6552	STICKIT	EQU *		09497
12526	6552	C120	D966		MOV KEY,R4		09498
12527	6556	0A24			SLA R4,2		09499
12528	6558	0224	3836		AI R4,KEYTAB		09500
12529	655C	05C4			INCT R4		09501
12530	655E	C114			MOV *R4,R4		09502
12531	6560	D554			MOVB *R4,*R5		09503
12532	6562	1040			JMP FINEXCHS		09504
12533			6564	STUFF	EQU *		09505
12534	6564	0205	0A62		LI R5,TXTBUF2		09506
12535	6568	A160	D98A		A PROGRS,R5		09507
12536	656C	10F2			JMP STICKIT		09508
12537	656E	04E0	D94C	TOOMNY	CLR WARNING		09509
12538	6572	0620	D98A		DEC PROGRS		09510
12539	6576	0728	DACA		SET0 OKEY	MAKE SURE 'F' KEY WILL BE DISPLAYED IF RE-ENT	09511
12540	657A	1034			JMP FINEXCHS		09512
12541	657C	C0C3		EEXCHK	MOV R3,R3	EEX BEFORE?	09513
12542	657E	16F7			JNE TOOMNY		09514
12543	6580	C0E0	333A		MOV C1,R3	SET EEX FLAG	09515
12544	6584	0649			DECT SOFT		09516
12545	6586	0201	6530		LI R1,EEXMSG		09517
12546	658A	0204	0A52		LI R4,TXTBUF2		09518
12547	658E	A120	D98A		A PROGRS,R4		09519
12548	6592	0200	0004		LI R0,4		09520
12549	6596	8820	D93A	333A	C PROGRS,C1		09521
12550	659C	1504			JGT *+10		09522
12551	659E	0201	65AE		LI R1,EEXONE		09523
12552	65A2	0200	0006		LI R0,6		09524
12553	65A6	0D31			MOV3 *R1+,*R4+		09525
12554	65A8	0600			DEC R0		09526
12555	65AA	16FD			JNE *-4		09527
12556	65AC	1018			JMP FINEXCHS		09528

12557	65AE		31	EEXONE	FCC '1.'		09529
	65AF		2E				
12558	65B0		45	EEXMSG	FCC 'E+00'		09538
	65B1		2B				
	65B2		30				
	65B3		30				
12559	65B4	C0C3		DECCHK	MOV R3,R3	EEX KEY YET?	09531
12560	65B6	160B			JNE TOOMNY	NO DEC AFTER EEX ALLOWED	09532
12561	65B8	C082			MOV R2,R2	DECIMAL POINT BEFORE?	09533
12562	65BA	16D9			JNE TOOMNY		09534
12563	65BC	0582			INC R2	OK --- FLAG IT	09535
12564	65BE	10D2			JMP STUFF		09536
12565			65C0	C4SCHK	EQU *		09537
12566	65C0	0620	D9BA		DEC PROGRS		09538
12567	65C4	0204	DA62		LI R4, TXTBUFR		09539
12568	65C8	C0C3			MOV R3,R3		09548
12569	65CA	1328			JEQ CHKSIGN		09541
12570	65CC	06A0	67D2		BL GETEEX		09542
12571	65D0	C139			MOV *SOFT+,R4		09543
12572	65D2	0584			INC R4		09544
12573	65D4	9814	33EC		CB *R4, NEGMNU		09545
12574	65D8	1603			JNE POSEXHIT		09546
12575	65DA	0520	33EE		MOVB POSMNU,*R4		09547
12576	65DE	1002			JMP FINEXCHS		09548
12577			65E0	POSEXHIT	EQU *		09549
12578	65E0	0520	33EC		MOVB NEGMNJ,*R4		09550
12579			65E4	FINEXCHS	EQU *		09551
12580	65E4	C1E0	D9C6		MOV REALTIME,R7	DO NOT PUT DIGITS ON SCREEN IF IN REALTIME MODE	09552
12581	65E8	1510			JGT NUMRETRN		09553
12582	65EA	130F			JEQ NUMRETRN		09554
12583	65EC	C660	3358		MOV C16,*SOFT		09555
12584	65F0	0649			DECT SOFT		09556
12585	65F2	C650	33F4		MOV CHARSTART,*SOFT		09557
12586	65F6	0619			DEC *SOFT		09558
12587	65F8	0649			DECT SOFT		09559
12588	65FA	0208	DA62		LI R0, TXTBUFR		09560
12589	65FE	C640			MOV R0,*SOFT		09561
12590	6600	0649			DECT SOFT		09562
12591	6602	C660	335A		MOV C17,*SOFT		09563
12592	6606	06A0	136C		BL TEXT		09564
12593			660A	NUMRETRN	EQU *		09565
12594	660A	0380			RTWP		09566
12595			650C	CHKSIGN	EQU *		09567
12596	660C	9814	33EC		CB *R4, NEGMNU		09568
12597	6610	16E7			JNE POSEXHIT		09569
12598	6612	0520	3433		MOVB SPACBYT,*R4		09570
12599	6616	10E5			JMP FINEXCHS		09571

12601						*****		
12602						*		
12603						*	SCAN INPUT STRING REQUESTED	
12604						*		
12605						*	CALL WITH R0 POINTING TO ASCII NUMERIC STRING	
12606						*	RETURNS WITH R0=NEXT CHARACTER ADDRESS IN STRING (1ST NON-DIGIT)	
12607						*	R1,R2= FP NUMBER CONVERTED	
12608						*		
12609						*	INCLUDES CONVERSION OF SCALING SYMBOLS IN SYMTAB	
12610						*		
12611						*	LEVEL 3 ROUTINE	
12612						*		
12613						*	REGISTER USAGE ---	
12614						*	R5 = CURRENT ADDRESS BEING SCANNED	
12615						*		
12616	6618		DBA0		SCANIN	WORD	WPLVL3	09573
12617	661A		661C			WORD	*+2	09574
12618	661C	C15D				MOV	*R13,R5	09575
12619	661E	C26D	0012			MOV	18(R13),SOFT GET SOFTSTACK POINTER	09576
12620	6622	C820	333A	D9BA		MOV	C1,PROGRS NEG FLAGS	09577
12621	6628	04C1				CLR	R1 R1,R2 - DOUBLE PRECISION INTEGER	09578
12622	662A	04C2				CLR	R2 VALUE	09579
12623	662C	04C3				CLR	R3 R3 - NORMALIZATION AMOUNT	09580
12624	662E	04C4				CLR	R4 R4 - DECIMAL POINT FLAG	09581
12625	6630	04C7				CLR	R7 R7 - EXPONENT	09582
12626	6632	04C8				CLR	R8 R8 - COUNT OF SIGNIFICANT DIGITS	09583
12627	6634	9815	3433			CB	*R5,SPACBYT ALLOW FIRST CHAR TO BE SPACE	09584
12628	6638	1601				JNE	*+4	09585
12629	663A	0585				INC	R5 SKIP PAST SPACE	09586
12630	663C	C185				MOV	R5,R5 SAVE THE FIRST CHARACTER LOCATION	09587
12631	663E	D315				MOV8	*R5,R12	09588
12632	6640	980C	33EE			CB	R12,POSMNU CHECK FOR + SIGN	09589
12633	6644	1306				JEQ	GOTSSN	09590
12634	6646	980C	33EC			CB	R12,NEGMNU CHECK FOR - SIGN	09591
12635	664A	1609				JNE	NXTCK	09592
12636	664C	E820	3348	D9BA		SOC	C8,PROGRS SET NEG MANTISSA FLAG	09593
12637	6652	0585			GOTSSN	INC	R5 GET PAST SIGN	09594
12638	6654	9815	33AC			CB	*R5,NULL ALLOW SPACE TO FOLLOW SIGN	09595
12639	6658	1601				JNE	*+4	09596
12640	665A	0585			PASSIT	INC	R5	09597
12641	665C	D315				MOV8	*R5,R12	09598
12642	665E	980C	33EA		NXTCK	CB	R12,EEXMNU CHECK IF INPUT IS E	09599
12643	6652	132A				JEQ	SWCHIT	09600
12644	6664	980C	33EB			CB	R12,DECMNU CHECK FOR DECIMAL POINT	09601
12645	6668	1603				JNE	KEEPON	09602
12646	666A	E120	3340			SOC	C4,R4 SET DECIMAL POINT FLAG	09603
12647	666E	10F5				JMP	PASSIT	09604
12648	6670	098C			KEEPON	SRL	R12,8	09605
12649	6672	6320	342C			S	NUMMNU,R12 CONVERT TO DECIMAL FROM ASCII	09606
12650	6676	1141				JLT	HOWBIG CHECK TO SEE RESULT IN RANGE	09607
12651	6678	028C	0009			CI	R12,9 0 - 9	09608
12652	667C	153E				JGT	HOWBIG	09609
12653	667E	C208				MOV	R8,R8 FIRST SIGNIFICANT DIGIT FOUND?	09610
12654	6680	1502				JGT	*+6	09611
12655	6682	C30C				MOV	R12,R12	09612
12656	6684	1309				JEQ	NOSIG IGNORE LEADING ZEROES	09613
12657	6686	0288	0009			CI	R8,9 NINE SIGNIFICANT DIGITS FOUND?	09614
12658	668A	1105				JLT	STILSIG	09615
12659	668C	2120	3340			COB	C4,R4 ALREADY HAVE NINE DIGITS	09616

12660	6690	1301			JEQ	**4			09617
12661	6692	0583			INC	R3	NEED TO KEEP TRACK OF SIZE OF NUMBER		09618
12662	6694	10E2			JMP	PASSIT			09619
12663	6696	0588			STI_SIG INC	R8	COUNT THE DIGIT		09620
12664	6698	2120	3340		NDSIG COC	C4,R4	CHECK DECIMAL POINT FLAG		09621
12665	669C	1601			JNE	**4			09622
12666	669E	0603			DEC	R3	KEEP TRACK OF DIGITS TO RIGHT OF DEC POINT		09623
12667	66A0	C282			MOV	R2,R10			09624
12668	66A2	3860	334C		MPY	C10,R1	MULTIPLY OLD BY 10		09625
12669	66A6	C042			MOV	R2,R1	DOUBLE PRECISION INTEGER MULTIPLY		09626
12670	66A8	3AA0	334C		MPY	C10,R10			09627
12671	66AC	C08B			MOV	R11,R2			09628
12672	66AE	A04A			A	R10,R1			09629
12673	66B0	A08C			A	R12,R2	ADD IN NEW DIGIT		09630
12674	66B2	1701			JNC	**4			09631
12675	66B4	0581			INC	R1			09632
12676	66B6	10D1			JMP	PASSIT			09633
12677	66B8	8185			SMCHIT C	R5,R5	DEFAULT MANT. = 1 IF ONLY E		09634
12678	66BA	1601			JNE	MANTEXP			09635
12679	66BC	0582			INC	R2			09636
12680	66BE	0585			MANTEXP INC	R5	HANDLE EXPONENT		09637
12681	66C0	D315			MOV8	*R5,R12			09638
12682	66C2	980C	33EC		C3	R12,NEGMNU	CHECK FOR NEG. EXPONENT		09639
12683	66C6	1604			JNE	NOXNEG			09640
12684	66C8	E820	3358	D93A	SOC	C16,PRJGRS	SET NEGATIVE EXPONENT FLAG		09641
12685	66CE	1003			JMP	INNUM			09642
12686	66D0	980C	33EE		NOXNEG C3	R12,POSMNU	CHECK FOR POS. EXPONENT		09643
12687	66D4	1602			JNE	EXPNUM1			09644
12688	66D6	0585			INNUM INC	R5			09645
12689	66D8	D315			MOV8	*R5,R12			09646
12690	66DA	D98C			EXPNUM1 SRL	R12,8			09647
12691	66DC	5320	342C		S	NUMMNU,R12	CONVERT FROM ASCKK TO DECIMAL		09648
12692	66E0	110C			JLT	HOWBIG	CHECK TO SEE IF IN RANGE 0 - 9		09649
12693	66E2	028C	0009		CI	R12,9			09650
12694	66E6	1509			JGT	HOWBIG			09651
12695	66E8	39E0	334C		MPY	C10,R7	MULTIPLY OLD BY 10		09652
12696	66EC	C1C7			MOV	R7,R7	IF R7 <> 0 THEN EXP. TOO LARGE		09653
12697	66EE	1602			JNE	TOOBIG			09654
12698	66F0	A20C			A	R12,R8	ADD IN DIGIT		09655
12699	66F2	1701			JNC	**4			09656
12700	66F4	0708			TOOBIG SETJ	R8			09657
12701	66F6	C1C8			MOV	R8,R7			09658
12702	66F8	10EE			JMP	INNUM	GET NEXT CHARACTER		09659
12703	66FA	C107			HOWBIG MOV	R7,R4	SAVE EXPONENT		09660
12704	66FC	06A0	7582		BL	DINTFP	CONVERT DOUBLE PRECISION INTEGER TO FP		09661
12705	6700	0649			DECT	SOFT	PUSH VALUE ONTO STACK		09662
12706	6702	C642			MOV	R2,*SOFT			09663
12707	6704	0649			DECT	SOFT			09664
12708	6706	C641			MOV	R1,*SOFT			09665
12709	6708	C220	D98A		MOV	PROGRS,R8			09666
12710	670C	2220	3358		COC	C16,R8	CHECK FOR NEGATIVE EXPONENT		09667
12711	6710	1601			JNE	**4			09668
12712	6712	0504			NEG	R4			09669
12713	6714	A103			A	R3,R4			09670
12714	6716	1337			JEQ	ALLDON			09671
12715	6718	1508			JGT	EXPOS			09672
12716	671A	0504			NEG	R4			09673
12717	671C	E820	3358	D93A	SOC	C16,PRJGRS	SET NEGATIVE EXPONENT FLAG		09674
12718	6722	0284	006E		CI	R4,110	CHECK FOR EXPONENT OUT OF RANGE		09675

12719	6726	1113		JLT	EXPOK		09676	
12720	6728	04F9		CLR	*SOFT+		09677	
12721	672A	04D9		CLR	*SOFT		09678	
12722	672C	0649		DECT	SOFT		09679	
12723	672E	102B		JMP	ALLDON		09680	
12724	6730	0284	0054	EXPJS	CI	R4,100	CHECK FOR EXPONENT OUT OF RANGE	09681
12725	6734	110C		JLT	EXPOK		09682	
12726	6736	0204	0063	LI	R4,99		LOAD WITH LARGEST EXPONENT	09683
12727	673A	2220	3348	CDC	C8,R8		CHECK FOR NEGATIVE MANTISSA	09684
12728	673E	1604		JNE	MAXPOS		09685	
12729	6740	0203	8000	LI	R3,\$8000		PUSH MAX NEGATIVE NUMBER	09686
12730	6744	C643		MOV	R3,*SOFT		ONTO SOFTSTACK	09687
12731	6746	1003		JMP	EXPOK		09688	
12732	6748	0203	7FFF	MAXPOS	LI	R3,\$7FFF	PUSH MAX POSITIVE NUMBER	09689
12733	674C	C643		MOV	R3,*SOFT		ONTO SOFTSTACK	09690
12734	674E	C220	D9BA	EXPJK	MOV	PROGRS,R8		09691
12735	6752	04C3		CLR	R3		09692	
12736	6754	3CE0	334C	DIV	C10,R3		DIVIDE EXPONENT BY 10	09693
12737	6758	2220	3358	CDC	C16,R8		09694	
12738	675C	1602		JNE	*+6		09695	
12739	675E	0503		NEG	R3		NEGATIVE EXPONENT	09696
12740	6760	0504		NEG	R4		09697	
12741	6762	0A24		SLA	R4,2		CALCULATE ONE'S TABLE INDEX	09698
12742	6764	1307		JEQ	TENNRH		09699	
12743	6766	C054	3522	MOV	ONETBL(R4),R1		ONES MANTISSA	09700
12744	676A	05C4		INCT	R4		09701	
12745	676C	C0A4	3522	MOV	ONETBL(R4),R2		ONES EXPONENT	09702
12746	6770	0420	70AA	BLW ^P	FPMPY		REMOVE ONES EXPONENT	09703
12747	6774	0A23		TENNRH	SLA	R3,2	GET ONES TABLE INDEX	09704
12748	6776	1307		JEQ	ALLDON		CHECK FOR EXPONENT = 0	09705
12749	6778	C063	3572	MOV	TENCTR(R3),R1		TENS EXPONENT	09706
12750	677C	05C3		INCT	R3		09707	
12751	677E	C0A3	3572	MOV	TENCTR(R3),R2		TENS MANTISSA	09708
12752	6782	0420	70AA	BLW ^P	FPMPY		REMOVE TENS EXPONENT	09709
12753	6786	C1C5		ALLDON	MOV	R5,R7	SAVE CURRENT CHARACTER POSITION	09710
12754	6788	9837	33AC	CB	*R7+,NJLL		IGNORE SKIP CHARACTERS	09711
12755	678C	13FD		JEQ	*-4		09712	
12756	678E	0607		DEC	R7		09713	
12757	6790	0204	3F3A	LI	R4,SYMTAB		CHECK FOR SCALING SYMBOLS	09714
12758	6794	D057		MOV3	*R7,R1		R4 CONTAINS THE TABLE POINTER	09715
12759	6796	0981		SRL	R1,8		09716	
12760	6798	8501		ALLDON2	C	R1,*R4	CHECK AGAINST TABLE VALUE	09717
12761	679A	1306		JEQ	HITSYMB		09718	
12762	679C	8814	3F6A	C	*R4,TABEND		NO SCALE IF END OF TABLE	09719
12763	67A0	130A		JEQ	NOSYMB		09720	
12764	67A2	0224	0005	AI	R4,6		09721	
12765	67A6	10F8		JMP	ALLDON2		09722	
12766	67A8	05C4		HITSYMB	INCT	R4	09723	
12767	67AA	C074		MOV	*R4+,R1		GET MANTISSA VALUE	09724
12768	67AC	C8B4		MOV	*R4+,R2		GET EXPONENT VALUE	09725
12769	67AE	0420	70AA	BLW ^P	FPMPY		09726	
12770	67B2	0507		INC	R7		POINT TO NEXT CHARACTER	09727
12771	67B4	C147		MOV	R7,R5		GET NEW CHARACTER POSITION	09728
12772	67B6	C079		NOSYMB	MOV	*SOFT+,R1	STORE BINARY MANTISSA IN R1	09729
12773	67B8	C0B9		MOV	*SOFT+,R2		STORE BINARY EXPONENT IN R2	09730
12774	67BA	2220	3348	CDC	C8,R8		CHECK FOR NEG EXPONENT	09731
12775	67BE	1601		JNE	*+4		09732	
12776	67C0	0501		NEG	R1		09733	
12777	67C2	C745		MOV	R5,*R13		SAVE NEXT CHARACTER POSITION	09734

SCAN NUMERIC INPUT AND CREATE FLOATING POINT NUMBER

09572

12778	67C4	04E0	D98A	CLR	PROGRS		09735
12779	67C8	C100		MOV	R13,R4		09736
12780	67CA	05C4		INCF	R4		09737
12781	67CC	C001		MOV	R1,*R4+	STORE RESULT IN CALLING WORKSPACE	09738
12782	67CE	C002		MOV	R2,*R4+		09739
12783	67D0	0380		RTWP			09740

GETEEK

09741

12785				*				
12786				*	FIND 'E' IN NUMERIC STRING IN TXTBUFR			
12787				*				
12788				*	RETJRN POSITION IN STACK			
12789				*				
12790				*	LEVEL 5			
12791				*				
12792				*	STACK OPERATIONS - PUSHES 1			
12793				*				
12794				*	DESTROYS R7			
12795				*				
12796			67D2		GETEEK EQU *			09742
12797	67D2	0207	0A63		LI R7,TXTBUFR+1			09743
12798			67D6		FINDEXSGN EQU *			09744
12799	67D6	9817	33EA		CB *R7,EEXMNU	GOT AN 'E'?		09745
12800	67DA	1302			JEQ EEXHIT			09746
12801	67DC	0587			INC R7	CHECK THE ENTIRE TXTBUFR		09747
12802	67DE	10FB			JMP FINDEXSGN			09748
12803	67E0	0649		EEXHIT	DEGT SOFT			09749
12804	67E2	0647			MOV R7,*SOFT	PUSH POSITION OF 'E' ONTO STACK		09750
12805	67E4	045B			B *R11			09751

CALCULATE SCALES

09753

```

12808 *****
12809 **
12810 ** CALCULATES: ZERO REFERENCE **
12811 ** VERTICAL SCALE **
12812 ** HORIZONTAL SCALE **
12813 ** VERTICAL CURSOR POSITION **
12814 ** HORIZONTAL CURSOR POSITION **
12815 **
12816 ** LEVEL 3 ROUTINE **
12817 **
12818 ** INPUT: WFM # ON SOFTSTACK **
12819 ** OUTPUT: R1,R2 GET FP VALUE **
12820 **
12821 ** STACK OPERATIONS: **
12822 ** SOFTSTK: POPS 1 **
12823 **
12824 *****
12825
12826 *****
12827 * ZERO REFERENCE = VOFFAB * -20
12828 *****
12829 67E6 DBA0 ZROREF WORD WPLVL3 LEVEL 3 ROUTINE 09754
12830 67E8 67EA WORD *+2 09755
12831 67EA C0E0 D95E MOV OPWFMH,R3 GET VERTICAL OFFSET IN DIVISIONS 09756
12832 67EE A0E0 3348 A VOFFAB,R3 09757
12833 67F2 C26D 0012 MOV 18(R13),SOFT 09758
12834 67F6 0649 DECT SOFT 09759
12835 67F8 0409 CLR *SOFT 09760
12836 67FA 0649 DECT SOFT 09761
12837 67FC C653 MOV *R3,*SOFT 09762
12838 67FE 0201 8000 LI R1,$8000 LOAD R1,R2 WITH -20 09763
12839 6802 0202 0005 LI R2,5 09764
12840 6806 0420 70AA BLWP FMPY 09765
12841 680A CB79 0002 MOV *SOFT+,2(R13) 09766
12842 680E CB59 0004 MOV *SOFT,4(R13) 09767
12843 6812 0380 RTWD 09768
12844
12845 *****
12846 * VERTICAL SCALE = VEXP / 20
12847 *****
12848 6814 DBA0 VRTSCL WORD WPLVL3 LEVEL 3 ROUTINE 09769
12849 6816 6818 WORD *+2 09770
12850 6818 C0E0 D95E MOV OPWFMH,R3 GET THE START OF THE WFM'S HEADER INFO 09771
12851 681C A0E0 3338 A VEXP,R3 INDEX INTO THE VERTICAL EXPONENT 09772
12852 6820 C26D 0012 MOV 18(R13),SOFT 09773
12853 6824 0649 DECT SOFT 09774
12854 6826 C0B3 MOV *R3+,R2 09775
12855 6828 C653 MOV *R3,*SOFT 09776
12856 682A 0649 DECT SOFT 09777
12857 682C C642 MOV R2,*SOFT 09778
12858 682E 0201 5000 LI R1,$5000 09779
12859 6832 0202 0005 LI R2,5 09780
12860 6836 0420 7058 BLWP FPDIV 09781
12861 683A CB79 0002 MOV *SOFT+,2(R13) 09782
12862 683E CB59 0004 MOV *SOFT,4(R13) 09783
12863 6842 0380 RTWD 09784
12864
12865 *****
12866 * HORIZONTAL SCALE = RESOLV * HEXP / 10

```

12867				*****					
12868	6844		DBA0	HRZSCL	WORD	HPLV,3	LEVEL 3 ROUTINE		09785
12869	6846		6848		WORD	*+2			09786
12870	6848	C26D	0012		MOV	18(R13),SOFT	GET THE SOFTWARE STACK POINTER		09787
12871	684C	C0E0	D95E		MOV	OPWF4H,R3			09788
12872	6850	A0E0	3340		A	HEXP,R3			09789
12873	6854	0229	FFFC		AI	SOFT,-4	PUT THE HEXP ON THE STACK		09790
12874	6858	3E73			MOV	*R3+,*SOFT+			09791
12875	685A	C653			MOV	*R3,*SOFT			09792
12876	685C	0649			DECT	SOFT			09793
12877	685E	C050	D970		MOV	RESOLV,R1	GET THE RESOLUTION IN R1,R2		09794
12878	6862	06A0	757E		BL	INT2FP			09795
12879	6866	0420	70AA		BLWP	FPMPY	RESOLV * HEXP		09796
12880	686A	C060	33FE		MOV	FP10H,R1			09797
12881	686E	C0A0	3340		MOV	FP10E,R2			09798
12882	6872	0420	7050		BLWP	FPDIV	RESOLV * HEXP / 10		09799
12883	6876	C879	0002		MOV	*SOFT+,2(R13)			09800
12884	687A	C879	0004		MOV	*SOFT+,4(R13)			09801
12885	687E	0380			RTWP				09802
12886									

12887				*****					
12888				*	CALCULATE VERTICAL CURSOR POSITION OR DELTA POSITION				
12889				*****					
12890	6880		DBA0	VCRPOS	WORD	HPLVL3	LEVEL ROUTINE		09803
12891	6882		6884		WORD	*+2			09804
12892	6884	C26D	0012		MOV	18(R13),SOFT	GET THE SOFTSTACK POINTER		09805
12893	6888	0649			DECT	SOFT			09806
12894	688A	04D9			CLR	*SOFT	CREATE EXPONENT OF ZERO		09807
12895	688C	0649			DECT	SOFT			09808
12896	688E	C060	D95C		MOV	OPWFMD,R1	GET THE WAVEFORM DATA ADDRESS		09809
12897	6892	C0A0	D954		MOV	CURSOR,R2	DETERMINE HOW MANY CURSORS ARE BEING		09810
12898	6896	0602			DEC	R2	DISPLAYED (1,2)		09811
12899	6898	1508			JST	VCR2			09812
12900	689A	C0A0	D956	VCR1	MOV	CURS1,R2	GET THE VERTICAL VALUE OF THE CURSOR		09813
12901	689E	0A12			SLA	R2,1			09814
12902	68A0	A042			A	R2,R1			09815
12903	68A2	C651			MOV	*R1,*SOFT			09816
12904	68A4	C060	3348		MOV	VOFFAB,R1			09817
12905	68A8	A060	D95E		A	OPWF4H,R1	ADD THE VERTICAL OFFSET		09818
12906	68AC	A651			A	*R1,*SOFT			09819
12907	68AE	1008			JMP	VFPMOV			09820
12908	68B0	C0A0	D958	VCR2	MOV	CURS2,R2	GET THE DIFFERENCE IN THE VERTICAL		09821
12909	68B4	0A12			SLA	R2,1	POSITIONS OF THE TWO CURSORS		09822
12910	68B6	A042			A	R2,R1			09823
12911	68B8	C651			MOV	*R1,*SOFT			09824
12912	68BA	C060	D956		MOV	CURS1,R1			09825
12913	68BE	0A11			SLA	R1,1			09826
12914	68C0	A060	D95C		A	OPWF4D,R1			09827
12915	68C4	6651			S	*R1,*SOFT			09828
12916	68C6	C0A0	3338	VFPMOV	MOV	VEXP,R2			09829
12917	68CA	A0A0	D95E		A	OPWF4H,R2	MULTIPLY VERTICAL VALUE BY 'VEXP'		09830
12918	68CE	C072			MOV	*R2+,R1			09831
12919	68D0	C092			MOV	*R2,R2			09832
12920	68D2	0420	70AA		BLWP	FPMPY			09833
12921	68D6	C879	0002		MOV	*SOFT+,2(R13)	POP THE RESULT OFF THE STACK		09834
12922	68DA	C879	0004		MOV	*SOFT+,4(R13)			09835
12923	68DE	0380			RTWP				09836
12924									
12925				*****					

CALCULATE SCALES

09753

```

12926          * CALCULATE HORIZONTAL CURSOR POSITION OR DELTA POSITION
12927          *****
12928  68E0          DBA0          HCRPOS  WORD  WPLVL3          LEVEL 3 ROUTINE          09837
12929  68E2          68E4          HCRD    WORD  *+2          09838
12930  68E4          C260          MOV    18(R13),SOFT  09839
12931  68E8          C060          MOV    CURSOR,R1    CHECK TO SEE HOW MANY CURSORS ARE
12932  68EC          0601          DEC    R1            BEING DISPLAYED (1,2)          09841
12933  68EE          1503          JGT   HCR2          09842
12934  68F0          C050          D956          HCR1    MOV    CURS1,R1    OR THE DIFFERENCE BETWEEN THE TWO IF
12935  68F4          1004          JMP   HCR3          BOTH ARE DISPLAYED          09844
12936  68F6          C050          D958          HCR2    MOV    CURS2,R1    09845
12937  68FA          6060          D956          S      CURS1,R1    09846
12938  68FE          0649          HCR3    DECT  SOFT    USE ONE LETTER EQUIVALENT EXPONENTS
12939  6900          C660          3356          MOV    C15,*SOFT   SET THE MANTISSA'S EXPONENT TO 15  09848
12940  6904          0649          DECT  SOFT    09849
12941  6906          C641          MOV    R1,*SOFT   09850
12942  6908          C0A0          3340          MOV    HEXP,R2    HCR = ELEMENT# * HEXP          09851
12943  690C          103E          JMP   VFPMOV+4    09852

```

12945				*				
12946				*	POP USER STACK TO R0,R1,R2			
12947				*				
12948				*	LEVEL 5 ROUTINE			
12949				*				
12950				*	INPUT ON USER STACK			
12951				*	OUTPUT IN R0,R1,R2			
12952				*				
12953				*	EQ STATUS SET IF FP POPPED			
12954				*				
12955				*	DESTROYS R7,R8,R12			
12956				*				
12957	690E	C2A0	33D6	POPREG	MOV XSTK,USER			09854
12958	6912	C03A			MOV *USER+,R0			09855
12959	6914	C07A			MOV *USER+,R1			09856
12960	6916	C09A			MOV *USER,R2			09857
12961	6918	C320	33D4	POPSTK	MOV WSTK,R12			09858
12962	691C	C220	33D6		MOV XSTK,R8			09859
12963	6920	0207	000F		LI R7,15			09860
12964	6924	CF38			MOV *R8+,*R12+			09861
12965	6926	0607			DEC R7			09862
12966	6928	16FD			JNE *-4			09863
12967	692A	C2A0	33D6		MOV XSTK,USER			09864
12968	692E	C1E0	33D4		MOV WSTK,R7	SET EQ FLAG FOR FP		09865
12969	6932	C107			MOV *R7,R7			09866
12970	6934	045B			B *R11			09867

12972				*				
12973				*	PUSH ELEMENT ONTO USERS STACK			
12974				*				
12975				*	LEVEL 5 ROUTINE			
12976				*				
12977				*	INPUT	R0 POINTS TO STACK-TYPE ELEMENT TO PUSH		
12978				*	OUTPUT	USER ADJUSTED FOR PUSHED ELEMENT		
12979				*		R0 IS UNCHANGED		
12980				*				
12981				*	INPUT R0,USER			
12982				*	OUTPUT R0,USER			
12983				*	DESTROYS R7,R8,R12			
12984				*				
12985	6936	C320	3302		PSHSTK	MOV TOPUSE,R12		09869
12986	693A	C220	330C			MOV TSTK,R8		09870
12987	693E	0648				DECT R8		09871
12988	6940	0207	000C			LI R7,12		09872
12989	6944	C718			PSHLOP	MOV *R8,*R12		09873
12990	6946	0648				DECT R8		09874
12991	6948	064C				DECT R12		09875
12992	694A	0607				DEC R7		09876
12993	694C	16FB				JNE PSHLOP		09877
12994	694E	C300				MOV R0,R12		09878
12995	695J	C2A0	3305			MOV XSTK,USER		09879
12996	6954	CEBC				MOV *R12+,*USER+		09880
12997	6956	CEBC				MOV *R12+,*USER+		09881
12998	6958	C6BC				MOV *R12+,*USER		09882
12999	695A	C2A0	3306		DONE3	MOV XSTK,USER		09883
13000	695E	C31A				MOV *USER,R12	SET EQ	09884
13001	6960	045B				B *R11		09885
13002	6962	0649			PSHREG	DECT SOFT		09886
13003	6964	C64B				MOV R11,*SOFT		09887
13004	6966	06A0	6936			BL PSHSTK		09888
13005	696A	CE80				MOV R0,*USER+		09889
13006	696C	CE81				MOV R1,*USER+		09890
13007	696E	C682				MOV R2,*USER		09891
13008	6970	C2F9				MOV *SOFT+,R11		09892
13009	6972	10F3				JMP DONE3		09893

ZEROW0

09894

13011	6974	C1E0	DAD2	ZEROW0	MOV	W0ADD,R7	ADDRESS OF W0'S DATA	09895
13012	6978	C220	D970		MOV	RESOLV,R8	CURRENT WAVEFORM RESOLUTION	09896
13013	697C	04F7			CLR	*R7+	ZERO W0	09897
13014	697E	0608			DEC	R8		09898
13015	6980	15FD			JGT	*-4		09899
13016	6982	045B			B	*R11		09900

ADRWFH

09901

```

13018 *
13019 *   CALCULATES ADDRESS OF HEADER AND DATA FOR WFM #
13020 *
13021 *   LEVEL 5 ROUTINE
13022 *
13023 *   INPUT:  (ADRWFH) R1 HAS WFM NUMBER
13024 *           SOFT HAS SOFTWARE STACK POINTER
13025 *           USER HAS USER STACK POINTER
13026 *   OUTPUT: R1 HAS WFM NUMBER
13027 *           1ST POP OF SOFT HAS HEADER ADD.
13028 *           2ND POP OF SOFT HAS DATA ADD.
13029 *
13030 *   INPUT R1,SOFT,USER
13031 *   OUTPUT SOFT,USER
13032 *   DESTROYS R1,R7,R8,R12
13033 *
13034 6984 C1E0 D970 ADRWFH MOV RESOLV,R7                                09902
13035 6988 C301      MOV R1,R12      WFM #                                09903
13036 698A 0A1C      SLA R12,1      2WFM#                                09904
13037 698C 39CC      MPY R12,R7     N X 2WFM#                             09905
13038 698E A220 DA06 A WFMBAS,R8    BASE + N X 2WFM#                     09906
13039 6992 0649      DECT SOFT                                           09907
13040 6994 C648      MOV R8,*SOFT    PUSH WFM DATA ADDRESS             09908
13041 6996 C1C1      MOV R1,R7                                           09909
13042 6998 0587      INC R7        WFM# + 1                               09910
13043 699A 0A57      SLA R7,5      32(WFM# + 1)                          09911
13044 699C C220 DA08 MOV TOPHED,R8                                           09912
13045 69A0 6207      S R7,R8      TOP - 64(WFM# + 1)                    09913
13046 69A2 0649      DECT SOFT                                           09914
13047 69A4 C648      MOV R8,*SOFT    PUSH HEADER ADDRESS               09915
13048 69A6 045B      B *R11

```

```

13050 *
13051 *      TRANSFER WAVEFORM HEADER
13052 *
13053 *      INPUT --- (XFRHEAD)
13054 *      1ST STACK POP IS ORIGIN WFM
13055 *      2ND STACK POP IS DESTINATION WFM
13056 *      (OPWH2W0)
13057 *      NONE - TRANSFERS OPWFM TO W0
13058 *
13059 *      LEVEL 4 ROUTINE
13060 *
13061 69A8      DBC0      OPWH2W0 WORD WPLVL4      09918
13062 69AA      69AC      WORD *+2      09919
13063 69AC      C26D      0012      MOV 18(R13),SOFT GET SOFTWARE STACK      09920
13064 69B0      0649      DECT SOFT      09921
13065 69B2      04D9      CLR *SOFT      TRANSFER OPWFM HEADER TO W0      09922
13066 69B4      0649      DECT SOFT      09923
13067 69B6      C660      D95A      MOV OPWFM,*SOFT      09924
13068 69BA      1004      JMP XFRH      09925
13069 69BC      DBC0      XFRHEAD WORD WPLVL4      09926
13070 69BE      69C0      WORD *+2      09927
13071 69C0      C26D      0012      MOV 18(R13),SOFT GET SOFTWARE STACK      09928
13072 69C4      C079      XFRH      MOV *SOFT+,R1      ORIGIN WFM #      09929
13073 69C6      06A0      6984      BL ADRWFM      09930
13074 69CA      C0F9      MOV *SOFT+,R3      09931
13075 69CC      C079      MOV *SOFT+,R1      09932
13076 69CE      C079      MOV *SOFT+,R1      DESTINATION WFM #      09933
13077 69D0      06A0      6984      BL ADRWFM      09934
13078 69D4      C139      MOV *SOFT+,R4      DESTINATION HEADER      09935
13079 69D6      C079      MOV *SOFT+,R1      DEST. DATA -- DISPOSE OF      09936
13080 69D8      C044      MOV R4,R1      09937
13081 69DA      A050      3354      A DISPLA,R1      09938
13082 69DE      C091      MOV *R1,R2      OLD DISPLAY STATUS OF DESTINATION      09939
13083 69E0      0205      0010      LI R5,16      09940
13084 69E4      C033      MOV *R3+,*R4+      XFER HEADER      09941
13085 69E6      0605      DEC R5      09942
13086 69E8      16FD      JNE *-4      09943
13087 69EA      C442      MOV R2,*R1      MAINTAIN OLD DISPLAY STATUS      09944
13088 69EC      CB49      0012      MOV SOFT,18(R13) SET CALLER STACK TO POPPED LEVEL      09945
13089 69F0      0380      RTW      09946
    
```

HORDLTA --- FIND HORIZONTAL VALUE OF POINT SPREAD

09947

```
13091 *
13092 *   FIND HORIZONTAL VALUE OF CURSOR OR POINT SPREAD ON OPWFH
13093 *
13094 *   INPUT ---  R1 = POINT SPREAD OR CURSOR LOCATION
13095 *   OUTPUT --- R1,R2 = FLOATING POINT HORIZONTAL VALUE OF POINT SPREAD
13096 *
13097 *   LEVEL 5 ROUTINE
13098 *
```

```
13099 69F2 0649   HORDLTA DECT SOFT                                09948
13100 69F4 C660 3356   MOV C15,*SOFT   SET MANTISSA TO 15                            09949
13101 69F8 0649   DECT SOFT                                           09950
13102 69FA C641   MOV R1,*SOFT                                         09951
13103 69FC C0A0 095E   MOV OPWFMH,R2                                        09952
13104 6A00 A0A0 3340   A HEXP,R2                                           09953
13105 6A04 C072   MOV *R2+,R1                                          09954
13106 6A06 C092   MOV *R2,R2                                          09955
13107 6A08 0420 70AA   BLWP FPMPY                                          09956
13108 6A0C C079   MOV *SOFT+,R1                                       09957
13109 6A0E C8B9   MOV *SOFT+,R2                                       09958
13110 6A10 045B   B *R11                                             09959
```

```

13112 *
13113 *   FIND A WFM CROSSING OF A SPECIFIED VALUE
13114 *
13115 *   LEVEL 3 ROUTINE
13116 *
13117 *   SEARCH IS ON OPWFM
13118 *
13119 *   R0 --- ADDRESS TO START SEARCH
13120 *   R1 --- WINDOW WIDTH OF SEARCH
13121 *   R2 --- WFM VALUE TO FIND CROSSING OF
13122 *   CROSFLAG = 1 TO SEARCH FROM INPUT POINT OR ANY OTHER VALUE
13123 *   TO SEARCH FROM MIDWAY BETWEEN STARTING AND NEXT POINT
13124 *
13125 *   RETURNS ---
13126 *   R0 POINTS TO ABSOLUTE LOCATION OF CROSSPOINT
13127 *   R1 HAS UNUSED WIDTH OF WINDOW
13128 *   STACK HAS POINT NUMBER IN FP NOTATION
13129 *   (INTERPOLATED)
13130 *   AN ERROR IS SET IF NO CROSSOVER IS FOUND
13131 *   (STACK WILL CONTAIN FP ZERO IF NO CROSSPOINT IS
13132 *   FOUND OR AN ERROR CONDITION EXISTS WHEN CALLED)
13133 *
13134 *   REGISTER USAGE ---
13135 *   R3 HIT ELEMENT NUMBER (AFTER CROSSPOINT)
13136 *   R4 LAST ELEMENT BEFORE CROSSOVER
13137 *   R5 NEGATIVE FLAG
13138 *   R6 FWD/REVS FLAG
13139 *   R7 LAST POINT VALUE
    
```

```

13141 5A12 DBA0 CNTCRJS WORD WPLVL3 09961
13142 6A14 6A16 WORD *+2 09962
13143 6A16 0728 DA1C SETJ CROSFLG 09963
13144 5A1A 1008 JMP STRTCROS 09964
13145 6A1C DBA0 FWDCRJS WORD WPLVL3 09965
13146 6A1E 6A20 WORD *+2 09966
13147 6A20 04C6 CLR R6 R6=0 FOR FWD CHECK 09967
13148 6A22 04E0 DA1C CLR CROSFLG 09968
13149 6A26 1005 JMP STRTCROS 09969
13150 6A28 DBA0 BAKCRJS WORD WPLVL3 09970
13151 6A2A 6A2C WORD *+2 09971
13152 6A2C C18E MOV R14,R6 R6<>0 FOR RVS CHECK 09972
13153 6A2E 04E0 DA1C CLR CROSFLG 09973
13154 6A32 6A32 STRTCROS EQU * 09974
13155 6A32 C168 D94A MOV FATAL,R5 09975
13156 6A36 1101 JLT *+4 09976
13157 6A38 102D JMP NOCRJS 09977
13158 6A3A C26D 0012 MOV 18(R13),SOFT GET STACK POINTER 09978
13159 6A3E C81D MOV *R13,R0 GET CALLERS R0,R1,R2 09979
13160 6A40 C06D 0002 MOV 2(R13),R1 09980
13161 6A44 C0AD 0004 MOV 4(R13),R2 09981
13162 6A48 04C5 CLR R5 09982
13163 6A4A C1F0 MOV *R0+,R7 09983
13164 6A4C C2E0 DA3E MOV CROSFLAG,R11 HAVE WE ASKED TO START AT CURRENT POINT? 09984
13165 6A50 150C JGT GETFRST YES 09985
13166 6A52 C2E0 DA1C MOV CROSFLG,R11 09986
13167 6A56 1609 JNE GETFRST 09987
13168 6A58 C186 MOV R6,R6 09988
13169 6A5A 1302 JEQ *+6 09989
13170 6A5C 0640 DECT R0 ADD LAST POINT IF BACK CROSS 09990
    
```

13171	6A5E	0640		DECT R0		09991
13172	6A60	C210		MOV *R0,R0	GET SECOND POINT	09992
13173	6A62	A1C8		A R8,R7	AVERAGE FIRST TWO POINTS	09993
13174	6A64	0817		SRA R7,1	DIVIDE SUM BY TWO TO GET AVERAGE	09994
13175	6A66	1701		JNC *+4	ROUND UP IF NECESSARY	09995
13176	6A68	0587		INC R7		09996
13177			6A6A	GETFRST EQU *		09997
13178	6A6A	C010		MOV *R13,R0	GET FIRST POINT AGAIN	09998
13179	6A6C	8087		STRCHK C R7,R2	STARTING HIGH OR LOW?	09999
13180	6A6E	1358		JEQ PLATEAU		10000
13181	6A70	1501		JGT *+4		10001
13182	6A72	0705		SET0 R5	STARTING LOW	10002
13183			6A74	CROSL0P EQU *		10003
13184	6A74	0601		DEC R1		10004
13185	6A76	130E		JEQ NOCROS	ERROR IF NO CROSSING	10005
13186	6A78	1100		JLT NOCROS	ERROR IF ENTIRE AREA WAS PLATEAU	10006
13187	6A7A	C100		MOV R0,R4		10007
13188	6A7C	C1F0		MOV *R0+,R7		10008
13189	6A7E	C186		MOV R6,R6	FWD/RVS?	10009
13190	6A80	1302		JEQ *+6		10010
13191	6A82	0640		DECT R0	BACK UP INSTEAD	10011
13192	6A84	0640		DECT R0		10012
13193	6A86	C100		MOV *R0,R7		10013
13194	6A88	C145		MOV R5,R5	LOOKING FOR TRANSITION TO HIGH OR LOW?	10014
13195	6A8A	1300		JEQ LOOK4LOW		10015
13196	6A8C	06A0	6B54	BL FUZZCHK		10016
13197	6A90	11F1		JLT CROSL0P		10017
13198	6A92	100C		JMP HITCROS		10018
13199	6A94	04E0	094A	NOCROS CLR FATAL		10019
13200	6A98	0649		DECT SOFT		10020
13201	6A9A	0719		SET0 *SOFT		10021
13202	6A9C	0649		DECT SOFT		10022
13203	6A9E	0409		CLR *SOFT		10023
13204	6AA0	0B49	0012	MOV SOFT,18(R13)		10024
13205	6AA4	0380		RTWP		10025
13206	6AA6	06A0	6B54	LOOK4LOW BL FUZZCHK		10026
13207	6AAA	15E4		JGT CROSL0P		10027
13208			6AAC	HITCROS EQU *		10028
13209	6AAC	0620	DAC0	DEC CROSNUM	HOW MANY CROSSES WERE WANTED?	10029
13210	6AB0	1104		JLT *+10		10030
13211	6AB2	1303		JEQ *+8		10031
13212	6AB4	0545		INV R5	LOOKING FOR CROSS IN OTHER DIRECTION	10032
13213	6AB6	C100		MOV *R0,R7		10033
13214	6AB8	10D9		JMP STRCHK		10034
13215	6ABA	C104		MOV *R4,R7		10035
13216	6ABC	61C2		S R2,R7	R7 NOW HAS LAST POINT VALUE MINUS DESIRED VALUE	10036
13217	6ABE	C210		MOV *R0,R0		10037
13218	6AC0	5214		S *R4,R0	R0 NOW HAS NEXT POINT VALUE MINUS LAST VALUE	10038
13219	6AC2	0747		ABS R7		10039
13220	6AC4	0748		ABS R0		10040
13221	6AC6	C748		MOV R0,*R13	RETURN POINTER TO ELEMENT AFTER CROSSOVER...	10041
13222	6AC8	C2C8		MOV R0,R11		10042
13223	6ACA	081B		SRA R11,1		10043
13224	6ACC	82C7		C R7,R11	RETURN SMALLEST POINT NUMBER IF 2 POINTS EQUAL	10044
13225	6ACE	1501		JGT *+4		10045
13226	6AD0	C744		MOV R4,*R13	... UNLESS CLOSER TO PREVIOUS ELEMENT VALUE	10046
13227	6AD2	0B41	0002	MOV R1,2(R13)	RETURN REMAINING WINDOW WIDTH	10047
13228	6AD6	C047		MOV R7,R1		10048
13229	6AD8	0420	6F0C	BLWP I2FP	FLOAT AND PUSH VALUE MISSED BY	10049

13230	6ADC	0649		DECT SOFT		10050
13231	6ADE	C642		MOV R2,*SOFT		10051
13232	6AE0	0649		DECT SOFT		10052
13233	6AE2	C641		MOV R1,*SOFT		10053
13234	6AE4	C048		MOV R8,R1		10054
13235	6AE6	0420	6F0C	BLWP I2FP	FLOAT TOTAL INTERPOINT VALUE	10055
13236	6AEA	0420	7050	BLWP FPDIV	GET RATIO OF INTERPOINT TIME TO FULL POINT TIME	10056
13237	6AEE	0720	D94C	SETO WARNING	IGNORE WARNINGS	10057
13238	6AF2	C2C4		MOV R4,R11		10058
13239	6AF4	62E0	D95C	S OPWFMD,R11		10059
13240	6AF8	0818		SRA R11,1	LAST ELEMENT #	10060
13241	6AFA	C048		MOV R11,R1		10061
13242	6AFC	0420	6F0C	BLWP I2FP		10062
13243	6B00	C186		MOV R6,R6		10063
13244	6B02	1301		JEQ **4		10064
13245	6B04	0519		NEG *SOFT	IF BACKWARD SEARCH, NEED POINT - PARTIAL	10065
13246	6B06	0420	6F84	BLWP FPADD	POINT + PARTIAL	10066
13247	6B0A	C849	0012	MOV SOFT,18(R13)	HORIZ. MANTISSA NOW ON STACK	10067
13248	6B0E	0380		RTWP		10068
13249			6B10	P_PLATEAU EQU *		10069
13250	6B10	C2E0	DA1C	MOV CROSFLG,R11	IF CNTGROS, RETURN 1ST POINT	10070
13251	6B14	161C		JNE FSTPT		10071
13252	6B16	04CC		CLR R12		10072
13253	6B18	C186		STILPLA MOV R6,R5	SKIP OVER THE PLATEAU,R6	10073
13254	6B1A	1602		JNE **6		10074
13255	6B1C	05C0		INCT R0		10075
13256	6B1E	1001		JMP **4		10076
13257	6B20	0640		DECT R0		10077
13258	6B22	0601		DEC R1		10078
13259	6B24	1387		JEQ NOCRJS		10079
13260	6B26	1186		JLT NOCRJS		10080
13261	6B28	C100		MOV *R0,R7		10081
13262	6B2A	06A8	6B54	BL FUZZCHK		10082
13263	6B2E	13F4		JEQ STILPLA		10083
13264	6B30	1101		JLT **4		10084
13265	6B32	070C		SETO R12		10085
13266	6B34	8820	DABE 333A	C CROSFLAG,C1	DOES FIRST POINT COUNT?	10086
13267	6B3A	1698		JNE STRTCHK		10087
13268	6B3C	C2E0	DAC0	MOV CROSNUM,R11	DIRECTION OF SEARCH IMPORTANT	10088
13269	6B40	1305		JEQ ONESRCH	WHEN A TOUCHING DETECTED	10089
13270	6B42	0648		DECT R11		10090
13271	6B44	1502		JGT **6		10091
13272	6B46	8305		C R5,R12		10092
13273	6B48	1695		JNE CROSLOP		10093
13274	6B4A	C14C		MOV R12,R5	SET SEARCH DIRECTION FLAG	10094
13275	6B4C	0640		ONESRCH DECT R0	BACK UP TO PLATEAU POINT	10095
13276	6B4E	C100		FSTPT MOV R0,R4		10096
13277	6B50	05C0		INCT R0		10097
13278	6B52	10AC		JMP HITCROS		10098

FUZZCHK --- CHECKS TO SEE IF TWO VALUES ARE EQUAL TO WITHIN 1 10099

```

13280 *
13281 * (THIS WAS NECESSARY FOR >VCRD, PER AND FREQ DUE TO
13282 * ERRORS THAT CAN BE INTRODUCED DURING COMPUTATIONS)
13283 *
13284 * LEVEL 5 ROUTINE
13285 *
13286 * INPUT - R2,R7 WAVEFORM MANTISSA VALUES TO COMPARE
13287 *
13288 * REGISTER USAGE
13289 * R8 - ZERO IF DIFFERENCE WITHIN + OR - ONE BIT
13290 * NEGATIVE IF R7 > R2
13291 * POSITIVE IF R2 > R7
13292 *
13293 *
    
```

```

13294 6854 C207 FJZZCHK MOV R7,R8 10100
13295 6856 8088 C R8,R2 10101
13296 6858 1306 JEQ DONCHK IF EQUAL JUST RETURN 10102
13297 685A C145 MOV R5,R5 10103
13298 685C 1602 JNE RESPOS 10104
13299 685E 0588 INC R8 SET EQUAL FLAG IF DIFFERENCE <= 1 BIT 10105
13300 6860 1001 JMP **4 10106
13301 6862 0608 RESPOS DEC R8 SET EQUAL FLAG IF DIFFERENCE <= 1 BIT 10107
13302 6864 8088 C R8,R2 10108
13303 6866 045B DONCHK B *R11 10109
    
```

13305				*				
13306				*	CONVERT FROM FLOATING POINT TO WFM MANTISSA			
13307				*				
13308				*	INPUT --- FP NUMBER IN R1,R2			
13309				*	OUTPUT -- WFM MANTISSA IN R1 CORRESPONDING TO # FOR OPWFM			
13310				*				
13311				*	MANT = (FP / VEXP) - OFFSET			
13312				*				
13313				*	LEVEL 3 ROUTINE			
13314				*				
13315	6B68		0B40	FP2WFM	WORD WPLVL3			10111
13316	6B6A		6B6C		WORD *+2			10112
13317	6B6C	C26D	0012		MOV 18(R13),SOFT GET SOFT STACK			10113
13318	6B70	0649			DECT SOFT			10114
13319	6B72	366D	0004		MOV 4(R13),*SOFT PUSH FP NUMBER			10115
13320	6B76	0649			DECT SOFT			10116
13321	6B78	C66D	0002		MOV 2(R13),*SOFT			10117
13322	6B7C	C02D	D95E		MOV OPWFMH,R0 OPWFM HEADER			10118
13323	6B80	C08D			MOV R0,R2			10119
13324	6B82	A0A0	3338		A VEXP,R2			10120
13325	6B86	C072			MOV *R2*,R1 GET VERT EXP.			10121
13326	6B88	C092			MOV *R2,R2			10122
13327	6B8A	042D	7046		BLWP FPDIVZ FP / VEXP WITH ZERO EXPONENT ON STACK			10123
13328	6B8E	1902			JND *+6			10124
13329	6B90	04E0	D94C		CLR WARNING			10125
13330	6B94	A02D	3348		A VOFFAB,R0 GET OFFSET			10126
13331	6B98	665D			S *R0,*SOFT FP - OFFSET ON STACK			10127
13332	6B9A	1902			JND *+6			10128
13333	6B9C	04E0	D94C		CLR WARNING			10129
13334	6BA0	C859	0002		MOV *SOFT,2(R13) RETURN MANTISSA			10130
13335	6BA4	038D			RTWP			10131


```

13337 *****
13338 **
13339 ** CONVERT WAVEFORM POINT TO FLOATING POINT VALUE **
13340 **
13341 ** LEVEL 3 ROUTINE **
13342 **
13343 ** INPUT: PNT2FP - POINT NUMBER IN R1 **
13344 ** WFM2FP - POINT VALUE IN R1 **
13345 ** OUTPUT: FLOATING POINT VALUE IN R1,R2 **
13346 **
13347 ** STACK OPERATIONS: **
13348 ** SOFTSTACK - USED BUT NO AFFECT TO CALLER **
13349 **
13350 *****
    
```

13351	6BA6		D8A0	PNT2FP	WORD	HPLVL3	LEVEL 3 ROUTINE	10133
13352	6BA8		6BAA		WORD	*+2		10134
13353	6BAA	C06D	0002		MOV	2(R13),R1	GET POINT NUMBER FROM CALLING ROUTINE	10135
13354	6BAE	0A11			SLA	R1,1		10136
13355	6BB0	A060	D95C		A	OPWFM,R1	ADDRESS OF POINT IN OPWFM	10137
13356	6BB4	C051			MOV	*R1,R1	GET POINT VALUE	10138
13357	6BB6	1004			JMP	PNTWFM		10139
13358								
13359	6BB8		D8A0	WFM2FP	WORD	HPLVL3	LEVEL 3 ROUTINE	10140
13360	6BBA		6B8C		WORD	*+2		10141
13361	6BBC	C06D	0002		MOV	2(R13),R1	GET POINT VALUE FROM CALLING ROUTINE	10142
13362	6BC0	C26D	0012	PNTWFM	MOV	18(R13),SOFT	GET CURRENT SOFTSTACK POINTER	10143
13363	6BC4	0649			DECT	SOFT		10144
13364	6BC6	04D9			CLR	*SOFT	PUSH EXPONENT OF 0 ONTO SOFTSTACK	10145
13365	6BC8	0649			DECT	SOFT		10146
13366	6BCA	C641			MOV	R1,*SOFT	PUSH POINT VALUE ONTO SOFTSTACK	10147
13367	6BCC	C060	D95E		MOV	OPWFM,R1	ADDRESS OF OPWFM'S DATA	10148
13368	6BD0	C081			MOV	R1,R2		10149
13369	6BD2	A060	3348		A	VOFFAB,R1	ADDRESS OF VERTICAL OFFSET	10150
13370	6BD6	A060	3338		A	VEXP,R1	ADDRESS OF VERTICAL EXPANDER	10151
13371	6BDA	A651			A	*R1,*SOFT	ADD VERTICAL OFFSET TO POINT VALUE	10152
13372	6BDC	C072			MOV	*R2+,R1	LOAD VERTICAL EXPANDER	10153
13373	6BDE	C092			MOV	*R2,R2		10154
13374	6BE0	0420	70AA		BLW	FPMPY	MULTIPLY POINT VALUE BY VEXP TO GET ACTUAL VALUE	10155
13375	6BE4	C879	0002		MOV	*SOFT+,2(R13)	RETURN RESULT TO CALLER	10156
13376	6BE8	C879	0004		MOV	*SOFT+,4(R13)		10157
13377	6BEC	0380			RTWP			10158

FILLPNT --- FILL UNDEFINED POINTS

10159

```

13379 *****
13380 **
13381 ** INTERPOLATE UNDEFINED POINTS IN WAVEFORM **
13382 ** **
13383 ** LEVEL 4 ROUTINE **
13384 ** **
13385 ** INPUT: WAVEFORM # OR ADDRESS ON SOFTSTACK **
13386 ** OUTPUT: UNDEFINED POINTS IN WAVEFORM INTERPOLATED **
13387 ** **
13388 ** NOTE --- **
13389 ** 'PNTFILL' - HAS DATA ADDRESS ON SOFTSTACK **
13390 ** 'FILLPNT' - HAS WAVEFORM # ON SOFTSTACK **
13391 ** **
13392 *****
13393 68EE DBC0 PNTFILL WORD WPLVL4 LEVEL 4 WORKSPACE 10160
13394 68F0 6BF2 WORD *+2 10161
13395 69F2 C26D 0012 MOV 18(R13),SOFT GET CURRENT SOFTSTACK POINTER 10162
13396 68F6 1008 JMP DATAORS 'PNTFILL' HAS DATA ADDRESS ON STACK 10163
13397 68F8 DBC0 FILLPNT WORD WPLVL4 10164
13398 68FA 6BFC WORD *+2 10165
13399 68FC C26D 0012 MOV 18(R13),SOFT GET CURRENT SOFTSTACK POINTER 10166
13400 6C00 C079 MOV *SOFT+,R1 POP WAVEFORM # 10167
13401 6C02 06A0 6984 BL ADRWFH GET HEADER AND DATA ADDRESS 10168
13402 6C06 05C9 INCT SOFT POP HEADER ADDRESS (NOT USED) 10169
13403 6C08 C0F9 DATADRS MOV *SOFT+,R3 POP DATA ADDRESS OFF SOFTSTACK 10170
13404 6C0A CB49 0012 MOV SOFT,18(R13) RETJRN NEW SOFTSTACK POINTER 10171
13405 6C0E C103 MOV R3,R4 10172
13406 6C10 C150 D970 MOV RESOLV,R5 GET CURRENT WAVEFORM RESOLUTION 10173
13407 6C14 C1A0 3390 MOV BADPNT,R6 GET VALUE OF UNDEFINED POINTS 10174
13408
13409 ***** FIND FIRST GOOD POINT *****
13410
13411 6C18 8184 C *R4+,R5 COMPARE NEXT POINT TO UNDEFINED VALUE 10175
13412 6C1A 1608 JNE FNDFRST IF NOT EQUAL, THIS IS GOOD POINT 10176
13413 6C1C 0605 DEC R5 IF EQUAL, DECREMENT POINT COUNT 10177
13414 6C1E 15FC JGT *-6 IF MORE POINTS, CONTINUE 10178
13415
13416 ***** ALL POINTS ARE UNDEFINED SO WAVEFORM IS ZEROED *****
13417
13418 6C20 C160 D970 MOV RESOLV,R5 GET POINTS/WAVEFORM 10179
13419 6C24 04F3 CLR *R3+ SET ALL POINTS TO ZERO 10180
13420 6C26 0605 DEC R5 10181
13421 6C28 15FD JGT *-4 10182
13422 6C2A 0380 RTWP RETURN TO CALLER 10183
13423
13424 ***** FIRST GOOD POINT IS EXTENDED TO LEFT EDGE OF WAVEFORM *****
13425
13426 6C2C 0644 FNDFRST DECT R4 REPOINT TO FIRST GOOD POINT 10184
13427 6C2E CCD4 MOV *R4,*R3+ EXTEND GOOD POINT TO LEFT 10185
13428 6C30 8103 C R3,R4 10186
13429 6C32 12FD JLE *-4 10187
13430 6C34 0605 DEC R5 DECREMENT POINT COUNT 10188
13431 6C36 1501 JGT FIND3A) 10189
13432 6C38 0380 RTWP IF LAST POINT RETURN TO CALLER 10190
13433
13434 ***** FIND NEXT UNDEFINED POINT *****
13435
13436 6C3A 8183 FINDBAD C *R3+,R6 FIND NEXT UNDEFINED POINT 10191
13437 6C3C 1303 JEQ FNDFRST WHEN FOUND FIND NEXT GOOD POINT 10192

```

FILLPNT --- FILL UNDEFINED POINTS

10159

13438	6C3E	0605	DEC	R5	DECREMENT POINT COUNT	10193	
13439	6C40	15FC	JGT	FIND3A)		10194	
13440	6C42	0380	RTWP		RETURN TO CALLER IF REST GOOD	10195	
13441							
13442			***** FIND NEXT GOOD POINT *****				
13443							
13444	6C44	0643	FNDNEXT	DECT R3	POINT BACK TO UNDEFINED POINT	10196	
13445	6C46	C103	MOV	R3,R4		10197	
13446	6C48	0643	DECT	R3	POINT BACK TO LAST GOOD POINT	10198	
13447	6C4A	8134	C	*R4+,R5	FIND NEXT GOOD POINT	10199	
13448	6C4C	1607	JNE	FNDLAST	WHEN FOUND INTERPOLATE POINTS BETWEEN	10200	
13449	6C4E	0605	DEC	R5	DECREMENT POINT COUNT	10201	
13450	6C50	15FC	JGT	*-6		10202	
13451							
13452			***** LAST GOOD POINT IS EXTENDED TO RIGHT EDGE OF WAVEFORM *****				
13453							
13454	6C52	0644	DECT	R4	POINT BACK TO LAST POINT IN WAVEFORM	10203	
13455	6C54	C513	MOV	*R3,*R4	EXTEND LAST GOOD POINT TO RIGHT	10204	
13456	6C56	8103	C	R3,R4		10205	
13457	6C58	1AFC	JL	*-6		10206	
13458	6C5A	0380	RTWP		RETURN TWO CALLER	10207	
13459							
13460			***** INTERPOLATE POINTS BETWEEN GOOD POINTS *****				
13461							
13462	6C5C	0644	FNDLAST	DECT R4	POINT BACK TO GOOD POINT	10208	
13463	6C5E	C004	MOV	R4,R0		10209	
13464	6C60	6003	S	R3,R0	DELTA HORIZONTAL BETWEEN GOOD POINTS	10210	
13465	6C62	0910	SRL	R0,1		10211	
13466	6C64	0708	SETO	R11	SET FLAG INDICATING NEGATIVE SLOPE	10212	
13467	6C66	C104	MOV	*R4,R7	VALUE OF RIGHT GOOD POINT	10213	
13468	6C68	6103	S	*R3,R7	DELTA VERTICAL BETWEEN GOOD POINTS	10214	
13469	6C6A	0747	ABS	R7	GET ABSOLUTE VALUE FOR TMS9900 'MPY' & 'DIV'	10215	
13470	6C6C	1101	JLT	*+4		10216	
13471	6C6E	04CB	CLR	R11	SET FLAG INDICATING POSITIVE SLOPE	10217	
13472	6C70	C333	MOV	*R3+,R12	SAVE VALUE OF LEFT GOOD POINT	10218	
13473	6C72	04CB	CLR	R8		10219	
13474	6C74	0588	FILL	INC R8	NEXT POINT NUMBER TO FILL	10220	
13475	6C76	C047	MOV	R7,R1	DELTA VERTICAL	10221	
13476	6C78	1309	JEQ	DELTAVD	IF DELTA VERTICAL = 0, SKIP CALCULATIONS	10222	
13477	6C7A	3848	MPY	R8,R1	POINT # * DELTA VERTICAL	10223	
13478	6C7C	3C40	DIV	R0,R1	(POINT # * DELTA VERTICAL) / DELTA HORIZONTAL	10224	
13479	6C7E	0A12	SLA	R2,1		10225	
13480	6C80	8002	C	R2,R0	ROUND RESULT	10226	
13481	6C82	1101	JLT	*+4		10227	
13482	6C84	0581	INC	R1		10228	
13483	6C86	C2CB	MOV	R11,R11	CHECK SLOPE OF INTERPOLATED DATA	10229	
13484	6C88	1301	JEQ	*+4		10230	
13485	6C8A	0501	NEG	R1	IF NEGATIVE, NEGATE ADJUSTMENT VALUE	10231	
13486	6C8C	C4CC	DELTAVD	MOV R12,*R3	MOVE VALUE OF LEFT POINT TO THIS POINT	10232	
13487	6C8E	ACC1	A	R1,*R3+	ADD ADJUSTMENT VALUE	10233	
13488	6C90	8103	C	R3,R4	CHECK IF ALL POINTS FILLED	10234	
13489	6C92	1AF0	JL	FILL		10235	
13490	6C94	10D2	JMP	FINDBAD	WHEN FILLED, FIND NEXT UNDEFINED POINT	10236	
13491	6C96	C2CB	MOV	R11,R11	CHECK VALUE OF SLOEPPE	10237	

13493				*				
13494				*	CHECK IF STACK(X) IS A WFM.			
13495				*	IF SO, IT IS THE NEW OPERATIONAL WFM.			
13496				*	IN THAT CASE, SET UP THE UPDATED OPWFM, OPWFMD, AND OPWFMH			
13497				*	AS WELL AS TURNING OFF THE DISPLAY FOR THE OLD OPERATIONAL			
13498				*	WFM AND TURNING ON THE DISPLAY FOR THE NEW ONE.			
13499				*				
13500				*	LEVEL 4 ROUTINE			
13501				*				
13502	6C98		DBC0	OPWCHG	WORD WPLVL4	LEVEL 4 ROUTINE		10239
13503	6C9A		6C9C		WORD *+2			10240
13504	6C9C	C26D	0012		MOV 18(R13),SOFT			10241
13505	6CA0	C060	D93A		MOV PROGRS,R1	NO CHANGE DURING NUMERIC ENTRY		10242
13506	6CA4	1520			JGT NOCHG			10243
13507	6CA6	C0E0	33D6		MOV XSTK,R3	IS X A WFM?		10244
13508	6CAA	C073			MOV *R3+,R1			10245
13509	6CAC	131C			JEQ NOCHG	NO, THEN NO NEW OPWFM		10246
13510	6CAE	84E0	D95A		C OPWFM,*R3	IS THIS A NEW OPWFM?		10247
13511	6CB2	1319			JEQ NOCHG	IF NOT, DON'T BOTHER WITH CALC OR DISPLAY		10248
13512	6CB4	0720	D972		SETJ RDTFLAG			10249
13513	6CB8	C060	D95A		MOV OPWFM,R1	OLD OPWFM #		10250
13514	6CBC	06A0	6984		BL ADRWFM			10251
13515	6CC0	3039			MOV *SOFT+,R0			10252
13516	6CC2	A020	3354		A DISPLA,R0			10253
13517	6CC6	4420	333C		SZC C2,*R0	TURN OFF OLD OPWFM		10254
13518	6CCA	C053			MOV *R3,R1	NEW OPWFM #		10255
13519	6CCC	3801	D95A		MOV R1,OPWFM			10256
13520	6CD0	06A0	6984		BL ADRWFM			10257
13521	6CD4	C039			MOV *SOFT+,R3	HEADER ADDRESS		10258
13522	6CD6	C800	D95E		MOV R0,OPWFMH			10259
13523	6CDA	A020	3354		A DISPLA,R0			10260
13524	6CDE	E420	333C		SOC C2,*R0	TURN ON NEW OPWFM		10261
13525	6CE2	C839	D95C		MOV *SOFT+,OPWFMD	DATA ADDRESS		10262
13526	6CE6	0380		NOCHG	RTWP			10263

BUZZIT

10264

13528				*				
13529				*	BUZZIT --- SOUND BUZZER FOR FIXED NOTE AND LENGTH AS WARNING TO USE			
13530				*	ALSO TURN ON ERROR LED FOR LITERR			
13531				*				
13532				*	LEVEL 6			
13533				*				
13534			6CE8	LITERR	EQU *			10265
13535	6CE8		DBE0		WORD WPLVL6			10266
13536	6CEA		6CEC		WORD *+2			10267
13537	6CEC	C320	33C2		MOV ERRLED,R12	LIGHT ERROR LED		10268
13538	6CF0	1000			SBJ 0			10269
13539	6CF2	1002			JMP ANDBUZZ			10270
13540			6CF4	BUZZIT	EQU *			10271
13541	6CF4		DBE0		WORD WPLVL5			10272
13542	6CF6		6CF8		WORD *+2			10273
13543	6CF8	0300	0007	ANDBUZZ	LIMI 7	DISABLE KEYBOARD INTERRUPTS WHILE BUZZER ON		10274
13544	6CFC	C320	333E		MOV SPEAKER,R12			10275
13545	6D00	0208	0096		LI R8,150	LOAD LENGTH OF NOTE		10276
13546	6D04	1E00		NOTE	SBJ 0	TURN ON SPEAKER		10277
13547	6D06	0207	0050		LI R7,80	LOAD IN NOTE FREQUENCY		10278
13548	6D0A	0607			DEC R7			10279
13549	6D0C	18FE			JH *-2			10280
13550	6D0E	1000			SBJ 0	TURN OFF SPEAKER		10281
13551	6D10	0207	0050		LI R7,80	SPEED AT WHICH SPEAKER TURNED ON		10282
13552	6D14	0607			DEC R7	AND OFF DETERMINES NOTE FREQUENCY		10283
13553	6D16	18FE			JH *-2			10284
13554	6D18	0608			DEC R8			10285
13555	6D1A	18F4			JH NOTE			10286
13556	6D1C	0380			RTWP			10287

FP2ELE --- CONVERT FROM FP HORIZONTAL VALUE TO ELEMENT NUMBER 10288

13558				*					
13559				*					
13560				*					
13561				*					
13562				*					
13563				*					
13564				*					
13565				*					
13566				*					
13567				*					
13568			6D1E		FP2ELE	EQU *			10289
13569	6D1E	0649				DECT SOFT			10290
13570	6D20	C64B				MOV R11,*SOFT	STORE RETURN ADDRESS		10291
13571	6D22	C1E0	D95E			MOV OPWFHM,R7	STORE RETURN ADDRESS		10292
13572	6D26	A1E0	3340			A HEXP,R7	GET HORIZONTAL SCALE		10293
13573	6D2A	0649				DECT SOFT			10294
13574	6D2C	C642				MOV R2,*SOFT	PUSH HORIZONTAL TIME ONTO STACK		10295
13575	6D2E	0649				DECT SOFT			10296
13576	6D30	C641				MOV R1,*SOFT			10297
13577	6D32	3077				MOV *R7+,R1			10298
13578	6D34	C097				MOV *R7,R2			10299
13579	6D36	0420	7050			BLWP FPDIV	HORIZ. TIME / HORIZ. SCALE		10300
13580	6D3A	C079				MOV *SOFT+,R1			10301
13581	6D3C	C0B9				MOV *SOFT+,R2			10302
13582	6D3E	06A0	75A0			BL FP2INT	ELEMENT #		10303
13583	6D42	C041				MOV R1,R1	IF R1 IS NEGATIVE, VALUE IS NOT		10304
13584	6D44	1103				JLT *+8	ON THE WAVEFORM -- ERRJR		10305
13585	6D46	8060	D970			C RESOLV,R1	IF R1 > # POINTS IN WAVEFORM, VALUE IS		10306
13586	6D4A	1502				JGT *+6	NOT ON WAVEFORM -- ERROR		10307
13587	6D4C	04E0	D94A			CLR FATAL			10308
13588	6D50	C2F9				MOV *SOFT+,R11			10309
13589	6D52	045B				B *R11			10310

13591	*****							**
13592	**							**
13593	** CALCULATE SUM OF POINTS IN OPWFM							**
13594	**							**
13595	** LEVEL 3 ROUTINE							**
13596	**							**
13597	** OUTPUT: RB - WINDOW WIDTH							**
13598	**							**
13599	** STACK OPERATIONS:							**
13600	** SOFTSTACK - PUSHES 2 (SUM)							**
13601	**							**
13602	** NOTE ---							**
13603	** THE FOLLOWING EQUATION IS USED TO CALCULATE THE SUM							**
13604	** OVER THE INTERVAL DEFINED BY THE CURSORS:							**
13605	**							**
13606	** $S = \text{SUM} [(Y(J+1) + Y(J)) / 2] : J = N \text{ TO } M - 1$							**
13607	**							**
13608	** N & M ARE DETERMINED BY THE NUMBER OF CURSORS BEING							**
13609	** DISPLAYED:							**
13610	**							**
13611	** # OF CURSORS N M							**
13612	** 0 LEFT-SIDE RIGHT-SIDE							**
13613	** 1 CURSOR #1 RIGHT-SIDE							**
13614	** 2 CURSOR #1 CURSOR #2							**
13615	**							**
13616	*****							**
13617	6D54	DBA0	SJM	WORD	WPLV.3	LEVEL 3 WORKSPACE	10312	
13618	6D56	6D58		WORD	*+2		10313	
13619	6D58	C25D 0012		MOV	18(R13),SOFT	GET CURRENT SOFTSTACK POINTER	10314	
13620	6D5C	06A0 5A62		BL	GETENDS	GET END-POINTS AND INTERVAL SIZE	10315	
13621	6D60	C740		MOV	R0,*R13	RETURN WINDOW WIDTH TO CALLING ROUTINE	10316	
13622	6D62	A1A8 095C		A	OPWFM0,R6	STARTING ADDRESS FOR SUM	10317	
13623	6D66	C1E0 095E		MOV	OPWFMH,R7	HEADER ADDRESS FOR OPWFM	10318	
13624	6D6A	A1E0 3348		A	V0FFAB,R7	ADDRESS OF VERTICAL OFFSET OF OPWFM	10319	
13625	6D6E	C1D7		MOV	*R7,R7	VERTICAL OFFSET	10320	
13626	6D70	C076		MOV	*R6+,R1	GET FIRST POINT	10321	
13627	6D72	A047		A	R7,R1	ADD VERTICAL OFFSET	10322	
13628	6D74	C081		MOV	R1,R2		10323	
13629	6D76	08F1		SRA	R1,15	CREATE SIGN-EXTENDED DOUBLE WORD INTEGER	10324	
13630	6D78	0600		DEC	R0	CHECK IF ONLY ONE POINT	10325	
13631	6D7A	131E		JEQ	ONEPNT		10326	
13632	6D7C	0812		SRA	R2,1	DIVIDE FIRST POINT BY 2	10327	
13633	6D7E	1703		JNC	*+8	ROUND DIVISION	10328	
13634	6D80	0582		INC	R2		10329	
13635	6D82	1701		JNC	*+4		10330	
13636	6D84	0581		INC	R1		10331	
13637	6D86	0600		DEC	R0	CHECK IF ONLY TWO POINTS	10332	
13638	6D88	130A		JEQ	TWOPNTS		10333	
13639	6D8A	C0F6	SJMREST	MOV	*R6+,R3	GET NEXT POINT TO SUM	10334	
13640	6D8C	A0C7		A	R7,R3	ADD VERTICAL OFFSET	10335	
13641	6D8E	C103		MOV	R3,R4		10336	
13642	6D90	08F3		SRA	R3,15	CREATE SIGN-EXTENDED DOUBLE WORD INTEGER	10337	
13643	6D92	A043		A	R3,R1	DO DOUBLE WORD SUM	10338	
13644	6D94	A084		A	R4,R2		10339	
13645	6D96	1701		JNC	*+4		10340	
13646	6D98	0581		INC	R1	ADD CARRY FROM LEAST SIGNIFICANT WORD	10341	
13647	6D9A	0600		DEC	R0	DECREMENT REMAINING POINT COUNT	10342	
13648	6D9C	15F6		JGT	SUMREST	IF MORE, CONTINUE	10343	
13649	6D9E	C0D6	TWOPNTS	MOV	*R6,R3	GET LAST POINT TO SUM	10344	

13650	6DA0	A0C7		A	R7,R3	ADD VERTICAL OFFSET	10345
13651	6DA2	C103		MOV	R3,R4		10346
13652	6DA4	08F3		SRA	R3,15	CREATE SIGN-EXTENDED DOUBLE WORD INTEGER	10347
13653	6DA6	0814		SRA	R4,1	DIVIDE LAST POINT BY 2	10348
13654	6DA8	1703		JNC	*+8	ROUND DIVISION	10349
13655	6DAA	0584		INC	R4		10350
13656	6DAC	1701		JNC	*+4		10351
13657	6DAE	0583		INC	R3		10352
13658	6DB0	A043		A	R3,R1	DO DOUBLE WORD SUM	10353
13659	6DB2	A084		A	R4,R2		10354
13660	6DB4	1701		JNC	*+4		10355
13661	6DB6	0581		INC	R1	ADD CARRY FROM LEAST SIGNIFICANT WORD	10356
13662	6DB8	0649		DECT	SOFT		10357
13663	6DBA	C650	3358	MOV	C16,*SOFT	PUSH EXPONENT OF 16 ONTO SOFTSTACK	10358
13664	6DBE	0649		DECT	SOFT		10359
13665	6DC0	C642		MOV	R2,*SOFT	PUSH DOUBLE WORD MANTISSA ONTO SOFTSTACK	10360
13666	6DC2	0649		DECT	SOFT		10361
13667	6DC4	C641		MOV	R1,*SOFT		10362
13668	6DC6	06A0	713C	BL	NRMLIZ	NORMALIZE THE SUM INTO SINGLE PRECISION	10363
13669	6DCA	C0A0	D95E	MOV	OPWFM4,R2	HEADER ADDRESS FOR OPWFM	10364
13670	6DCE	A0A0	3338	A	VEXP,R2	ADDRESS OF VERTICAL MULTIPLIER OF OPWFM	10365
13671	6DD2	C072		MOV	*R2+,R1	VERTICAL MULTIPLIER	10366
13672	6DD4	C092		MOV	*R2,R2		10367
13673	6DD6	0420	70AA	BLWP	FPMPY	ACTUAL VALUE OF SUM	10368
13674	6DDA	CB49	0012	MOV	SOFT,16(R13)	RETURN NEW SOFTSTACK POINTER	10369
13675	6DDE	0380		RTW ²			10370


```

13677 *****
13678 **
13679 ** CALCULATE SUM OF SQUARES OF POINTS IN OPWM
13680 **
13681 ** LEVEL 3 ROUTINE
13682 **
13683 ** OUTPUT: R0 - WINDOW WIDTH
13684 **
13685 ** STACK OPERATIONS:
13686 **     SOFTSTACK - PUSHES 2 (SUM OF SQUARES)
13687 **
13688 ** NOTE ---
13689 **     THE FOLLOWING EQUATION IS USED TO CALCULATE THE SUM
13690 **     OF SQUARES OVER THE INTERVAL DEFINED BY THE CURSORS:
13691 **
13692 **     S = SUM ( ( Y(J+1)**2 + Y(J)**2 ) / 2 ) ; J = N TO M - 1
13693 **
13694 **     N & M ARE DETERMINED BY THE NUMBER OF CURSORS BEING
13695 **     DISPLAYED:
13696 **
13697 **     # OF CURSORS      N      M
13698 **     0      LEFT-SIDE  RIGHT-SIDE
13699 **     1      CURSOR #1  RIGHT-SIDE
13700 **     2      CURSOR #1  CURSOR #2
13701 **
13702 *****
    
```

Address	Hex	Hex	Hex	OpCode	Operand	Description	Address
13703	6DE0	DBA0		SJMSQR	WORD WPLVL3	LEVEL 3 WORKSPACE	10372
13704	6DE2	6DE4			WORD **2		10373
13705	6DE4	C26D	0012	MOV	18(R13),SOFT	GET CURRENT SOFTSTACK POINTER	10374
13706	6DE8	06A0	5A62	BL	GETENDS	GET END-POINTS AND INTERVAL SIZE	10375
13707	6DEC	C740		MOV	R0,*R13	RETURN WINDOW WIDTH TO CALLING ROUTINE	10376
13708	6DEE	A1A0	D95C	A	OPWMH,R6	STARTING ADDRESS FOR SUM	10377
13709	6DF2	C1E0	D95E	MOV	OPWMH,R7	HEADER ADDRESS FOR OPWM	10378
13710	6DF6	A1E0	3348	A	VOFFAB,R7	ADDRESS OF VERTICAL OFFSET OF OPWM	10379
13711	6DFA	C1D7		MOV	*R7,R7	VERTICAL OFFSET	10380
13712	6DFC	04C1		CLR	R1	CREATE TRIPLE WORD INTEGER FOR SUM	10381
13713	6DFE	C036		MOV	*R6+,R2	GET FIRST POINT	10382
13714	6E00	A087		A	R7,R2	ADD VERTICAL OFFSET	10383
13715	6E02	0742		ABS	R2	ABSOLUTE VALUE OF R2 FOR 'MPY' INSTRUCTION	10384
13716	6E04	3892		MPY	R2,R2	SQUARE THE FIRST POINT	10385
13717	6E06	0600		DEC	R0	CHECK IF ONLY ONE POINT	10386
13718	6E08	132A		JEQ	PNTONE		10387
13719	6E0A	0913		SRL	R3,1	DIVIDE FIRST POINT BY 2	10388
13720	6E0C	1701		JNC	**4	ROUND DIVISION	10389
13721	6E0E	0583		INC	R3		10390
13722	6E10	0912		SRL	R2,1		10391
13723	6E12	1702		JNC	**6		10392
13724	6E14	0223	8000	AI	R3,\$8000	ADD CARRY	10393
13725	6E18	0600		DEC	R0	CHECK IF ONLY TWO POINTS	10394
13726	6E1A	130E		JEQ	PNTSTW0		10395
13727	6E1C	3136		SRRREST	MOV *R6+,R4	GET NEXT POINT TO SQUARE	10396
13728	6E1E	A107		A	R7,R4	ADD VERTICAL OFFSET	10397
13729	6E20	0744		ABS	R4	ABSOLUTE VALUE OF R4 FOR 'MPY' INSTRUCTION	10398
13730	6E22	3904		MPY	R4,R4	SQUARE THIS POINT	10399
13731	6E24	A0C5		A	R5,R3	DO TRIPLE PRECISION SUM	10400
13732	6E26	1703		JNC	**8		10401
13733	6E28	0582		INC	R2	ADD CARRY	10402
13734	6E2A	1701		JNC	**4		10403
13735	6E2C	0581		INC	R1		10404

13736	6E2E	A084		A	R4,R2		10405	
13737	6E30	1701		JNC	*+4		10406	
13738	6E32	0581		INC	R1	ADD CARRY	10407	
13739	6E34	0600		DEC	R0	DECREMENT REMAINING POINT COUNT	10408	
13740	6E36	15F2		JGT	SQRREST	IF MORE, CONTINUE	10409	
13741	6E38	C116		PNTSTWO	MOV	*R6,R4	GET LAST POINT	10410
13742	6E3A	A107		A	R7,R4	ADD VERTICAL OFFSET	10411	
13743	6E3C	0744		ABS	R4	ABSOLUTE VALUE OF R4 FOR 'MPY' INSTRUCTION	10412	
13744	6E3E	3904		MPY	R4,R4	SQUARE LAST POINT	10413	
13745	6E40	0915		SRL	R5,1	DIVIDE LAST POINT BY 2	10414	
13746	6E42	1701		JNC	*+4	ROUND DIVISION	10415	
13747	6E44	0585		INC	R5		10416	
13748	6E46	0914		SRL	R4,1		10417	
13749	6E48	1702		JNC	*+6		10418	
13750	6E4A	0225	8000	AI	R5,\$8000	ADD CARRY	10419	
13751	6E4E	A0C5		A	R5,R3	DO TRIPLE PRECISION SUM	10420	
13752	6E50	1703		JNC	*+8		10421	
13753	6E52	0582		INC	R2	ADD CARRY	10422	
13754	6E54	1701		JNC	*+4		10423	
13755	6E56	0581		INC	R1		10424	
13756	6E58	A084		A	R4,R2		10425	
13757	6E5A	1701		JNC	*+4		10426	
13758	6E5C	0581		INC	R1	ADD CARRY	10427	
13759	6E5E	0649		PNTONE	DECT	SOFT	10428	
13760	6E60	C660	335A	MOV	C17,*SOFT	PUSH EXPONENT OF 17 ONTO SOFTSTACK	10429	
13761	6E64	C041		MOV	R1,R1		10430	
13762	6E66	1604		JNE	SHIFT1		10431	
13763	6E68	C082		MOV	R2,R2	CHECK FOR ZERO	10432	
13764	6E6A	1602		JNE	SHIFT1		10433	
13765	6E6C	C0C3		MOV	R3,R3		10434	
13766	6E6E	1305		JEQ	SQRNRN		10435	
13767	6E70	0A11		SHIFT1	SLA	R1,1	PERFORM TRIPLE PRECISION NORMALIZE	10436
13768	6E72	1912		JNO	SHIFT2	SHIFT UNTIL OVERFLOW	10437	
13769	6E74	1701		JNC	*+4	ON OVERFLOW UNDO LAST SHIFT OF R1	10438	
13770	6E76	0581		INC	R1		10439	
13771	6E78	0B11		SRC	R1,1		10440	
13772	6E7A	0649		SQRNR1	DECT	SOFT	10441	
13773	6E7C	C641		MOV	R1,*SOFT	PUSH MANTISSA ONTO SOFTSTACK	10442	
13774	6E7E	C0A0	D95E	MOV	OPWFMH,R2	HEADER ADDRESS OF OPWFM	10443	
13775	6E82	A0A0	3338	A	VEXP,R2	ADDRESS OF VERTICAL MULTIPLIER OF OPWFM	10444	
13776	6E86	C072		MOV	*R2+,R1	VERTICAL MULTIPLIER	10445	
13777	6E88	C092		MOV	*R2,R2		10446	
13778	6E8A	0420	70AA	BLWP	FPMPY	MULTIPLY BY VEXP**2	10447	
13779	6E8E	0420	70AA	BLWP	FPMPY		10448	
13780	6E92	CB49	0012	MOV	SOFT,18(R13)	RETURN NEW SOFTSTACK POINTER	10449	
13781	6E96	0380		RTW			10450	
13782	6E98	0A12		SHIFT2	SLA	R2,1	SHIFT ALL THREE WORDS	10451
13783	6E9A	1701		JNC	*+4		10452	
13784	6E9C	0581		INC	R1	ADD CARRY	10453	
13785	6E9E	0A13		SLA	R3,1		10454	
13786	6EA0	1701		JNC	*+4		10455	
13787	6EA2	0582		INC	R2	ADD CARRY	10456	
13788	6EA4	0619		DEC	*SOFT	DECREMENT EXPONENT TO BALANCE SHIFT	10457	
13789	6EA6	10E4		JMP	SHIFT1	CONTINUE SHIFTING	10458	

NEWHSCL

10459

```

13791 *
13792 *      CALCULATE A NEW HORIZONTAL SCALE FACTOR AND PLACE IN WFM HEADER
13793 *
13794 *      LEVEL 3 ROUTINE
13795 *
13796 *      PASS ON STACK ---
13797 *      FIRST POP IS WFM NUMBER TO MODIFY
13798 *      SECOND POP IS FP SCALE FACTOR TO SET IN WFM HEADER
13799 *
13800 6EAB      DBA0      NEWHSCL  WORD WPLVL3      10460
13801 6EAA      6EAC      WORD  **2      10461
13802 6EAC      C25D      0012      MOV 18(R13),SOFT  10462
13803 6EB0      C079      MOV *SOFT+,R1    10463
13804 6EB2      06A0      6984      BL  ADRWFM      10464
13805 6EB6      C0F9      MOV *SOFT+,R3    10465
13806 6EB8      C079      MOV *SOFT+,R1    10466
13807 6EBA      C060      33FE      MOV FP10M,R1    10467
13808 6EBE      C0A0      3340      MOV FP10E,R2    10468
13809 6EC2      0420      70AA      BLWP FPMPY      10 * HSCALE  10469
13810 6EC6      C060      0970      MOV RESOLV,R1   10470
13811 6ECA      06A0      757E      BL INT2FP      10471
13812 6ECE      0420      7050      BLWP FPDIV     10 * HSCALE / (RESOLV - 1)  10472
13813 6ED2      A0E0      3340      A  HEXP,R3     10473
13814 6ED6      CCF9      MOV *SOFT+,*R3+  10474
13815 6ED8      C4F9      MOV *SOFT+,*R3  10475
13816 6EDA      CB49      0012      MOV SOFT,18(R13) 10476
13817 6EDE      0380      RTWP          10477

```

13819			*						
13820			*	CALCULATE NEW VERTICAL SCALE FACTORS FOR DESIRED WFM					
13821			*	AND PLACE IT IN THE WFM HEADER					
13822			*						
13823			*	FIRST STACK POP IS WFM NUMBER TO MODIFY					
13824			*	SECOND POP IS VERTICAL SCALE DESIRED					
13825			*						
13826			*	VERTICAL SCALE FACTOR = SCALE * 20					
13827			*						
13828			*	LEVEL 3					
13829			*						
13830	6EE0			DBA0	NEWSCL	WORD	WPLVL3	10479	
13831	6EE2			6EE4		WORD	*+2	10480	
13832	6EE4	326D		0012		MOV	18(R13),SOFT	10481	
13833	6EE8	C079				MOV	*SOFT+,R1	10482	
13834	6EEA	06A0		6984		BL	ADRWFM	10483	
13835	6EEE	30F9				MOV	*SOFT+,R3	10484	
13836	6EF0	C079				MOV	*SOFT+,R1	10485	
13837	6EF2	C050		33F8		MOV	FP20M,R1	10486	
13838	6EF6	C0A0		33FA		MOV	FP20E,R2	10487	
13839	6EFA	0420		70AA		BLWP	FPMPY 20 * SCALE	10488	
13840	6EFE	A0E0		3338		A	VEXP,R3	10489	
13841	6F02	CCF9				MOV	*SOFT+,*R3+	10490	
13842	6F04	C4F9				MOV	*SOFT+,*R3	10491	
13843	6F06	C849		0012		MOV	SOFT,18(R13)	10492	
13844	6F0A	0380				RTWP		10493	

13846				*			
13847				*	I2F2	---	CONVERT INTEGER TO FP ON LEVEL 4
13848				*			
13849				*		LEVEL 4	
13850				*			
13851		6F0C			I2F2	EQU *	10495
13852	6F0C		DBC0			WORD WPLVL4	10496
13853	6F0E		6F10			WORD *+2	10497
13854	6F10	C26D	0012		MOV	19(R13),SOFT	GET SOFTSTACK POINTER FROM CALLER
13855	6F14	C06D	0002		MOV	2(R13),R1	GET CALLER'S R1
13856	6F18	06A0	757E		BL	INT2FP	
13857	6F1C	CB41	0002		MOV	R1,2(R13)	PUT RESULT IN CALLER'S R1 AND R2
13858	6F20	CB42	0004		MOV	R2,4(R13)	
13859	6F24	0380			RTWP		10503

```

13861 *
13862 * ROUTINE TO INITIALIZE A WAVEFORM
13863 *
13864 * LEVEL 5 ROUTINE
13865 *
13866 * INPUT: WFM # ON SOFTSTACK
13867 * OUTPUT: WFM #N INITIALIZED
13868 * DESTROYS: R7,R8,R12
13869 *
13870 6F26 C1D9 NJLLWFM MOV *SOFT,R7 POP WFM # 10505
13871 6F28 C648 MOV R11,*SOFT SAVE RETURN ADDRESS 10506
13872 6F2A D649 DECF SOFT 10507
13873 6F2C C641 MOV R1,*SOFT SAVE R1 10508
13874 6F2E C047 MOV R7,R1 10509
13875 6F30 B6A0 6984 BL ADRWFM GET HEADER AND DATA ADDRESS FOR WFM #N 10510
13876 6F34 0207 343A LI R7,NULLHEAD ADDRESS OF INITIALIZED HEADER 10511
13877 6F38 C239 MOV *SOFT+,R8 HEADER ADDRESS 10512
13878 6F3A 020C 0010 LI R12,16 16 WORDS IN HEADER 10513
13879 6F3E CE37 MOV *R7+,*R8+ 10514
13880 6F40 D60C DEC R12 10515
13881 6F42 15FD JGT *-4 10516
13882 6F44 31F9 MOV *SOFT+,R7 DATA ADDRESS 10517
13883 6F46 C220 D978 MOV RESOLV,R8 POINTS/WAVEFORM 10518
13884 6F4A 04F7 CLR *R7+ SET WAVEFORM TO 0 10519
13885 6F4C 0608 DEC R8 10520
13886 6F4E 15FD JGT *-4 10521
13887 6F50 C079 MOV *SOFT+,R1 RESTORE R1 10522
13888 6F52 C2F9 MOV *SOFT+,R11 RESTORE RETURN ADDRESS 10523
13889 6F54 045B B *R11 10524
13890 *DCCM
    
```

13893	*****							
13894	**					**		
13895	**				ADD/SUBTRACT TWO FLOATING POINT NUMBERS (B=B+A,B=B-A)	**		
13896	**					**		
13897	**				LEVEL 4 ROUTINE	**		
13898	**					**		
13899	**				INPUT- A IN R1,R2	**		
13900	**				B ON SOFTSTK	**		
13901	**				OUTPUT- RESULT ON SOFTSTK	**		
13902	**					**		
13903	**				STACK OPERATIONS-	**		
13904	**				SOFTSTK- POPS 2 PUSHES 2	**		
13905	**					**		
13906	**				NOTE ---	**		
13907	**				USE FPADD & FPSUB FOR NORMALIZED RESULT.	**		
13908	**				USE FPADDZ & FPSUBZ FOR RESULT WITH EXPONENT ZERO.	**		
13909	**					**		
13910	**				STATUS BITS AFFECTED- A, Z, C, O	**		
13911	**					**		
13912	*****							
13913	6F56		DBC0	FPSUBZ	WORD	WPLVL4	LEVEL 4 ROUTINE	10527
13914	6F59		6F5A		WORD	*+2		10528
13915	6F5A	020C	71E4	LI	R12,NRMLIZZ		LOAD ZERO EXPONENT ROUTINE POINTER	10529
13916	6F5E	1004		JMP	SUBZSU3			10530
13917	6F60		DBC0	FPSJ3	WORD	WPLVL4	LEVEL 4 ROUTINE	10531
13918	6F62		6F64		WORD	*+2		10532
13919	6F64	020C	713C	LI	R12,NRMLIZ		LOAD NORMALIZE ROUTINE POINTER	10533
13920	6F68	C0AD	0004	SJBZSJ3	MOV	4(R13),R2	GET EXPONENT OF ARGUMENT A	10534
13921	6F6C	C06D	0002		MOV	2(R13),R1	GET MANTISSA OF ARGUMENT A	10535
13922	6F70	0501			NEG	R1	SUBTRACTION IS JUST NEGATE AND ADD	10536
13923	6F72	1910			JND	ADDSUB		10537
13924	6F74	0582			INC	R2	IF MANTISSA WAS \$8000 SHIFT TO \$4000 AND	10538
13925	6F76	0911			SRL	R1,1	INCREMENT EXPONENT	10539
13926	6F78	100D			JMP	ADDSUB		10540
13927	6F7A		DBC0	FPAJZ	WORD	WPLVL4	LEVEL 4 ROUTINE	10541
13928	6F7C		6F7E		WORD	*+2		10542
13929	6F7E	020C	71E4	LI	R12,NRMLIZZ		LOAD ZERO EXPONENT ROUTINE POINTER	10543
13930	6F82	1004		JMP	ADDZADD			10544
13931	6F84		DBC0	FPAJZ	WORD	WPLVL4	LEVEL 4 ROUTINE	10545
13932	6F86		6F88		WORD	*+2		10546
13933	6F88	020C	713C	LI	R12,NRMLIZ		LOAD NORMALIZE ROUTINE POINTER	10547
13934	6F8C	C0AD	0004	ADDZADD	MOV	4(R13),R2	GET EXPONENT OF ARGUMENT A	10548
13935	6F90	C06D	0002		MOV	2(R13),R1	GET MANTISSA OF ARGUMENT A	10549
13936	6F94	1351		ADDSU3	JEQ	LOAD3	IF ARGUMENT A IS ZERO RESULT IS 3	10550
13937	6F96	C26D	0012		MOV	18(R13),SOFT	GET SOFTSTACK POINTER FROM CALLING ROUTINE	10551
13938	6F9A	C0F9			MOV	*SOFT+,R3	GET MANTISSA OF ARGUMENT B	10552
13939	6F9C	1346			JEQ	LOADA	IF ARGUMENT B IS ZERO RESULT IS Z	10553
13940	6F9E	C019			MOV	*SOFT,R0	GET EXPONENT OF ARGUMENT 3	10554
13941	6FA0	6002			S	R2,RJ	SUBTRACT EXPONENTS TO GET DIFFERENCE	10555
13942	6FA2	1902			JND	*+6	CHECK FOR OVERFLOW	10556
13943	6FA4	1842			JOC	LOADA		10557
13944	6FA6	1748			JNC	LOAD3		10558
13945	6FA8	0740			ABS	R0	TAKE ABSOLUTE VALUE OF EXPONENT DIFFERENCE	10559
13946	6FAA	1534			JGT	EXPBGTA	IF VALUE WAS POSITIVE THEN B > A	10560
13947	6FAC	1127			JLT	EXPAGT3	IF VALUE WAS NEGATIVE THEN A > B	10561
13948	6FAE	04C4			CLR	R4	CLEAR SECOND WORD OF MANTISSA	10562
13949	6FB0	A0C1		ADDITION	A	R1,R3	ADD MANTISSA'S	10563
13950	6FB2	1915			JND	ASNRMLZ	CHECK FOR OVERFLOW AFTER ADDITION	10564
13951	6FB4	181B			JOC	RSLTNEG	IF CARRY RESULT SHOULD BE NEGATIVE	10565

ADDITION/SUBTRACTION

10526

13952	6FB6	0914		SRL	R4,1	IF NO CARRY RESULT SHOULD BE POSITIVE	10566	
13953	6FB8	0913		SRL	R3,1	SHIFT MANTISSA RIGHT ONE POSITION	10567	
13954	6FBA	1702		JNC	*+6		10568	
13955	6FBC	0224	8000	AI	R4,\$8000		10569	
13956	6FC0	0599		OVFAJJ	INC	*SOFT	INCREMENT EXPONENT TO MATCH SHIFT	10570
13957	6FC2	1900		JNO	ASNRMLZ		10571	
13958	6FC4	0619		DEC	*SOFT	EXPONENT OVERFLOW SO SET BACK TO LARGEST	10572	
13959	6FC6	0201	7FFF	LI	R1,\$7FFF		10573	
13960	6FCA	02CF		STST	R15	SET STATUS OF RESULT	10574	
13961	6FCC	1501		JGT	*+4		10575	
13962	6FCE	0501		NEG	R1	RESULT SHOULD BE NEGATIVE	10576	
13963	6FD0	024F	E7FF	ANDI	R15,\$E7FF	MASK OFF CARRY AND OVFL BITS	10577	
13964	6FD4	A3E0	3406	A	OV,R15	SET THE OVFL BIT	10578	
13965	6FD8	0649		DECT	SOFT		10579	
13966	6FDA	C641		MOV	R1,*SOFT	STORE LARGEST MANTISSA	10580	
13967	6FDC	0380		RTW ^D			10581	
13968	6FDE	0649		ASNRMLZ	DECT	SOFT	PUT MANTISSA ON SOFTSTACK	10582
13969	6FE0	C644		MOV	R4,*SOFT	FOR NORMALIZE ROUTINES	10583	
13970	6FE2	0649		DECT	SOFT		10584	
13971	6FE4	C643		MOV	R3,*SOFT		10585	
13972	6FE6	069C		BL	*R12	BRANCH TO APPROPRIATE NORMALIZING ROUTINE	10586	
13973	6FE8	C3CC		MOV	R12,R15	MOVE STATUS OF RESULT TO R15	10587	
13974	6FEA	0380		RTW ^D		RETURN TO CALLING ROUTINE	10588	
13975	6FEC	0914		RSLTNEG	SRL	R4,1	SHIFT MANTISSA ONE POSITION	10589
13976	6FEE	0913		SRL	R3,1		10590	
13977	6FF0	1702		JNC	*+6		10591	
13978	6FF2	0224	8000	AI	R4,\$8000		10592	
13979	6FF6	0223	8000	AI	R3,\$8000	ADD NEGATIVE SIGN BIT	10593	
13980	6FFA	10E2		JMP	OVFAJJ		10594	
13981	6FFC	C103		EXPASTB	MOV	R3,R4	MOVE MANTISSA TO R3 AND R4	10595
13982	6FFE	0280	0010	CI	R0,15	CHECK IF SHIFT COUNT IS MORE THAN 15	10596	
13983	7002	1513		JGT	LOADA	IS YES, JUST LOAD A AS RESULT	10597	
13984	7004	C642		MOV	R2,*SOFT	MOVE EXPONENT OF ARGUMENT A TO RESULT	10598	
13985	7006	0803		SRA	R3,R0	SHIFT MANTISSA BY R0	10599	
13986	7008	0220	FFF0	AI	R0,-16		10600	
13987	700C	13D1		JEQ	ADDITION	CHECK FOR R0 = 16	10601	
13988	700E	0500		NEG	R0		10602	
13989	7010	0A04		SLA	R4,R0		10603	
13990	7012	10CE		JMP	ADDITION		10604	
13991	7014	C101		EXP3GTA	MOV	R1,R4	MOVE MANTISSA TO R1 AND R4	10605
13992	7016	0280	0010	CI	R0,15	CHECK IF SHIFT COUNT IS MORE THAN 15	10606	
13993	701A	150E		JGT	LOAD3	IF YESM JUST LOAD B AS RESULT	10607	
13994	701C	0801		SRA	R1,R0	SHIFT MANTISSA BY R0	10608	
13995	701E	0220	FFF0	AI	R0,-16		10609	
13996	7022	13C6		JEQ	ADDITION	CHECK FOR R0 = 16	10610	
13997	7024	0500		NEG	R0		10611	
13998	7026	0A04		SLA	R4,R0		10612	
13999	7028	18C3		JMP	ADDITION		10613	
14000	702A	C642		LJAJA	MOV	R2,*SOFT	MOVE ARGUMENT A TO RESULT	10614
14001	702C	0649		DECT	SOFT		10615	
14002	702E	C641		MOV	R1,*SOFT		10616	
14003	7030	02CF		STST	R15	SET STATUS OF RESULT	10617	
14004	7032	024F	E7FF	ANDI	R15,\$E7FF	SET OVERFLOW AND CARRY TO 0	10618	
14005	7036	0380		RTW ^D		RETURN TO CALLING ROUTINE	10619	
14006	7038	C26D	0012	LJAD3	MOV	16(R13),SOFT	RESET SOFTSTACK POINTER	10620
14007	703C	C659		MOV	*SOFT,*SOFT		10621	
14008	703E	02CF		STST	R15	SET STATUS OF RESULT	10622	
14009	7040	024F	E7FF	ANDI	R15,\$E7FF	SET OVERFLOW AND CARRY TO 0	10623	
14010	7044	0380		RTW ^D		RETURN TO CALLING ROUTINE	10624	


```

14012 *****
14013 **
14014 ** MULTIPLY/DIVIDE TWO FLOATING POINT NUMBERS (B=B*A,B=B/A) **
14015 **
14016 ** LEVEL 4 ROUTINE **
14017 **
14018 ** INPJT- A IN R1,R2 **
14019 ** B ON SOFTSTK **
14020 ** OUTPUT- RESULT ON SOFTSTK **
14021 **
14022 ** STACK OPERATIONS- **
14023 ** SOFTSTK- POPS 2 PUSHES 2 **
14024 **
14025 ** NOTE --- **
14026 ** USE FMPY & FPDIV FOR NORMALIZED RESULT. **
14027 ** USE FMPYZ & FPDIVZ FOR RESULT WITH EXPONENT ZERO. **
14028 **
14029 ** STATUS BITS AFFECTED- A, Z, C, O **
14030 **
14031 *****
    
```

14032	7046		D3C0	FPDIVZ	WORD	WPLVL4	LEVEL 4 ROUTINE	10626
14033	7048		704A		WORD	*+2		10627
14034	704A	023C	71E4		LI	R12,NRMLIZZ	LOAD POINTER TO ZERO EXPONENT ROUTINE	10628
14035	704E	1004			JMP	DIVZDIV		10629
14036	7050		D3C0	FPDIV	WORD	WPLVL4	LEVEL 4 ROUTINE	10630
14037	7052		7054		WORD	*+2		10631
14038	7054	020C	713C		LI	R12,NRMLIZ	LOAD POINTER TO NORMALIZE ROUTINE	10632
14039	7058	C26D	0012	DIVZDIV	MOV	16(R13),SOFT	GET CURRENT SOFTSTACK POINTER	10633
14040	705C	04C0			CLR	R0	CLEAR SIGN FLAG	10634
14041	705E	C0A9	0004		MOV	4(R13),R2	GET EXPONENT OF ARGUMENT A	10635
14042	7062	C06D	0002		MOV	2(R13),R1	GET MANTISSA OF ARGUMENT A AND	10636
14043	7066	1506			JGT	NRMLA	CHECK ITS SIGN	10637
14044	7068	1359			JEQ	DIVBY0	IF A = 0 THEN DIVISION BY 0	10638
14045	706A	0580			INC	R0	IF NEGATIVE, NEGATE TO GET POSITIVE	10639
14046	706C	0501			NEG	R1	& INCREMENT SIGN FLAG	10640
14047	706E	1902			JNO	NRMLA	IF NEGATION CAUSED OVERFLOW VALUE OF	10641
14048	7070	0911			SRL	R1,1	MANTISSA WAS \$8000 SO SHIFT TO \$4000 AND	10642
14049	7072	0582			INC	R2	INCREMENT EXPONENT	10643
14050	7074	0602		NRMLA	DEC	R2		10644
14051	7076	0A11			SLA	R1,1	NORMALIZE ARGUMENT A	10645
14052	7078	19FD			JNO	NRMLA		10646
14053	707A	0911			SRL	R1,1		10647
14054	707C	0582			INC	R2		10648
14055	707E	0502			NEG	R2	INVERT ARGUMENT A	10649
14056	7080	C203	4000		LI	R3,\$4000	LOAD R3,R4 WITH A 1	10650
14057	7084	04C4			CLR	R4		10651
14058	7086	3CC1			DIV	R1,R3	DIVIDE MANTISSA INTO 1	10652
14059	7088	1902			JNO	*+6		10653
14060	708A	0A13			SLA	R3,1	HANDLE 1/1 INDIVIDUALLY	10654
14061	708C	0582			INC	R2		10655
14062	708E	C043			MOV	R3,R1	PUT RESULT IN R1,R2	10656
14063	7090	0582			INC	R2		10657
14064	7092	0911			SRL	R1,1	MAKE VALUE POSITIVE	10658
14065	7094	171C			JNC	MARGAOK		10659
14066	7096	0581			INC	R1	ROUND RESULT	10660
14067	7098	191A			JNO	MARGAOK		10661
14068	709A	0911			SRL	R1,1		10662
14069	709C	0582			INC	R2		10663
14070	709E	1017			JMP	MARGAOK		10664

14071	70A0		DBC0	FMPYZ	WORD	WPLVL4	LEVEL 4 ROUTINE	10665
14072	70A2		70A4		WORD	*+2		10666
14073	70A4	020C	71E4		LI	R12,NRMLIZZ	LOAD POINTER TO ZERO EXPONENT ROUTINE	10667
14074	70A8	1004			JMP	MPYZMPY		10668
14075	70AA		DBC0	FMPY	WORD	WPLVL4	LEVEL 4 ROUTINE	10669
14076	70AC		70AE		WORD	*+2		10670
14077	70AE	020C	713C		LI	R12,NRMLIZ	LOAD POINTER TO NORMALIZE ROUTINE	10671
14078	7092	C260	0012	MPYZMPY	MOV	18(R13),SOFT	GET SOFTSTACK POINTER	10672
14079	70B6	04C0			CLR	R0	CLEAR SIGN FLAG	10573
14080	70B8	C0AD	0004		MOV	4(R13),R2	GET EXPONENT OF ARGUMENT A	10674
14081	70BC	C060	0002		MOV	2(R13),R1	GET MANTISSA FROM CALLER'S WORKSPACE	10675
14082	70C0	1506			JST	MARGAOK	AND CHECK IT'S SIGN	10676
14083	70C2	1333			JEQ	MPYBY0	IF A = 0, RESULT = 0	10677
14084	70C4	0580			INC	R0	IF NEGATIVE, NEGATE TO GET POSITIVE	10678
14085	70C6	0501			NEG	R1	& INCREMENT SIGN FLAG	10679
14086	70C8	1902			JNO	MARGAOK	IF NEGATION CAUSED OVERFLOW VALUE OF	10680
14087	70CA	0911			SRL	R1,1	MANTISSA WAS \$8000 SO SHIFT TO \$4000 AND	10681
14088	70CC	0582			INC	R2	INCREMENT EXPONENT	10582
14089	70CE	04C6		MARGAOK	CLR	R6	SET ALL STATUS BITS TO ZERO	10683
14090	70D0	C0F9			MOV	*SOFT+,R3	CHECK SIGN OF EACH ARGUMENT AS	10684
14091	70D2	1506			JGT	MARG30K	INSTRUCTION 'MPY' WORKS ONLY ON UNSIGNED	10685
14092	70D4	132B			JEQ	MPYDIV0		10686
14093	70D6	0580			INC	R0	16 BIT MAGNITUDE VALUES. IF AN ARGUMENT	10687
14094	70D8	0503			NEG	R3	IS NEGATIVE, INCREMENT SIGN FLAG AND	10688
14095	70DA	1902			JNO	MARG30K	CHECK FOR A MANTISSA OF \$8000	10689
14096	70DC	0913			SRL	R3,1	IF FOUND, SHIFT TO \$4000 AND INCREMENT	10690
14097	70DE	0599			INC	*SOFT	EXPONENT OF ARGUMENT B	10591
14098	70E0	A642		MARG30K	A	R2,*SOFT	ADD EXPONENTS	10692
14099	70E2	1901			JNO	*+4	CHECK FOR UNDERFLOW/OVERFLOW	10593
14100	70E4	1012			JMP	MOVRFLW		10694
14101	70E6	0599			INC	*SOFT	INCREMENT EXPONENT TO ADJUST BINARY POINT	10695
14102	70E8	1901			JNO	*+4	BECAUSE 'MPY' RESULTS IN TWO BINARY	10696
14103	70EA	100F			JMP	MOVRFLW	POSITIONS TO LEFT OF BINARY POINT.	10597
14104	70EC	3043			MPY	R3,R1	MULTIPLY MANTISSAS	10598
14105	70EE	0600		MADJSGN	DEC	R0	CORRECT SIGN OF DOUBLE-WORD MANTISSA	10599
14106	70F0	1604			JNE	MNRMLZ	RESULT IS NEGATIVE ONLY IF R0=1	10700
14107	70F2	0541			INV	R1		10701
14108	70F4	0502			NEG	R2	NEGATE DOUBLE-WORD MANTISSA IN R1,R2	10702
14109	70F6	1701			JNC	MNRMLZ		10703
14110	70F8	0581			INC	R1		10704
14111	70FA	0649		MNRMLZ	DECT	SOFT		10705
14112	70FC	C642			MOV	R2,*SOFT	PUT RESULT ON STACK FOR 'NRMLIZ'	10706
14113	70FE	0649			DECT	SOFT		10707
14114	7100	C641			MOV	R1,*SOFT		10708
14115	7102	069C			BL	*R12	BRANCH TO APPROPRIATE NORMALIZE ROUTINE	10709
14116	7104	C3CC			MOV	R12,R15	MOVE STATUS OF RESULT TO R15	10710
14117	7106	E3C6			SJC	R6,R15	ADD STATUS BITS (C,0)	10711
14118	7108	0380			RTW ^p		RETURN TO CALLING ROUTINE	10712
14119	710A	1810		MOVRFLW	JOC	MPYDIV0	IF CARRY STATUS BIT IS SET THEN UNDERFLOW	10713
14120	710C	E1AD	3406		SOC	OV,R6	SET OVERFLOW STATUS BIT	10714
14121	7110	04C2			CLR	R2		10715
14122	7112	C060	338E		MOV	CH7FFF,R1	LOAD MAXIMUM VALUE	10716
14123	7116	C660	338E		MOV	CH7FFF,*SOFT		10717
14124	711A	10E9			JMP	MADJSGN		10718
14125	711C	04E0	D94C	DIV3YJ	CLR	WARNING	GIVE WARNING IF DIVIDE BY 0 ATTEMPTED	10719
14126	7120	CE59			MOV	*SOFT,*SOFT+	CHECK SIGN OF ARGUMENT B	10720
14127	7122	15F4			JGT	MOVRFLW+2		10721
14128	7124	1303			JEQ	MPYDIV0	IF ZERO LOAD ZERO	10722
14129	7126	0580			INC	R0	IF NEGATIVE INCREMENT SIGN FLAG	10723

14130	7128	10F1	JMP	MOVFLW+2		10724	
14131	712A	05C9	MPY3Y0	INCT	SOFT	POINT TO EXPONENT OF B	10725
14132	712C	04C1	MPYJIV0	CLR	R1	LOAD RESULT OF 0	10726
14133	712E	04C2		CLR	R2		10727
14134	7130	04D9		CLR	*SOFT		10728
14135	7132	10E3	JMP	MNRMLZ			10729

14137				*****		
14138				**		**
14139				**	NORMALIZE A FLOATING POINT NUMBER	**
14140				**		**
14141				**	LEVEL 5 ROUTINE	**
14142				**		**
14143				**	INPUT- FLOATING POINT NUMBER ON SOFTSTACK	**
14144				**	NRMCHK - 1 WORD MANTISSA, 1 WORD EXPONENT	**
14145				**	NRMLIZ - 2 WORD MANTISSA, 1 WORD EXPONENT	**
14146				**	OUTPUT- NORMALIZED FLOATING POINT NUMBER ON	**
14147				**	SOFTSTACK.	**
14148				**	1 WORD MANTISSA, 1 WORD EXPONENT	**
14149				**		**
14150				**	INPUT- SOFT	**
14151				**	OUTPJT- SOFT	**
14152				**	DESTROYS- R7,R8,R12	**
14153				**		**
14154				**	STACK OPERATIONS-	**
14155				**	SOFTSTK- NRMCHK - POPS 2 PUSHES 2	**
14156				**	NRMLIZ - POPS 3 PUSHES 2	**
14157				**		**
14158				*****		
14159	7134	C1F9		NRMCHK	MOV *SOFT+,R7	MOVE SINGLE WORD MANTISSA TO R7 10731
14160	7136	1311			JEQ NRMLZ0	CHECK IF INPUT ARGUMENT IS ZERO 10732
14161	7138	04C8			CLR R8	CREATE DOUBLE WORD MANTISSA 10733
14162	713A	1026			JMP NRMLZ+2	CONTINUE PROCESSING AS IF NRMLIZ WAS CALLED 10734
14163	713C	C1F9		NRMLIZ	MOV *SOFT+,R7	MOVE FIRST WORD OF MANTISSA TO R7 10735
14164	713E	160F			JNE CHKFFFF	IF FIRST WORD <> 0, CHECK IF = \$FFFF 10736
14165	7140	C1F9			MOV *SOFT+,R7	OTHERWISE MOVE SECOND WORD TO FIRST WORD 10737
14166	7142	130B			JEQ NRMLZ0	& CHECK IF INPUT WAS ZERO. 10738
14167	7144	151A			JGT ADJEXP	
14168	7146	0917			SRL R7,1	IF WORD SWITCH CAUSED SIGN CHANGE UNDO 10739
14169	7148	1704			JNC *+10	ONE SHIFT AND ROUND RESULT 10741
14170	714A	0587			INC R7	
14171	714C	1902			JNO *+6	CHECK FOR ROUND FROM \$7FFF TO \$8000 10743
14172	714E	0917			SRL R7,1	IF FOUND SHIFT MANTISSA TO \$4000 AND 10744
14173	7150	0599			INC *SOFT	INCREMENT EXPONENT 10745
14174	7152	6660	3356		S C15,*SOFT	ADJUST EXPONENT TO MATCH SHIFTING 10746
14175	7156	1931			JNO NRMLZED+2	CHECK FOR OVERFLOW 10747
14176	7158	1039			JMP NOVRFLW	
14177	715A	0409		NRMLZ0	CLR *SOFT	IF ARGUMENT IS ZERO RETURN NORMALIZED ZERO. 10749
14178	715C	102E			JMP NRMLZED+2	
14179	715E	0287	FFFF	CHKFFFF	CI R7,\$FFFF	IF FIRST WORD <> \$FFFF NORMALIZE ARGUMENT 10751
14180	7162	1611			JNE NRMLZ	OTHERWISE MOVE SECOND WORD TO FIRST WORD 10752
14181	7164	C1F9			MOV *SOFT+,R7	& IF NO SIGN CHANGE CONTINUE NORMALLY 10753
14182	7166	1109			JLT ADJEXP	
14183	7168	0917			SRL R7,1	IF SIGN CHANGE, UNDO ONE SHIFT 10755
14184	716A	1701			JNC *+4	
14185	716C	0587			INC R7	
14186	716E	0227	8000	AI	R7,\$8000	MAKE VALUE NEGATIVE 10758
14187	7172	6660	3356		S C15,*SOFT	ADJUST EXPONENT TO MATCH SHIFTING 10759
14188	7176	1921			JNO NRMLZED+2	CHECK FOR OVERFLOW 10760
14189	7178	1029			JMP NOVRFLW	
14190	717A	6660	3358	ADJEXP	S C16,*SOFT	ADJUST EXPONENT TO MATCH SHIFTING 10762
14191	717E	1901			JNO *+4	CHECK FOR OVERFLOW 10763
14192	7180	1025			JMP NOVRFLW	
14193	7182	04C8			CLR R8	SET SECOND WORD OF MANTISSA TO ZERO 10765
14194	7184	1001			JMP NRMLZ+2	
14195	7186	C239		NRMLZ	MOV *SOFT+,R8	GET SECOND WORD OF MANTISSA 10767

NORMALIZATION

10730

14196	7188	C300		MOV	R0,R12	SAVE REGISTER 0 AS IT CAN'T BE DESTROYED	10768	
14197	718A	0420		CLR	R0	SET SHIFT COUNT TO ZERO	10769	
14198	718C	0600		DEC	R0	DECREMENT SHIFT COUNT FOR EACH SHIFT REQUIRED	10770	
14199	718E	0A17		SLA	R7,1	TO NORMALIZE MANTISSA	10771	
14200	7190	19FD		JNO	*-4	SHIFTING STOPS WHEN OVERFLOW OCCURS	10772	
14201	7192	1701		JNC	*+4	UNDO LAST SHIFT	10773	
14202	7194	0587		INC	R7	IF A 1 WAS SHIFTED OUT PUT IT BACK	10774	
14203	7196	0B17		SRC	R7,1		10775	
14204	7198	0220	0011	AI	R0,17	DETERMINE SHIFT COUNT FOR SECOND WORD OF	10776	
14205	719C	0908		SRL	R0,R0	MANTISSA	10777	
14206	719E	1701		JNC	*+4	ROUND NORMALIZED RESULT	10778	
14207	71A0	0588		INC	R0		10779	
14208	71A2	A1C8		A	R0,R7	ADD BITS FROM SECOND WORD TO FIRST WORD	10780	
14209	71A4	1904		JNO	*+10	CHECK IF OVERFLOW RESULTED FROM ROUNDING	10781	
14210	71A6	1701		JNC	*+4	IF IT DID, DIVIDE MANTISSA BY 2 AND INCREMENT	10782	
14211	71A8	0587		INC	R7	EXPONENT. SHIFT BIT IN TO KEEP SAME SIGN	10783	
14212	71AA	0B17		SRC	R7,1		10784	
14213	71AC	0580		INC	R0		10785	
14214	71AE	0220	FFF0	AI	R0,-16	RESET R0 TO EXPONENT ADJUSTMENT	10786	
14215	71B2	A648		A	R0,*SOFT	ADJUST EXPONENT TO ACCOUNT FOR SHIFTING	10787	
14216	71B4	1901		JNO	NRMLZED	CHECK FOR OVERFLOW	10788	
14217	71B6	100A		JMP	NOVRFLW		10789	
14218	71B8	C00C		MOV	R12,R0	RESTORE REGISTER 0	10790	
14219	71BA	0287	8000	CI	R7,\$8000	CHECK FOR \$8000 MANTISSA	10791	
14220	71BE	1602		JNE	*+6	\$8000 SCREWS UP WITH 'ABS' & 'NEG'	10792	
14221	71C0	0B17		SRA	R7,1	SHIFT TO \$C000	10793	
14222	71C2	0599		INC	*SOFT	ADJUST EXPONENT TO MATCH MANTISSA SHIFT	10794	
14223	71C4	0649		DECT	SOFT		10795	
14224	71C6	0647		MOV	R7,*SOFT	PUT SINGLE WORD MANTISSA BACK ON STACK	10796	
14225	71C8	02CC		STST	R12	PUT STATUS OF ARGUMENT IN R12	10797	
14226	71CA	045B		B	*R11	RETURN TO CALLING ROUTINE	10798	
14227	71CC	C00C		NOVRFLW	MOV	R12,R0	RESTORE REGISTER 0	10799
14228	71CE	0409		CLR	*SOFT	LOAD MINIMUM FLOATING POINT NUMBER	10800	
14229	71D0	04C7		CLR	R7		10801	
14230	71D2	02CC		STST	R12	GET STATUS OF NORMALIZED ARGUMENT	10802	
14231	71D4	E320	3408	SOC	COV,R12	ADD CARRY & OVERFLOW STATUS	10803	
14232	71D8	0649		DECT	SOFT		10804	
14233	71DA	0647		MOV	R7,*SOFT	PUT MANTISSA ON STACK	10805	
14234	71DC	045B		B	*R11	RETURN TO CALLING ROUTINE	10806	

14236				*****		
14237				**		**
14238				**	CREATE FLOATING POINT NUMBER WITH A ZERO EXPONENT	**
14239				**		**
14240				**	LEVEL 5 ROUTINE	**
14241				**		**
14242				**	INPUT- FLOATING POINT NUMBER ON SOFTSTACK.	**
14243				**	NRMCHKZ - 1 WORD MANTISSA, 1 WORD EXPONENT**	**
14244				**	NRMLIZZ - 2 WORD MANTISSA, 1 WORD EXPONENT**	**
14245				**	OUTPUT- FLOATING POINT NUMBER WITH AN EXPONENT	**
14246				**	OF ZERO ON SOFTSTACK.	**
14247				**	1 WORD MANTISSA, 1 WORD EXPONENT	**
14248				**		**
14249				**	INPUT- SOFT	**
14250				**	OUTPUT- SOFT	**
14251				**	DESTROYS- R0,R7,R8,R12	**
14252				**		**
14253				**	STACK OPERATIONS-	**
14254				**	SOFTSTK- NRMCHKZ - POPS 2 PUSHES 2	**
14255				**	NRMLIZZ - POPS 3 PUSHES 2	**
14256				**		**
14257				*****		*****
14258	71DE	C1F9		NRMCHKZ	MOV *SOFT+,R7	GET FIRST WORD OF MANTISSA 10808
14259	71E0	04C8			CLR R8	DEFAULT SECOND WORD TO \$0000 10809
14260	71E2	1002			JMP *+6	10810
14261	71E4	C1F9		NRMLIZZ	MOV *SOFT+,R7	GET FIRST WORD OF MANTISSA 10811
14262	71E6	C239			MOV *SOFT+,R8	GET SECOND WORD OF MANTISSA 10812
14263	71E8	04CC			CLR R12	SET STATUS TO 0 10813
14264	71EA	0287	0000		CI R7,0	CHECK IF FIRST WORD IS \$0000 10814
14265	71EE	130E			JEQ FIRST0	YES, THIS IS JUST THE SIGN EXTENDED 10815
14266	71F0	0287	FFFF		CI R7,\$FFFF	CHECK IF FIRST WORD IS \$FFFF 10816
14267	71F4	1616			JNE ZEROEXP	YES, THIS IS JUST THE SIGN EXTENDED 10817
14268	71F6	C1C8		FIRST0	MOV R8,R7	SHIFT MANTISSA LEFT 16 10818
14269	71F8	04C8			CLR R8	SET SECOND WORD TO 0 10819
14270	71FA	1111			JLT SIGNJK	IF SIGN IS STILL SAME OK 10820
14271	71FC	0917			SRL R7,1	IF SIGN CHANGE, SHIFT SHOULD BE 15 10821
14272	71FE	1702			JNC *+6	CHECK FOR CARRY OUT 10822
14273	7200	0208	8000		LI R8,\$8000	PUT CARRY INTO SECOND WORD 10823
14274	7204	0599			INC *SOFT	ADJUST EXPONENT 10824
14275	7206	0227	8000		AI R7,\$8000	MAKE FIRST WORD NEGATIVE AGAIN 10825
14276	720A	1009			JMP SIGNJK	10826
14277	720C	C1C8		FIRST0	MOV R8,R7	SHIFT MANTISSA LEFT 16 10827
14278	720E	04C8			CLR R8	SET SECOND WORD TO 0 10828
14279	7210	1506			JGT SIGNJK	IF SIGN CHANGE, SHIFT SHOULD BE 15 10829
14280	7212	133A			JEQ ZNRMLZ0	IF 0, THEN RESULT IS 0 10830
14281	7214	0917			SRL R7,1	IF SIGN CHANGE, SHIFT SHOULD BE 15 10831
14282	7216	1702			JNC *+6	CHECK FOR CARRY OUT 10832
14283	7218	0208	8000		LI R8,\$8000	PUT CARRY INTO SECOND WORD 10833
14284	721C	0599			INC *SOFT	ADJUST EXPONENT 10834
14285	721E	6668	3358	SIGNJK	S C16,*SOFT	ADJUST EXPONENT 10835
14286	7222	3019		ZEROEXP	MOV *SOFT,R0	GET EXPONENT FROM CALLER 10836
14287	7224	150F			JGT ZNRMLSL	IF POSITIVE THEN SHIFT LEFT 10837
14288	7226	1104			JLT ZNRMLSR	IF NEGATIVE THEN SHIFT RIGHT 10838
14289	7228	0A18			SLA R8,1	IF ZERO ROUND AND WE'RE DONE 10839
14290	722A	172E			JNC ZNRMLZ0	10840
14291	722C	0587			INC R7	10841
14292	722E	102C			JMP ZNRMLZ0	10842
14293	7230	0500		ZNRMLSR	NEG R0	IF NEGATIVE SHIFT RIGHT 10843
14294	7232	0280	0010		CI R0,16	IF >15 JUST LOAD ZERO 10844

14295	7236	1102		JLT	*+6		10845
14296	7238	04C7		CLR	R7		10846
14297	723A	1026		JMP	ZNRMLZ0		10847
14298	723C	0807		SRA	R7,R0	IF <=15 SHIFT TO MAKE EXPONENT 0	10848
14299	723E	1724		JNC	ZNRMLZ0	ROUND RESULT	10849
14300	7240	0587		INC	R7		10850
14301	7242	1822		JMP	ZNRMLZ0		10851
14302	7244	0280	000F	ZNRMLSL	CI	R0,15	CHECK IF R0 > 15
14303	7248	1517		JGT	ZNRMOVR		IF SO, OVERFLOW HAS OCCURED
14304	724A	020C	8000	LI	R12,\$8000		CREATE MASK OF BITS TO BE SHIFTED OFF
14305	724E	080C		SRA	R12,R0		
14306	7250	4307		SZC	R7,R12	CHECK IF ALL 1'S WILL BE SHIFTED OFF	
14307	7252	1308		JEQ	LSHFTOK	YES, THEN SHIFT THEM	
14308	7254	020C	8000	LI	R12,\$8000	NO, CHECK IF ALL 0'S WILL BE SHIFTED OFF	
14309	7258	080C		SRA	R12,R0		
14310	725A	0547		INV	R7		
14311	725C	4307		SZC	R7,R12		
14312	725E	0547		INV	R7		
14313	7260	030C		MOV	R12,R12		
14314	7262	160A		JNE	ZNRMOVR	IF BOTH 0'S AND 1'S SHIFTED THEN OVERFLOW	
14315	7264	0A07		LSHFTOK	SLA	R7,RJ	SHIFT TO MAKE EXPONENT ZERO
14316	7266	04CC		CLR	R12		
14317	7268	0500		NEG	R0		
14318	726A	0220	0010	AI	R0,16		
14319	726E	0908		SR	R8,RJ	ADD BITS FROM R8	
14320	7270	1701		JNC	*+4	ROUND RESULT	
14321	7272	0588		INC	R8		
14322	7274	A1C8		A	R8,R7		
14323	7276	1008		JMP	ZNRMLZ0		
14324	7278	C320	3406	ZNRMOVR	MOV	OV,R12	
14325	727C	C207		MOV	R7,R8	SAVE SIGN OF MANTISSA	
14326	727E	0207	7FFF	LI	R7,\$7FFF	LOAD MAXIMUM VALUE	
14327	7282	C208		MOV	R8,R8	CHECK ORIGINAL SIGN	
14328	7284	1501		JGT	ZNRMLZ0		
14329	7286	0507		NEG	R7		
14330	7288	04D9		ZNRMLZ0	CLR	*SOFT	OTHERWISE ZERO EXPONENT AND
14331	728A	0649		DECT	SOFT	PUSH MANTISSA BACK ON STACK	
14332	728C	0647		MOV	R7,*SOFT	VALUE TO BE PUSHED IS IN R7	
14333	728E	02C7		STST	R7	SET STATUS BITS L>,A>,<=	
14334	7290	0247	E3FF	ANDI	R7,\$E3FF		
14335	7294	E307		SJC	R7,R12	ADD OVERFLOW & CARRY STATUS	
14336	7296	045B		B	*R11	RETURN TO CALLER	


```

14338 *****
14339 **
14340 ** CALCULATE NATURAL LOGARITHM OF X **
14341 **
14342 ** LEVEL 3 ROUTINE **
14343 **
14344 ** INPUT- X IS ON SOFTSTACK **
14345 ** OUTPUT- LN(X) ON SOFTSTACK **
14346 ** DESTROYS- R0-R8 **
14347 **
14348 ** STACK OPERATIONS- **
14349 ** SOFTSTK- POPS 2 PUSHES 2 **
14350 **
14351 ** NOTE --- **
14352 ** FOLLOWING ALGORITHM FOR LN(Y) IS USED- **
14353 ** 1) REPRESENT Y AS M*2**N, 0.5<= M <1 **
14354 ** 2) DEFINE T = (M-1/(2**5))/(M+1/(2**5)) **
14355 ** ABS(T)<0.172 1/(2**5)=0.707107 **
14356 ** 3) DEFINE W = 2*(T+T*(D*T**2-2**15)) **
14357 ** D=0.3392334 **
14358 ** 4) LN(Y) = (W-Q/2) + N*Q **
14359 ** WHERE Q APPROX LN(2) **
14360 ** 5) SPECIAL CASES- **
14361 ** Y<=0 OR Y=+INFINITY --> ERROR **
14362 ** .96875 <= Y <= 1.028 (DECIMAL) USE-- **
14363 ** LN(Y)=(Y-1)*(1+Z+Z*Z) **
14364 ** WHERE Z = -(Y-1)/2 **
14365 **
14366 *****
    
```

14367	7298		DBA0	F0NLOG	WORD	WPLV_L3	LEVEL 3 ROUTINE	10888
14368	729A		729C		WORD	*+2		10889
14369	729C	326D	0012		MOV	16(R13),SOFT	GET SOFTSTACK POINTER	10890
14370	72A0	06A0	7134		BL	NRMC+K	MAKE SJRE INPUT NUMBER IS NORMALIZED	10891
14371	72A4	C879			MOV	*SOFT+,R1	POP MANTISSA OFF SOFTSTACK	10892
14372	72A6	1131			JLT	LNXERR	LN(X<0) IS UNDEFINED	10893
14373	72A8	1333			JEQ	LNXERR1	LN(0) GIVES MAXIMUM NEGATIVE VALUE	10894
14374	72AA	C099			MOV	*SOFT,R2	POP EXPONENT OFF SOFTSTACK	10895
14375	72AC	0649			DECT	SOFT	REPOINT TO INPUT NUMBER	10896
14376	72AE	0282	0001		CI	R2,1		10897
14377	72B2	1602			JNE	LOG1		10898
14378	72B4	0A11			SLA	R1,1	ADJUST NUMBER IF EXPONENT=1	10899
14379	72B6	04C2			CLR	R2		10900
14380	72B8	C082		LOG1	MOV	R2,R2	M SCALE(15BP)	10901
14381	72BA	1633			JNE	NONSPCL		10902
14382	72BC	0281	7C00		CI	R1,\$7C00	IS < 0.96875	10903
14383	72C0	1A30			JL	NONSPCL		10904
14384	72C2	0281	8394		CI	R1,\$8394	IS >= 1.028	10905
14385	72C6	1B2D			JH	NONSPCL		10906
14386	72C8	0221	8000		AI	R1,\$8000	M-1 (15BP)	10907
14387	72CC	C101			MOV	R1,R4		10908
14388	72CE	0504			NEG	R4		10909
14389	72D0	C144			MOV	R4,R5	Z (16BP)	10910
14390	72D2	0814			SRA	R4,1	Z (15BP)	10911
14391	72D4	0224	8000		AI	R4,\$8000	1+Z (15BP)	10912
14392	72D8	0745			ABS	R5		10913
14393	72DA	3905			MPY	R5,R4	(1+Z)Z (15BP)	10914
14394	72DC	C041			MOV	R1,R1		10915
14395	72DE	1101			JLT	LOG2	SET SIGN	10916
14396	72E0	0504			NEG	R4	PRODUCT TO SIGN Z	10917

10887

14397	72E2	0224	0000	LOG2	AI	R4,\$0000	1+Z+Z*Z (15BP)	10918
14398	72E6	0584			INC	R4	ROUND UP ALWAYS	10919
14399	72E8	C141			MOV	R1,R5		10920
14400	72EA	0207	0001		LI	R7,1		10921
14401	72EE	0741			ABS	R1		10922
14402	72F0	3844			MPY	R4,R1	ABS(M-1)*(1+Z+Z*Z) (30BP)	10923
14403	72F2	0649		LOGP<1	DECT	SOFT		10924
14404	72F4	CE41			MOV	R1,*SOFT+	PUSH RESULT ONTO SOFTSTACK	10925
14405	72F6	CE42			MOV	R2,*SOFT+		10926
14406	72F8	C647			MOV	R7,*SOFT		10927
14407	72FA	0229	FFFC		AI	SOFT,-4	POINT BACK TO MANTISSA	10928
14408	72FE	06A0	713C		BL	NRMLIZ	NORMALIZE RESULT	10929
14409	7302	C145			MOV	R5,R5		10930
14410	7304	1501			JGT	*+4		10931
14411	7306	0519			NEG	*SOFT		10932
14412	7308	0380			RTWP			10933
14413	730A	04E0	094A	LNERR	CLR	FATAL		10934
14414	730E	0380			RTWP			10935
14415	7310	C660	338E	LNERR1	MOV	CH7FFF,*SOFT	LOAD MAXIMUM NEGATIVE VALUE	10936
14416	7314	0649			DECT	SOFT		10937
14417	7316	C660	338E		MOV	CH7FFF,*SOFT		10938
14418	731A	0519			NEG	*SOFT		10939
14419	731C	04E0	094C		CLR	WARNING		10940
14420	7320	0380			RTWP			10941
14421	7322	C079		NONSPCL	MOV	*SOFT+,R1	GET M (AGAIN)	10942
14422	7324	C0C1			MOV	R1,R3	M(15BP)	10943
14423	7326	60E0	340A		S	INVSQR2,R3	C1=.8504(16) (15BP)	10944
14424				**			C1 IS STORED IN INVSQR2 **	
14425				**			INVSQR2=\$5A82 **	
14426				**			C=.3505(16) **	
14427	732A	0A13			SLA	R3,1	M-C1 (16BP)	10945
14428	732C	0603			DEC	R3	M-C	10946
14429	732E	C143			MOV	R3,R5	SAVE SIGN	10947
14430	7330	0743			ABS	R3	PUT ABS(M-C)	10948
14431	7332	0A13			SLA	R3,1	INTO R3,R4	10949
14432	7334	04C4			CLR	R4	SCALED (33BP)	10950
14433	7336	A060	340A		A	INVSQR2,R1	M+C1 (15BP)	10951
14434	733A	3CC1			DIV	R1,R3	ABS T (10BP)	10952
14435	733C	C043			MOV	R3,R1		10953
14436	733E	C183			MOV	R3,R6		10954
14437	7340	38C1			MPY	R1,R3		10955
14438	7342	38C1			MPY	R1,R3	ABS(T**3) (22BP)	10956
14439	7344	38E0	340C		MPY	DCNST,R3	ABS(D*T**3) (21BP)	10957
14440				**			D IS STORED IN DCNST	
14441				**			D=0.5608(16) SCALED(15BP) **	
14442				**			DCNST=\$2B6C **	
14443	7348	09C6			SRL	R6,12	ABS(T*(2**15)) (21BP)	10958
14444	734A	60C5			S	R6,R3	ABS(T**3*D-(2**15)*T)	10959
14445	734C	0933			SRL	R3,3	SCALE (10BP)	10960
14446	734E	1701			JNC	LOG4	ROUND	10961
14447	7350	0583			INC	R3		10962
14448	7352	A0C1		LOG4	A	R1,R3	ABS W (17BP)	10963
14449				**			NEED Q/2 SCALED (17BP) **	
14450				**			Q=.3172(16) **	
14451				**			KEPT IN LN2 **	
14452				**			LN2=\$B172 **	
14453	7354	60E0	3426		S	LN2,R3	(17BP)	10964
14454	7358	0503			NEG	R3	(17BP)	10965
14455	735A	04C2			CLR	R2		10966

14456	735C	0913		SRL	R3,1	(Q/2-ABS(W)) (16BP)	10967
14457	735E	1702		JNC	LOG5		10968
14458	7360	0582		INC	R2		10969
14459	7362	0812		SRC	R2,1	DOUBLE PRECISION	10970
14460	7364	C145		LOG5	MOV	R5,R5	CHECK SIGN T
14461	7366	1518		JGT	LOG6		10972
14462	7368	14C6		CLR	R6		10973
14463	736A	0619		DEC	*SOFT	NEGATIVE T	10974
14464	736C	1903		JND	LOG13		10975
14465	736E	0201	A746	LI	R1,\$A746	SMALLEST ANS	10976
14466	7372	100F		JMP	LOG10		10977
14467	7374	1318		LOG13	JEQ	LOG8	E=0
14468	7376	0059		MOV	*SOFT,R1	E (0BP)	10979
14469	7378	0741		ABS	R1		10980
14470	737A	3860	3426	MPY	LN2,R1	Q (16BP)	10981
14471	737E	C659		MOV	*SOFT,*SOFT	GET SIGN E	10982
14472	7380	1504		JGT	LOG9	FIX SIGN PRODUCT	10983
14473	7382	0541		INV	R1		10984
14474	7384	0502		NEG	R2		10985
14475	7386	1701		JNC	LOG9		10986
14476	7388	0581		INC	R1		10987
14477	738A	A083		LOG9	A	R3,R2	ADD IN +OR- (Q/2-W)
14478	738C	1701		JNC	LOG7		10989
14479	738E	0586		INC	R6		10990
14480	7390	A046		LOG7	A	R6,R1	10991
14481	7392	0207	800F	LOG10	LI	R7,15	E=15
14482	7396	0205	0001	LOGPK2	LI	R5,1	10993
14483	739A	0649		DECT	SOFT		10994
14484	739C	10AA		JMP	LOGPK1		10995
14485	739E	0543		LOG5	INV	R3	T>0
14486	73A0	0502		NEG	R2	NEGATE (Q/2-W) DP	10997
14487	73A2	1701		JNC	*+4		10998
14488	73A4	0583		INC	R3		10999
14489	73A6	0706		SETJ	R6		11000
14490	73A8	C659		MOV	*SOFT,*SOFT		11001
14491	73AA	10E4		JMP	LOG13		11002
14492	73AC	C043		LOG5	MOV	R3,R1	E=0,DP USED
14493	73AE	0707		SETJ	R7	SCALE(32BP)	11004
14494	73B0	10F2		JMP	LOGPK2		11005

```

14496 *****
14497 **
14498 ** CALCULATE E**X **
14499 **
14500 ** LEVEL 3 ROUTINE **
14501 **
14502 ** INPUT- FLOATING POINT X ON SOFTSTK **
14503 ** OUTPJT- FLOATING POINT E**X ON SOFTSTK **
14504 ** DESTROYS- R0-R8 **
14505 **
14506 ** STACK OPERATIONS- **
14507 ** SOFTSTK- POPS 2 PUSHES 2 **
14508 **
14509 ** NOTE --- **
14510 ** FOLLOWING ALGORITHM WAS USED- **
14511 ** (PADE METHOD - RATIONAL-FRACTION APPROX.) **
14512 **
14513 ** 1) REPRESENT X AS ( N*LN(2) + R ) **
14514 ** N = X/LN(2) <ROUNDED OFF>: **
14515 ** ABS(R) < 0.5 **
14516 **
14517 ** 2) APPROXIMATE E**X AS --- **
14518 ** 2**N * (A+R*(B+C*R)) / (A-R*(B-C*R)) **
14519 ** WHERE A=12: B=6: & C=1 **
14520 **
14521 ** 3) ERROR FOR THIS ALGORITHM IS- **
14522 ** ERROR < 7*10E-6 *2**N **
14523 ** IF R IS CALCULATED CAREFULLY **
14524 **
14525 *****
    
```

Address	OpCode	OpCodeHex	OpCodeHex2	OpCodeHex3	OpCodeHex4	Instruction	Label
14526	73B2	DBA0				FPNEXP WORD MPLV.L3 LEVEL 3 ROUTINE	11007
14527	73B4	73B6				WORD **2	11008
14528	73B5	C25D	0012			MOV 18(R13),SOFT GET SOFTSTK POINTER	11009
14529	73BA	C0F9				MOV *SOFT+,R3	11010
14530	73BC	C119				MOV *SOFT,R4 SAVE X FOR FUTURE USE	11011
14531	73BE	0649				DECT SOFT	11012
14532	73C0	C050	3424			MOV LN2MANT,R1 CALCULATE N = X/LN(2)	11013
14533	73C4	C0A0	3338			MOV LN2EXP,R2	11014
14534	73C8	0420	7050			BLWP FPDIV	11015
14535	73CC	C079				MOV *SOFT+,R1	11016
14536	73CE	C0B9				MOV *SOFT+,R2 ROUND N TO NEAREST INTEGER BY FIRST	11017
14537	73D0	06A0	75A0			BL FP2INT CONVERTING TO INTEGER (ROUNDED)	11018
14538	73D4	1902				JNO **6	11019
14539	73D6	0460	74A6			B EXPOV IF N >32,767 EXP HAS OVERFLOWED	11020
14540	73DA	C201				MOV R1,R8 SAVE N FOR LATER	11021
14541	73DC	0A11				SLA R1,1 N (1BP)	11022
14542	73DE	1902				JNO **6 TEST FOR LARGE N	11023
14543	73E0	0460	74A6			B EXPOV	11024
14544	73E4	0741				ABS R1 @*N@* (1BP) TO R1	11025
14545	73E6	C181				MOV R1,R6 AND R5	11026
14546	73E8	3860	3426			MPY LNC1,R1 LNC1*@*N@* TO R1,R2	11027
14547	73EC	39A0	3428			MPY LNC2,R6 LNC2*@*N@* TO R6,R7	11028
14548						** LNC1=\$B172, LNC2=\$17F8 **	
14549	73F0	A086				A R6,R2 C=LN2=LNC1LNC2 (32BP)	11029
14550	73F2	1701				JNC **4 @*N@**C TO R1,R2	11030
14551	73F4	0581				INC R1 WITH (17BP)	11031
14552	73F6	C183				MOV R3,R6 MANTISSA: M	11032
14553	73F8	0746				ABS R6 @*N@* TO R6,R7	11033
14554	73FA	C1C6				MOV R6,R7	11034

14555	73FC	0200	000E	LI	R0,14		11035
14556	7400	6004		S	R4,R6	14-E>= 0 ?	11036
14557	7402	1151		JLT	EXPOV	OVERFLOW IF YES	11037
14558	7404	1304		JEQ	E1	NO SHIFT ON M	11038
14559	7406	0280	0010	CI	R0,15		11039
14560	740A	1545		JGT	ESMALL	E<=-3	11040
14561	740C	0906		SRL	R6,R0	SHIFT RIGHT 14-E	11041
14562	740E	0500		NEG	R0	-(14-E)	11042
14563	7410	0220	0010	AI	R0,15	16-(14-E)=E+2	11043
14564	7414	1301		JEQ	*+4	NO SHIFT IF =	11044
14565	7416	0A07		SLA	R7,R0	SHIFT LEFT E+2	11045
14566	7418	0546		INV	R6	NEGATE	11046
14567	741A	0507		NEG	R7		11047
14568	741C	1701		JNC	*+4	R6,R7 TO FORM	11048
14569	741E	0506		INC	R6	-X (17B ^D)	11049
14570	7420	A046		A	R6,R1		11050
14571	7422	A007		A	R7,R2	N*C-X	11051
14572	7424	1701		JNC	*+4		11052
14573	7426	0501		INC	R1	N*C-X TO R1,R2	11053
14574	7428	C0C3		MOV	R3,R3	R IN R1,R2	11054
14575	742A	1104		JLT	EVS		11055
14576	742C	0541		INV	R1	NOTE R1 ALL SIGN BITS	11056
14577	742E	0502		NEG	R2	OVERFLOW POSSIBLE	11057
14578	7430	1701		JNC	*+4		11058
14579	7432	0501		INC	R1		11059
14580	7434	C008		MOV	R8,R0	RECOVER N	11060
14581	7436	0649		DECT	SOFT	PREPARE TO	11061
14582	7438	C660	3354	MOV	C14,*SOFT	NORMALIZE	11062
14583	743C	0649		DECT	SOFT		11063
14584	743E	C642		MOV	R2,*SOFT		11064
14585	7440	0649		DECT	SOFT		11065
14586	7442	C641		MOV	R1,*SOFT		11066
14587	7444	06A0	713C	BL	NRMLIZ		11067
14588	7448	C079		MOV	*SOFT+,R1		11068
14589	744A	C099		MOV	*SOFT,R2		11069
14590	744C	C660	333E	MOV	FP6E,*SOFT		11070
14591	7450	0649		DECT	SOFT		11071
14592	7452	C660	33FC	MOV	FP6M,*SOFT		11072
14593	7456	0420	6F84	BLWP	FPADD	COMPUTE B+R	11073
14594	745A	0420	70AA	BLWP	FPMPY	COMPUTE R*(B+R)	11074
14595	745E	0649		DECT	SOFT		11075
14596	7460	C660	333E	MOV	FP6E,*SOFT		11076
14597	7464	0649		DECT	SOFT		11077
14598	7466	C660	33FC	MOV	FP6M,*SOFT		11078
14599	746A	0519		NEG	*SOFT	NEGATE 3	11079
14600	746C	0420	6F84	BLWP	FPADD	COMPUTE -B+R	11080
14601	7470	0420	70AA	BLWP	FPMPY	COMPUTE -R*(B-R)	11081
14602	7474	C060	33F6	MOV	FP12M,R1		11082
14603	7478	C0A0	3340	MOV	FP12E,R2		11083
14604	747C	0420	6F84	BLWP	FPADD	COMPUTE A+R**2-B*R	11084
14605	7480	C0F9		MOV	*SOFT+,R3	SAVE A+R**2-B*R FOR	11085
14606	7482	C139		MOV	*SOFT+,R4	FUTURE USE	11086
14607	7484	0420	6F84	BLWP	FPADD	COMPUTE A+R**2+B*R	11087
14608	7488	C043		MOV	R3,R1	CALCULATE E**X	11088
14609	748A	C084		MOV	R4,R2		11089
14610	748C	0420	7050	BLWP	FPDIV		11090
14611	7490	05C9		INCT	SOFT		11091
14612	7492	A640		A	R0,*SOFT	MULTIPLY TIMES 2**N	11092
14613	7494	0380		RTWP			11093

14614	7496	0220	FFF0	ESMALL	AI	R0,-16		11094
14615	749A	04C6			CLR	R6		11095
14616	749C	0280	0010		CI	R0,15		11096
14617	74A0	15C9			JGT	EVS	IF JGT, ZERO WILL	11097
14618	74A2	0907			SRL	R7,R0	BE IN R1,R2	11098
14619	74A4	10B9			JMP	E2	WHICH IS X-N*LN2	11099
14620	74A6	04C1		EXPJV	CLR	R1	LOAD 0 IF NEGATIVE, MAX IF POSITIVE	11100
14621	74A8	C0C3			MOV	R3,R3		11101
14622	74AA	1102			JLT	*+6		11102
14623	74AC	0201	7FFF		LI	R1,\$7FFF		11103
14624	74B0	0649			DECT	SOFT	PUSH MAXIMUM VALUE ONTO SOFTSTACK	11104
14625	74B2	C641			MOV	R1,*SOFT		11105
14626	74B4	0649			DECT	SOFT		11106
14627	74B6	C641			MOV	R1,*SOFT		11107
14628	74B8	0380			RTWP			11108

```

14630 *****
14631 **
14632 ** FLOATING POINT SQUARE ROOT (B=SQRT(B)) **
14633 **
14634 ** LEVEL 3 ROUTINE **
14635 **
14636 ** INPUT- B ON SOFTSTACK **
14637 ** OUTPUT- SQRT(B) ON SOFTSTACK **
14638 ** DESTROYS- R1-R5 **
14639 **
14640 ** STACK OPERATIONS- **
14641 ** SOFTSTACK - POPS 2 PUSHES 2 **
14642 **
14643 ** NOTE --- **
14644 ** ALGORITHM FOR SQRT(X) **
14645 **
14646 ** 1) EXPRESS X = W * 2 ** (2 * N) * .25 <= W < 1 **
14647 ** 2) THEN SQRT(X) = SQRT(W) * 2**N **
14648 ** 3) SQRT(W) = (Y+W/Y)/2 + ERRJR **
14649 ** 4) Y = A-B/(1+C) **
14650 ** 5) A=2.224609375 (2.163)B8 **
14651 ** 3=3.187500000 (3.140)B8 **
14652 ** C=1.599609375 (1.463)B8 **
14653 ** 6) 0 <= ERROR < 1/3 * 10**(-5) **
14654 **
14655 *****
    
```

Address	Hex	Hex	Hex	Label	Comment	Address
14656	74BA	DBA0		FPSQRT	WORD WPLVL3	LEVEL 3 ROUTINE 11110
14657	74BC	74BE			WORD *+2	11111
14658	74BE	C26D	0012	MOV	18(R13),SOFT	GET SOFTSTACK POINTER FROM CALLER 11112
14659	74C2	06A0	7134	BL	NRMCHK	NORMALIZE INPUT ARGUMENT 11113
14660	74C6	04CC		CLR	R12	SET STATUS BITS TO 0 11114
14661	74C8	C079		MOV	*SOFT+,R1	GET MANTISSA OF ARGUMENT 11115
14662	74CA	1327		JEQ	SQRT0	IF ZERO THEN RETURN ZERO 11116
14663	74CC	1503		JGT	*+8	11117
14664	74CE	E328	3406	SOC	OV,R12	SET OVERFLOW STATUS FOR NEGATIVE ARGUMENT 11118
14665	74D2	1021		JMP	SQRT0-4	11119
14666	74D4	C099		MOV	*SOFT,R2	GET EXPONENT FROM CALLER 11120
14667	74D6	0812		SRA	R2,1	DIVIDE EXPONENT BY 2 TO GET N 11121
14668	74D8	1702		JNC	*+6	IF EXPONENT WAS ODD 11122
14669	74DA	0811		SRA	R1,1	DIVIDE MANTISSA BY 2 & 11123
14670	74DC	0582		INC	R2	INCREMENT EXPONENT 11124
14671	74DE	3642		MOV	R2,*SOFT	PUT NEW EXPONENT BACK ON STACK 11125
14672	74E0	04C2		CLR	R2	CLEAR SECOND WORD OF MANTISSA 11126
14673	74E2	0811		SRA	R1,1	ADJUST W'S BINARY POINT TO MATCH C'S 11127
14674	74E4	1701		JNC	*+4	SAVE BIT SHIFTED OFF IN R2 11128
14675	74E6	0582		INC	R2	11129
14676	74E8	0812		SRC	R2,1	SHIFT IN BIT FROM R1 11130
14677	74EA	C0C1		MOV	R1,R3	11131
14678	74EC	A0E0	341E	A	SQRTC,R3	CALCULATE W+C 11132
14679	74F0	C120	341C	MOV	SQRTB,R4	11133
14680	74F4	04C5		CLR	R5	11134
14681	74F6	3D03		DIV	R3,R4	CALCULATE B/(W+C) 11135
14682	74F8	0A15		SLA	R5,1	11136
14683	74FA	8143		C	R3,R5	ROUND DIVISION 11137
14684	74FC	1B01		JH	*+4	11138
14685	74FE	0584		INC	R4	11139
14686	7500	0504		NEG	R4	GET -B/(W+C) 11140
14687	7502	A120	341A	A	SQRTA,R4	CALCULATE A-B/(W+C) 11141
14688	7506	0A14		SLA	R4,1	READJUST BINARY POINT 11142

14689	7508	3C44			DIV R4,R1	CALCULATE W/Y (Y=A-B/(W+C))	11143
14690	750A	0A12			SLA R2,1		11144
14691	750C	8084			C R4,R2	ROUND AFTER DIVISION	11145
14692	750E	1B01			JH *+4		11146
14693	7510	0581			INC R1		11147
14694	7512	A044			A R4,R1	CALCULATE Y+W/Y	11148
14695	7514	0911			SRL R1,1	CALCULATE (Y+W/Y)/2	11149
14696	7516	0649			DECT SOFT		11150
14697	7518	C641			MOV R1,*SOFT	RETURN MANTISSA TO SOFTSTACK	11151
14698	751A	02CF		SQRT0	STST R15	SET STATUS BITS A>.=	11152
14699	751C	024F	E7FF		ANDI R15,\$E7FF	SET STATUS BITS CARRY & OVERFLOW TO 0	11153
14700	7520	E3CC			SJC R12,R15	ADD STATUS BIT 0	11154
14701	7522	0380			RTWP	RETURN TO CALLER	11155

ARITHMETIC COMPARISON

11156

```

14703 *****
14704 **
14705 ** FLOATING POINT ARITHMETIC (A>,=) COMPARISON (A-B) **
14706 **
14707 ** LEVEL 4 ROUTINE **
14708 **
14709 ** INPUT- A IN R1,R2 **
14710 ** B ON SOFTSTACK **
14711 ** OUTPUT- NONE **
14712 **
14713 ** STACK OPERATIONS- **
14714 ** NONE **
14715 **
14716 ** STATUS BITS AFFECTED- A>, = **
14717 **
14718 *****
14719 7524 DBC0 FPCMPR WORD WPLVL4 LEVEL 4 ROUTINE 11157
14720 7526 7528 WORD *+2 11158
14721 7528 C26D 0012 MOV 16(R13),SOFT GET SOFTSTACK POINTER 11159
14722 752C 0649 DECT SOFT PUT A ON SOFTSTACK 11160
14723 752E C66D 0004 MOV 4(R13),*SOFT 11161
14724 7532 0649 DECT SOFT 11162
14725 7534 C66D 0002 MOV 2(R13),*SOFT 11163
14726 7538 06A0 7134 BL NRMCHK NORMALIZE ARGUMENT A 11164
14727 753C C079 MOV *SOFT+,R1 POP A OFF SOFTSTACK 11165
14728 753E C099 MOV *SOFT+,R2 11166
14729 7540 06A0 7134 BL NRMCHK NORMALIZE ARGUMENT B 11167
14730 7544 C0F9 MOV *SOFT+,R3 POP B OFF SOFTSTACK 11168
14731 7546 C139 MOV *SOFT+,R4 11169
14732 7548 024F 9FFF ANDI R15,$9FFF SET A> & = STATUS BITS TO 0 11170
14733 754C C041 MOV R1,R1 CHECK SIGN OF ARGUMENT A 11171
14734 754E 1307 JEQ EXPEQL IF A=0 THEN JUST COMPARE 3 TO 0 11172
14735 7550 110A JLT AMNTNEG IF A<0 CHECK IF B IS POSITIVE OR 0 11173
14736 7552 C0C3 MOV R3,R3 CHECK SIGN OF ARGUMENT B 11174
14737 7554 130E JEQ AGTB JUMP IF B=0 & A>0 11175
14738 7556 110D JLT AGTB JUMP IF B<0 & A>0 11176
14739 7558 8102 C R2,R4 IF BOTH A & B ARE POSITIVE THEN COMPARE 11177
14740 755A 150B JGT AGTB ACTUAL VALUES OF A & B 11178
14741 755C 110F JLT ALTB 11179
14742 755E 80C1 EXPEQL C R1,R3 IF EXPONENT'S ARE =, CHECK MANTISSAS 11180
14743 7560 1508 JGT AGTB 11181
14744 7562 110C JLT ALTB 11182
14745 7564 1009 JMP AEQB 11183
14746 7566 C0C3 AMNTNEG MOV R3,R3 CHECK SIGN OF B 11184
14747 7568 1309 JEQ ALTB JUMP IF A<0 & B=0 11185
14748 756A 1508 JGT ALTB JUMP IF A<0 & B>0 11186
14749 756C 8102 C R2,R4 IF SIGN ARE EQUAL CHECK VALUES 11187
14750 756E 13F7 JEQ EXPEQL IF EXPONENTS EQUAL CHECK MANTISSAS 11188
14751 7570 1505 JGT ALTB 11189
14752 7572 022F 4000 AGTB AI R15,$4000 SET A> STATUS BIT 11190
14753 7576 0380 RTWP 11191
14754 7578 022F 2000 AEQB AI R15,$2000 SET = STATUS BIT 11192
14755 757C 0380 ALTB RTWP RETURN TO CALLER 11193

```


INTEGER TO FLOATING POINT

11194

```

14757 *****
14758 *** **
14759 *** CONVERT INTEGER TO FLOATING POINT **
14760 *** **
14761 *** LEVEL 5 ROUTINE **
14762 *** **
14763 *** INPUT- INT2FP - INTEGER IN R1 **
14764 *** DINTFP - INTEGER IN R1,R2 **
14765 *** OUTPUT- FLOATING POINT NUMBER IN R1,R2 **
14766 *** DESTROYS- NONE **
14767 *** **
14768 *** STACK OPERATIONS- **
14769 *** SOFTSTK- USED BUT NO AFFECT TO CALLER **
14770 *** **
14771 *** NOTE --- **
14772 *** INT2FP CONVERTS A 1 WORD INTEGER TO **
14773 *** FLOATING POINT **
14774 *** DINTFP CONVERTS A DOUBLE WORD INTEGER **
14775 *** TO FLOATING POINT **
14776 *** **
14777 *****
14778 757E C081 INT2FP MOV R1,R2 CREATE A DOUBLE-WORD MANTISSA IN R1,R2 11195
14779 7580 08F1 SRA R1,15 R1 GET SIGN EXTENSION OF R2 11196
14780 7582 0649 DINTFP DECT SOFT SAVE RETURN ADDRESS 11197
14781 7584 C649 MOV R11,*SOFT 11198
14782 7586 0649 DECT SOFT 11199
14783 7588 C660 3360 MOV C31,*SOFT CREATE A FLOATING POINT NUMBER ON 11200
14784 758C 0649 DECT SOFT SOFTSTK FOR 'NRMLIZ' 11201
14785 758E C642 MOV R2,*SOFT PUT DOUBLE-WORD MANTISSA ON STACK 11202
14786 7590 0649 DECT SOFT 11203
14787 7592 C641 MOV R1,*SOFT 11204
14788 7594 06A0 713C BL NRMLIZ NORMALIZE FLOATING POINT NUMBER 11205
14789 7598 C079 MOV *SOFT+,R1 11206
14790 759A C0B9 MOV *SOFT+,R2 POP OFF FLOATING POINT NUMBER 11207
14791 759C C2F9 MOV *SOFT+,R11 POP RETURN ADDRESS 11208
14792 759E 045B B *R11 RETURN TO CALLER 11209

```

14794				*****			
14795				**			**
14796				**	CONVERT FLOATING POINT TO INTEGER (ROUND/TRUNCATE)		**
14797				**			**
14798				**	LEVEL 5 ROUTINE		**
14799				**			**
14800				**	INPUT- FLOATING POINT NUMBER IN R1,R2		**
14801				**	OUTPJT- INTEGER IN R1		**
14802				**	DESTROYS- R7,R8		**
14803				**			**
14804				**	STACK OPERATIONS-		**
14805				**	NONE		**
14806				**			**
14807				**	NOTE ---		**
14808				**	FP2INT ROUNDS OFF RESULT		**
14809				**	FPTRNC TRUNCATES RESULT		**
14810				**	STATUS BITS AFFECTED- A>, L>, =, 0		**
14811				**			**
14812				*****			
14813	75A0	0207	0001	FP2INT	LI	R7,1	ROUND RESULT 11211
14814	75A4	1001			JMP	*+4	11212
14815	75A6	04C7		FPTRNC	CLR	R7	TRUNCATE RESULT 11213
14816	75A8	C200			MOV	R0,R8	SAVE R0 11214
14817	75AA	C041			MOV	R1,R1	CHECK IF MANTISSA IS ZERO 11215
14818	75AC	130A			JEQ	INTSTST	11216
14819	75AE	C002			MOV	R2,R0	IF EXPONENT IS NEGATIVE HANDLE SEPARATELY 11217
14820	75B0	1112			JLT	EXPNGTV	11218
14821	75B2	0500			NEG	R0	SHIFT COUNT = 15 - EXPONENT 11219
14822	75B4	0220	000F		AI	R0,15	11220
14823	75B6	1109			JLT	OVFLW	IF <0 THEN NUMBER IS TOO LARGE FOR 16 BITS 11221
14824	75BA	1303			JEQ	INTSTST	IF =0 THEN DON'T SHIFT 11222
14825	75BC	0801			SRA	R1,R0	SHIFT TO CONVERT TO INTEGER 11223
14826	75BE	1761			JNC	INTSTST	11224
14827	75C0	A047			A	R7,R1	ROUND/TRUNCATE RESULT 11225
14828	75C2	C008		INTSTST	MOV	R8,R0	RESTORE R0 11226
14829	75C4	C201			MOV	R1,R8	11227
14830	75C6	04C1			CLR	R1	SET STATUS OF RESULT WITH OVERFLOW 11228
14831	75C8	A048			A	R8,R1	STATUS BIT RESET 11229
14832	75CA	045B			B	*R11	11230
14833	75CC	C008		OVFLW	MOV	R8,R0	RESTORE R0 11231
14834	75CE	0201	4000		LI	R1,84000	11232
14835	75D2	0A11			SLA	R1,1	SET OVERFLOW STATUS BIT 11233
14836	75D4	045B			B	*R11	11234
14837	75D6	08F1		EXPNGTV	SRA	R1,15	SHIFT TO -1 OR 0 11235
14838	75D8	C1C7			MOV	R7,R7	IF TRUNCATE THEN THIS IS CORRECT 11236
14839	75DA	13F3			JEQ	INTSTST	11237
14840	75DC	04C1			CLR	R1	IF ROUND THEN RESULT = 0 11238
14841	75DE	10F1			JMP	INTSTST	11239

```

14843 *****
14844 **
14845 ** FLOATING POINT CONVERSION FROM BASE 2 TO BASE 10 **
14846 ** **
14847 ** LEVEL 3 ROUTINE **
14848 ** **
14849 ** INPUT- 3 WORDS ON SOFTSTACK --- **
14850 ** 1) 1 WORD CONVERSION FORMAT **
14851 ** 2) 2 WORD BASE 2 FP NUMBER **
14852 ** OUTPUT- N WORDS ON SOFTSTACK --- **
14853 ** 1) 1 WORD STATUS OF CONVERSION **
14854 ** 2) N-1 WORDS OF ASCII CHARACTERS **
14855 ** DESTROYS- R0-R12 **
14856 ** **
14857 ** NOTE --- **
14858 ** FORMAT DESCRIPTION WORD- **
14859 ** **
14860 ** FRZ EJDJ PPPP SSSS # OF BITS **
14861 ** 111 1111 1 1 **
14862 ** 111 1111 1 --- OUTPUT BUFFER SIZE (4) **
14863 ** 111 1111 ----- PRECISION DIGITS (4) **
14864 ** 111 111 ----- UNUSED (1) **
14865 ** 111 11 ----- OVERFLOW TYPE (1) **
14866 ** 111 1 ----- JUSTIFICATION (1) **
14867 ** 111 ----- EXPONENT TYPE (1) **
14868 ** 11 ----- TRAILING ZEROS (1) **
14869 ** 1 ----- TRUNCATE/ROUND (1) **
14870 ** ----- FORMAT TYPE (2) **
14871 ** **
14872 ** FORMAT TYPE - 00 - INTEGER **
14873 ** 01 - FIXED POINT **
14874 ** 10 - SCIENTIFIC **
14875 ** 11 - ENGINEERING **
14876 ** TRUNCATE/ROUND - 0/1 - TRUNCATE/ROUND **
14877 ** TRAILING ZEROS - 0/1 - NO/YES **
14878 ** EXPONENT TYPE -- 0/1 - DIGITS/LETTER EQUIVALENT **
14879 ** JUSTIFICATION -- 0/1 - RIGHT/LEFT **
14880 ** OVERFLOW TYPE -- 0/1 - *****/1E+100 **
14881 ** **
14882 ** CONVERSION STATUS WORD- **
14883 ** THIS WORD INDICATES WHETHER OR NOT THE BASE **
14884 ** 10 NUMBER FIT IN THE SPECIFIED BUFFER SPACE. **
14885 ** IF IT DID, THIS WORD'S VALUE WILL EQUAL THE **
14886 ** BUFFER SPACE REQUESTED OTHERWISE IT WILL BE **
14887 ** THE NEGATIVE OF BUFFER SPACE REQUIRED. **
14888 ** **
14889 *****
14890 *
14891 * FOLLOWING SECTION CONVERTS A NUMBER FROM FLOATING POINT
14892 * BASE 2 TO FLOATING POINT BASE TEN USING A TABLE LOOKUP. TWO
14893 * TABLES ARE USED- ONE WITH EACH VALUE A MULTIPLE OF 1*10**10
14894 * AND OTHER WITH A MULTIPLE OF 10. THIS CONVERSION RESULTS
14895 * IN A MANTISSA BETWEEN 1.000 AND 9.999 IF IT IS NOT ZERO.
14896 *
14897 * REGISTER USAGE IN THIS SECTION ---
14898 * INPUT-
14899 * SOFT - SOFTSTACK POINTER
14900 * OUTPUT-
14901 * R0 - CONVERSION FORMAT
  
```

14902				*	R1 - INTEGER PORTION OF MANTISSA (1,9)	
14903				*	R2 - FRACTION PORTION OF MANTISSA (.000,.999)	
14904				*	R3 - EXPONENT	
14905				*	R12- MANTISSA'S SIGN FLAG	
14906				*		
14907	75E0		DBA0	FP2AS3	WORD WPLVL3	LEVEL 3 ROUTINE 11241
14908	75E2		75E4		WCRJ *+2	11242
14909	75E4	C26D	0012		MOV 10(R13),SOFT	GET SOFTSTACK POINTER 11243
14910	75E8	C039			MOV *SOFT+,R0	POP CONVERSION FORMAT OFF STACK 11244
14911	75EA	06A0	7134		BL NRMCHK	NORMALIZE INPUT NUMBER 11245
14912	75EE	04CC			CLR R12	SET FLAG INDICATING POSITIVE VALUE 11246
14913	75F0	0759			ABS *SOFT	GET ABSOLUTE VALUE OF MANTISSA 11247
14914	75F2	1906			JNO NT8000	CHECK IF MANTISSA IS \$0000 11248
14915	75F4	C079			MOV *SOFT+,R1	IF SO, SHIFT TO \$4000 AND 11249
14916	75F6	0911			SRL R1,1	INCREMENT EXPONENT TO MATCH SHIFT 11250
14917	75F8	0599			INC *SOFT	11251
14918	75FA	0649			DECT SOFT	11252
14919	75FC	C641			MOV R1,*SOFT	11253
14920	75FE	1002			JMP *+6	11254
14921	7600	134E		NT8000	JEQ CNVZERO	IF VALUE IS ZERO DON'T BOTHER CONVERTING 11255
14922	7602	1501			JGT *+4	IF ORIGINAL VALUE WAS NEGATIVE SET 11256
14923	7604	05CC			INCT R12	FLAG INDICATING A NEGATIVE VALUE 11257
14924	7606	0203	0034		LI R3,TENSIZ	LOAD TENS TABLE INDEX 11258
14925	760A	C079			MOV *SOFT+,R1	POP NUMBER OFF STACK INTO R1,R2 11259
14926	760C	C099			MOV *SOFT,R2	11260
14927	760E	0649			DECT SOFT	11261
14928	7610	05C3			INCT R3	11262
14929	7612	0223	FFFC	RANGE10	AI R3,-4	POINT TO NEXT EXPONENT IN TABLE 11263
14930	7616	1143			JLT CNVZERO	IF VALUE NOT FOUND IN TABLE THEN UNDERFLOW 11264
14931	7618	80A3	354A		C TENTBL(R3),R2	COMPARE EXPONENTS 11265
14932	761C	15FA			JGT RANGE10	IF TABLE ENTRY IS LARGER CONTINUE LOOKING 11266
14933	761E	1105			JLT FND10	IF LESS THAN THIS IS ONE WE WANT 11267
14934	7620	0643			DECT R3	11268
14935	7622	8053	354A		C TENTBL(R3),R1	IF EQUAL CHECK MANTISSAS 11269
14936	7626	15F4			JGT RANGE10-2	11270
14937	7628	05C3			INCT R3	11271
14938	762A	0203	0052	FND10	CI R3,TENSIZ-2	CHECK FOR OVERFLOW 11272
14939	762E	133E			JEQ CNVOVER	11273
14940	7630	C0A3	354A		MOV TENTBL(R3),R2	PUT TABLE ENTRY IN R1,R2 11274
14941	7634	0643			DECT R3	11275
14942	7636	C063	354A		MOV TENTBL(R3),R1	11276
14943	763A	0420	7050		BLWP FPDIV	DIVIDE BY THIS RANGE TO GET NUMBER WITHIN 11277
14944	763E	0923			SRL R3,2	CALCULATE TENS' EXPONENT RANGE 11278
14945	7640	38E0	334C		MPY C10,R3	RANGE = (INDEX / 4)*10 - 100 11279
14946	7644	0224	FF9C		AI R4,-100	11280
14947	7648	0203	0028		LI R3,ONESIZ	1.000*10**0 AND 9.999*10**9 11281
14948	764C	C079			MOV *SOFT+,R1	POP NUMBER OFF STACK INTO R1,R2 11282
14949	764E	C099			MOV *SOFT,R2	11283
14950	7650	0649			DECT SOFT	11284
14951	7652	05C3			INCT R3	11285
14952	7654	0223	FFFC	RANGE1	AI R3,-4	POINT TO NEXT EXPONENT IN TABLE 11286
14953	7658	1118			JLT RNGFND	11287
14954	765A	80A3	3522		C ONETBL(R3),R2	COMPARE EXPONENTS 11288
14955	765E	15FA			JGT RANGE1	IF TABLE ENTRY IS LARGER CONTINUE LOOKING 11289
14956	7660	1105			JLT FND1	IF LESS THAN THIS IS ONE WE WANT 11290
14957	7662	0643			DECT R3	11291
14958	7664	8063	3522		C ONETBL(R3),R1	IF EQUAL CHECK MANTISSAS 11292
14959	7668	15F4			JGT RANGE1-2	11293
14960	766A	05C3			INCT R3	11294

14961	766C	C0A3	3522	FND1	MOV	ONET3L(R3),R2	PUT TABLE ENTRY IN R1,R2	11295
14962	7670	0643			DECT	R3		11296
14963	7672	C063	3522		MOV	ONET3L(R3),R1		11297
14964	7676	0420	7050		BLMP	FPDIV	DIVIDE THIS RANGE INTO VALUE FROM ABOVE	11298
14965	767A	0923			SRL	R3,2	CALCULATE TENS' EXPONENT RANGE	11299
14966	767C	A103			A	R3,R4	R3 <-- BASE 10 EXPONENT (-99,99)	11300
14967	767E	0284	0063		CI	R4,99	CHECK FOR OVERFLOW	11301
14968	7682	1514			JGT	CNVOVER		11302
14969	7684	0284	FF9D		CI	R4,-99		11303
14970	7688	110A			JLT	CNVZERO		11304
14971	768A	C0C4		RNGFN3	MOV	R4,R3		11305
14972	768C	C0B9			MOV	*SOFT+,R2	SOFTSTK <-- BASE 10 MANTISSA (1.010,9.999)	11306
14973	768E	04C1			CLR	R1		11307
14974	7690	0A11		MNTSHFT	SLA	R1,1	PUT INTEGER PORTION OF MANTISSA IN R1	11308
14975	7692	0A12			SLA	R2,1	& FRACTION PORTION IN R2	11309
14976	7694	1701			JNC	*+4		11310
14977	7696	0581			INC	R1	SHIFT ALL ONES OUT OF R2 INTO R1	11311
14978	7698	0619			DEC	*SOFT		11312
14979	769A	1135			JLT	FORMAT	SHIFT N+1 TIMES TO ACCOUNT FOR BINARY POINT	11313
14980	769C	10F9			JMP	MNTSHFT	POSITION (N=BASE 2 EXPONENT)	11314
14981				*				
14982				*		IF UNDERFLOW OR ZERO OUTPUT A VALUE OF 0		
14983				*				
14984	769E	04C1			CNVZERO	CLR R1	INTEGER PORTION OF MANTISSA = 0	11315
14985	76A0	04C2			CLR	R2	FRACTION PORTION OF MANTISSA = 0	11316
14986	76A2	04C3			CLR	R3	EXPONENT = 0	11317
14987	76A4	0240	DFFF		ANDI	R0,\$DFFF	FLAG TO TRUNCATE THESE VALUES	11318
14988	76A8	05C9			INCT	SOFT	POSITION SOFTSTK TO EXPONENT	11319
14989	76AA	102E			JMP	FORMAT		11320
14990				*				
14991				*		IF OVERFLOW OUTPUT A STRING OF ASTERISKS		
14992				*				
14993	76AC	C25D	0012		CNVJVER	MOV 18(R13),SOFT	POP FLOATING POINT NUMBER OFF STACK	11321
14994	76B0	0229	0004		AI	SOFT,4		11322
14995	76B4	2020	337A		COC	CH200,R0	WHICH OVERFLOW TYPE IS SPECIFIED?	11323
14996	76B8	1318			JEQ	E.100	1E+100	11324
14997	76BA	0240	000F		ANDI	R0,\$000F	KEEP BUFFER SIZE ONLY	11325
14998	76BE	1602			JNE	*+6		11326
14999	76C0	0200	000A		LI	R0,10	DEFAULT TO 10 IF 0	11327
15000	76C4	C040			MOV	R0,R1		11328
15001	76C6	C660	34E0		MOV	BLANK,*SOFT	START WITH ' '	11329
15002	76CA	0649			DECT	SOFT		11330
15003	76CC	0641			DECT	R1	IS THERE STILL SPACE?	11331
15004	76CE	1106			JLT	ASTRTRN		11332
15005	76D0	1305			JEQ	ASTRTRN		11333
15006	76D2	C660	34DE	ASTFILL	MOV	ASTERISK,*SOFT	FILL BUFFER WITH '*'S	11334
15007	76D6	0649			DECT	SOFT		11335
15008	76D8	0601			DEC	R1	DECREMENT COUNT OF BUFFER SIZE	11336
15009	76DA	15FB			JGT	ASTFILL		11337
15010	76DC	C66C	34E5	ASTRTRN	MOV	ESIGNS(R12),*SOFT	END WITH SIGN	11338
15011	76E0	0649			DECT	SOFT		11339
15012	76E2	C648			MOV	R0,*SOFT	PUT BUFFER SIZE ONTO STACK	11340
15013	76E4	C849	0012		MOV	SOFT,18(R13)	RETURN CURRENT SOFTSTACK POINTER	11341
15014	76E8	0380			RTWP		RETURN TO CALLE	11342
15015								
15016	76EA	0200	0007	E.100	LI	R0,7	SIX CHARACTERS IN +1E+100	11343
15017	76EE	C040			MOV	R0,R1		11344
15018	76F0	0641			DECT	R1		11345
15019	76F2	00A1	7702	E100FLL	MOV3	EOVER(R1),R2	GET NEXT OVERFLOW DATA BYTE	11346

15020	76F6	0982	SRL	R2,0	PUT BYTE IN LOW BYTE, ZERO HIGH BYTE	11347
15021	76F8	0642	MOV	R2,*SOFT	PUSH DATA ONTO SOFTSTACK	11348
15022	76FA	0649	DECT	SOFT		11349
15023	76FC	0601	DEC	R1	DECREMENT BYTE COUNT	11350
15024	76FE	11EE	JLT	ASTRTRN	WHEN NEGATIVE ADD MANTISSA SIGN	11351
15025	7700	10F8	JMP	E100FLL		11352
15026						
15027	7702	31	EOVER	FGC	'1E+100'	11353

7703	45
7704	2B
7705	31
7706	30
7707	30

15028

15029 *****

15030 *

15031 * INTEGER AND PRECISION DIGIT COUNTS.

15032 *

15033 * REGISTER USAGE IN SECTION ---

15034 * INPUT-

15035 * R0 - CONVERSION FORMAT

15036 * R1 - INTEGER DIGIT OF MANTISSA

15037 * R2 - FRACTION DIGITS OF MANTISSA

15038 * R3 - EXPONENT

15039 * OUTPUT-

15040 * R4 - INTEGER DIGIT COUNT

15041 * R5 - PRECISION DIGIT COUNT

15042 *

15043 * RULES ---

15044 * I = # OF INTEGER DIGITS

15045 * P = # OF PRECISION DIGITS (BITS 4-7 OF FORMAT WORD)

15046 * M = BASE 10 MANTISSA (0-1.000,9.999)

15047 * E = BASE 10 EXPONENT (-99,99)

15048 *

15049 * ALL FORMATS (XX)

15050 * IF (P=0) THEN P=16

15051 * INTEGER FORMAT (00)

15052 * I = E + 1

15053 * IF (I>P) THEN SCIENTIFIC

15054 * IF (I<1) THEN M=0

I=1

15055 * P = I

15056 * FIXED POINT FORMAT (01)

15057 * I = E + 1

15058 * IF (I>P) THEN SCIENTIFIC

15059 * IF (I<1) THEN M=M*(10**E)

I=1

15060 * SCIENTIFIC FORMAT (10)

15061 * I = 1

15062 * ENGINEERING FORMAT (11)

15063 * I = 1<=I<=3

15064 * EXPONENT IS A MULTIPLE OF 3

15065 *

15066 7708 C140 FORMAT MOV R0,R5 PUT PRECISION DIGIT COUNT INTO R5 11354

15067 770A 0245 00F0 ANDI R5,000F0 IT IS BITS 4-7 OF FORMAT WORD 11355

15068 770E 0945 SRL R5,4 11356

15069 7710 1602 JNE *+6 11357

15070 7712 0205 0010 LI R5,15 IF (P=0) THEN P=16 11358

15071 7716 C000 MOV R0,R0 DECODE FORMAT INTO (INTEGER/FIXED) 11359

15072 7718 110F JLT SCIENG & (SCIENTIFIC/ENGINEERING) 11360

15073 771A C103 INTFIX MOV R3,R4 11361

15074	771C	0584			INC R4	I = E + 1	11362
15075	771E	8144			C R4,R5	IF (I>3) THEN SCIENTIFIC	11363
15076	7720	150E			JGT SCI		11364
15077	7722	04C3			CLR R3	SET EXPONENT TO ZERO	11365
15078	7724	2020	3386		COC CH4000,R0	DECODE INTO (INTEGER) & (FIXED)	11366
15079	7728	131A			JEQ MANTJUT		11367
15080	772A	C184			MOV R4,R4	IF (I<1) THEN M=0	I=1 11368
15081	772C	1503			JGT *+8		11369
15082	772E	04C1			CLR R1	M = 0	11370
15083	7730	0204	0001		LI R4,1	I = 1	11371
15084	7734	C144			MOV R4,R5	P = I	11372
15085	7736	1013			JMP MANTJUT		11373
15086	7738	2020	3386	SCIENG	COC CH4000,R0	DECODE INTO (SCIENTIFIC) & (ENGINEERING)	11374
15087	773C	1303			JEQ *+8		11375
15088	773E	0204	0001	SCI	LI R4,1	I = 1	11376
15089	7742	100D			JMP MANTJUT		11377
15090	7744	04C5			CLR R6		11378
15091	7746	C1C3			MOV R3,R7		11379
15092	7748	0227	0063		AI R7,99	ADJUST EXPONENT RANGE TO (0,198)	11380
15093	774C	3DA0	333E		DIV C3,R6	I = REMAINDER + 1	11381
15094	7750	C107			MOV R7,R4		11382
15095	7752	0584			INC R4		11383
15096	7754	39A0	333E		MPY C3,R5	E = DIVIDEND * 3	11384
15097	7758	0227	FF9D		AI R7,-99	ADJUST RANGE BACK TO (-99,99)	11385
15098	775C	C0C7			MOV R7,R3		11386
15099					* FOLLOWING SECTION PUTS BASE 10 FLOATING		
15100					* POINT NUMBER ON SOFTSTACK IN ASCII.		
15101					* REGISTER USAGE IN THIS SECTION---		
15102					* INPUT-		
15103					* R0 - CONVERSION FORMAT		
15104					* R1 - INTEGER DIGIT OF MANTISSA		
15105					* R2 - FRACTION DIGITS OF MANTISSA		
15106					* R3 - EXPONENT		
15107					* R4 - INTEGER DIGIT COUNT		
15108					* R5 - PRECISION DIGIT COUNT		
15109					* R12- MANTISSA'S SIGN FLAG		
15110					* OUTPUT MANTISSA WHICH IS IN R1,R2		
15111					* MANTJUT MOV R5,R5		MOVE PRECISION DIGIT COUNT TO R6 11387
15112					* DEC R6		PUT A ZERO ON AS FIRST DIGIT IN CASE 11388
15113					* CLR *SOFT		ROUNDING NEEDS ANOTHER DIGIT 11389
15114	775E	C185			* MOV R4,R4		CHECK IF INTEGER DIGIT COUNT IS NEGATIVE 11390
15115	7760	0606			* JGT DIGIT		11391
15116	7762	04D9			* PREZERO DECT SOFT		IF SO, THEN ADD ABS(I)+1 ZEROS TO BEGINNING 11392
15117	7764	C104			* CLR *SOFT		11393
15118	7766	1507			* DEC R6		CHECK TO SEE IF REQUESTED PRECISION HAS BEEN 11394
15119	7768	0649			* JLT ROUND-4		OUTPUT 11395
15120	776A	04D9			* INC R4		CHECK TO SEE IF ALL ZEROS HAVE BEEN ADDED 11396
15121	776C	0606			* JGT DIGIT		11397
15122	776E	1111			* JMP PREZERO		11398
15123	7770	0584			* DIGIT CI R1,10		CHECK IF VALUE ALREADY ROUNDED TO 10 11399
15124	7772	1501			* JL *+8		11400
15125	7774	10F9			* INC *SOFT		IF SO, FIRST DIGIT IS A 1 11401
15126	7776	0281	000A	DIGIT	* AI R1,-10		11402
15127	777A	1A03			* DECT SOFT		11403
15128	777C	0599			* MOV R1,*SOFT		PUT NEXT DIGIT ON SOFTSTACK 11404
15129	777E	0221	FFF5		* DEC R6		DECREMENT PRECISION DIGIT COUNT TO SEE IF 11405
15130	7782	0649					
15131	7784	C641					
15132	7786	0606					

15133	7788	1105		JLT	ROUND	REQUESTED PRECISION HAS BEEN OUTPUT	11406	
15134	778A	C042		MOV	R2,R1		11407	
15135	778C	3860	334C	MPY	C10,R1	GET NEXT DIGIT TO OUTPUT IN R1 AND LEAVE	11408	
15136	7790	10F2		JMP	DIGIT	REMAINDER IN R2 FOR NEXT LOOP	11409	
15137	7792	2204	0001	LI	R4,1	SET INTEGER DIGIT COUNT TO 1	11410	
15138	7796	C209		ROUND	MOV	SAVE CURRENT SOFTSTACK POINTER	11411	
15139	7798	2420	3382	CZC	CH2000,R0	CHECK IF VALUE SHOULD BE ROUNDED	11412	
15140	779C	1308		JEQ	TRNCTE	IF NOT SKIP THIS PART	11413	
15141	779E	0A12		SLA	R2,1	CHECK IF REMAINDER IS .5 OR GREATER	11414	
15142	77A0	1706		JNC	TRNCTE	IF NOT DON'T ROUND UP	11415	
15143	77A2	0599		RNDLOOP	INC	INCREMENT LEAST SIGNIFICANT DIGIT	11416	
15144	77A4	8819	334C	C	*SOFT,C10	CHECK IF ROUNDING CAUSED DIGIT'S VALUE TO	11417	
15145	77A8	1A02		JL	TRNCTE	INCREASE TO 10	11418	
15146	77AA	04F9		CLR	*SOFT+	IF IT DID SET THIS DIGIT TO 0 AND ROUND UP	11419	
15147	77AC	10FA		JMP	RNDLOOP	NEXT DIGIT	11420	
15148	77AE	C1C9		TRNCTE	MOV	SAVE THIS POSITION OF SOFTSTACK	11421	
15149	77B0	C248		MOV	R8,SOFT	RESTORE POINTER TO LEAST SIGNIFICANT DIGIT	11422	
15150	77B2	0A15		SLA	R5,1		11423	
15151	77B4	A245		A	R5,SOFT	ADD DIGIT COUNT TO POINT TO MOST	11424	
15152	77B6	0915		SRL	R5,1	SIGNIFICANT DIGIT	11425	
15153	77B8	0584		INC	R4	INCREMENT INTEGER COUNT TO ALLOW FOR ADDED DI	11426	
15154	77BA	0585		INC	R5	DO SAME FOR PRECISION DIGIT COUNT	11427	
15155	773C	81C9		C	SOFT,R7	IF ROUNDING MADE EXTRA DIGIT A ONE THEN	11428	
15156	77BE	1610		JNE	INTOUT	FURTHER ADJUSTMENTS ON FORMAT ARE NEEDED	11429	
15157	77C0	C000		MOV	R0,R0	ADJUST SCIENTIFIC & ENGINEERING FORMATS	11430	
15158	77C2	150E		JGT	INTOUT		11431	
15159	77C4	0605		DEC	R5	DECREMENT PRECISION DIGIT COUNT	11432	
15160	77C6	2020	3386	CJC	CH4000,R0	ENGINEERING FORMAT?	11433	
15161	77CA	1303		JEQ	**8		11434	
15162	77CC	0583		INC	R3	FOR SCIENTIFIC FORMAT, INCREMENT EXPONENT &	11435	
15163	77CE	0604		DEC	R4	DECREMENT INTEGER DIGIT COUNT SO OUTPUT	11436	
15164	77D0	1007		JMP	INTOJT	MANTISSA WILL BE IN RANGE (1.000,9.999)	11437	
15165	77D2	2284	0004	CI	R4,4	FOR ENGINEERING FORMAT CHECK IF INTEGER COUNT	11438	
15166	77D6	1104		JLT	INTOUT	IS 4. IF SO INCREMENT EXPONENT BY 3 & SET	11439	
15167	77D8	0223	0003	AI	R3,3	INTEGER COUNT TO 1 SO MANTISSA RANGE IS	11440	
15168	77DC	0204	0001	LI	R4,1	(1.000,999.9) & EXPONENT IS A MULTIPLE OF 3	11441	
15169	77E0	5144		INTOUT	S	CALCULATE FRACTION DIGIT COUNT	11442	
15170	77E2	C209		MOV	SOFT,R8	SAVE POINTER TO MOST SIGNIFICANT DIGIT	11443	
15171	77E4	A660	336A	A	CH30,*SOFT	CONVERT INTEGER DIGIT TO ASCII	11444	
15172	77E8	0649		DECT	SOFT		11445	
15173	77EA	2604		DEC	R4	DECREMENT INTEGER COUNT TO SEE WHEN	11446	
15174	77EC	15FB		JGT	INTOJT+4	DECIMAL POINT SHOULD BE INCLUDED	11447	
15175	77EE	C059		MOV	*SOFT,R1	SAVE FIRST FRACTION DIGIT	11448	
15176	77F0	C660	34FA	MOV	PERIOD1,*SOFT	PUT A DECIMAL POINT IN ITS PLACE	11449	
15177	77F4	0605		FRCTOUT	DEC	DECREMENT FRACTION DIGIT COUNT	11450	
15178	77F6	1107		JLT	ZTRAIL	WHEN ALL FRACTION DIGITS ARE CONVERTED STOP	11451	
15179	77F8	0649		DECT	SOFT	POINT TO NEXT DIGIT	11452	
15180	77FA	C099		MOV	*SOFT,R2	SAVE NEXT FRACTION DIGIT	11453	
15181	77FC	0221	0030	AI	R1,\$30	CONVERT FRACTION DIGIT TO ASCII	11454	
15182	7800	C641		MOV	R1,*SOFT	PUT PREVIOUS FRACTION DIGIT IN ITS PLACE	11455	
15183	7802	C042		MOV	R2,R1		11456	
15184	7804	10F7		JMP	FRCTOUT		11457	
15185	7806	2020	3380	ZTRAIL	CJC	CH1000,R0	CHECK IF TRAILING ZEROS SHOULD BE ELIMINATED	11458
15186	780A	1304		JEQ	PERCHK		11459	
15187	780C	8839	336A	C	*SOFT+,CH30	IF DIGIT IS A ZERO DON'T INCLUDE IT	11460	
15188	7810	13FD		JEQ	*-4		11461	
15189	7812	0649		DECT	SOFT		11462	
15190	7814	8839	34FA	PERCHK	C	*SOFT+,PERIOD1	IF NO FRACTION DIGITS THEN NO DECIMAL POINT	11463
15191	7818	1301		JEQ	*+4		11464	

15192	781A	0649		DECT	SOFT		11465
15193				*			
15194				*	OUTPUT EXPONENT WHICH IS IN R3		
15195				*			
15196	781C	04C1		EXPOJT	CLR R1	SET A FLAG TO INDICATE A POSITIVE EXPONENT	11466
15197	781E	0743			ABS R3	GET ABSOLUTE VALUE OF EXPONENT	11467
15198	7820	1324			JEQ REVERSE	IF EXPONENT IS ZERO DON'T INCLUDE IT	11468
15199	7822	1501			JGT *+4		11469
15200	7824	05C1			INCT R1	SET FLAG TO INDICATE A NEGATIVE EXPONENT	11470
15201	7826	0649			DECT SOFT		11471
15202	7828	2420	337E		CZC CH00,R0	CHECK IF A LETTER EQUIVALENT EXPONENT	11472
15203	782C	1300			JEQ EXPNUM	SHOULD BE USED, IF NOT OUTPUT DIGITS	11473
15204	782E	0283	080C		CI R3,12	CHECK IF EXPONENT IS IN RANGE (-12,12)	11474
15205	7832	150A			JGT EXPNUM	IF NOT, A DIGIT EXPONENT MUST BE USED	11475
15206	7834	0923			SRL R3,2		11476
15207	7836	0A13			SLA R3,1	CALCULATE DISPLACEMENT INTO LETTER TABLES	11477
15208	7838	0841			MOV R1,R1	CHECK IF POSITIVE OR NEGATIVE EXPONENT	11478
15209	783A	1303			JEQ *+8		11479
15210	783C	0663	34F2		MOV NEGEXP(R3),*SOFT	USE LETTER FOR NEGATIVE EXPONENTS	11480
15211	7840	1014			JMP REVERSE		11481
15212	7842	0663	34EA		MOV POSEXP(R3),*SOFT	USE LETTER FOR POSITIVE EXPONENTS	11482
15213	7846	1011			JMP REVERSE		11483
15214	7848	0660	34FC	EXPNUM	MOV E,*SOFT	PUT LETTER 'E' ON STACK	11484
15215	784C	0649			DECT SOFT		11485
15216	784E	0661	34E6		MOV ESIGNS(R1),*SOFT	PUT EXPONENT SIGN ON SOFTSTACK	11486
15217	7852	0103			MOV R3,R4		11487
15218	7854	04C3			CLR R3		11488
15219	7856	3CE0	334C		DIV C10,R3	GET TENS DIGIT IN R3 AND ONES DIGIT IN R4	11489
15220	785A	0223	0030		AI R3,\$30	CONVERT TENS DIGIT TO ASCII	11490
15221	785E	0224	0030		AI R4,\$30	CONVERT ONES DIGIT TO ASCII	11491
15222	7862	0649			DECT SOFT		11492
15223	7864	0643			MOV R3,*SOFT	PUT TENS DIGIT ON SOFTSTACK	11493
15224	7866	0649			DECT SOFT		11494
15225	7868	0644			MOV R4,*SOFT	PUT ONES DIGIT ON SOFTSTACK	11495
15226				*			
15227				*	REVERSE ORDER OF ENTRIES ON SOFTSTACK SO WHEN POPPED		
15228				*	NUMBER WILL COME OUT IN CORRECT ORDER.		
15229				*			
15230	786A	01C9		REVERSE	MOV SOFT,R7	SAVE POSITION OF SOFTSTACK POINTER	11496
15231	786C	0188			MOV R8,R5	SAVE POINTER TO MOST SIGNIFICANT DIGIT	11497
15232	786E	0048			MOV R8,R1		11498
15233	7870	6047			S R7,R1	CALCULATE NUMBER OF WORDS TO REVERSE	11499
15234	7872	05C1			INCT R1		11500
15235	7874	0921			SRL R1,2	DIVIDE COUNT BY 4	11501
15236	7876	0117		SWAP	MOV *R7,R4	SWITCH TWO WORDS TO REVERSE STACK	11502
15237	7878	00D8			MOV *R8,*R7*		11503
15238	787A	0604			MOV R4,*R8		11504
15239	787C	0648			DECT R8		11505
15240	787E	0601			DEC R1	DECREMENT REVERSING COUNT	11506
15241	7880	15FA			JGT SWAP		11507
15242	7882	8819	336A		C *SOFT,3H30	CHECK IF MOST SIGNIFICANT DIGIT IS STILL	11508
15243	7886	1301			JEQ *+4	A ZERO, IF IT IS OVERWRITE IT WITH	11509
15244	7888	0649			DECT SOFT	MANTISSA'S SIGN	11510
15245	788A	066C	34E2		MOV MSIGNS(R12),*SOFT		11511
15246	788E	8819	34E8		C *SOFT,3LANK	CHECK IF MANTISSA SIGN IS A BLANK	11512
15247	7892	1601			JNE *+4	IF IT IS, POP IT OFF	11513
15248	7894	05C9			INCT SOFT		11514
15249	7896	5189			S SOFT,R5	CALCULATE NUMBER OF WORDS PUSHED ONTO SOFTSTK	11515
15250	7898	05C6			INCT R6		11516

ASCII OUTPUT CONVERSION

11240

15251	789A	0916		SRL	R6,1		11517
15252	789C	C140		MOV	R0,R5		11518
15253	789E	0245	000F	ANDI	R5,\$000F	GET DESIRED BUFFER SIZE	11519
15254	78A2	1304		JEQ	ADD SIZE	IF BUFFER SIZE IS ZERO OUTPUT SIZE USED	11520
15255	78A4	8195		CHKBJF	C	R5,R5	CHECK TO SEE IF BUFFER HAS BEEN FILLED
15256	78A6	1507		JGT	ADDSPCS	IF NOT ADD BLANKS TO FILL IT UP	11522
15257	78A8	1301		JEQ	*+4	IF YES PUT THIS COUNT ON STACK	11523
15258	78AA	0506		NEG	R6	IF OVERFILLED, NEGATE COUNT AND PUT ON STACK	11524
15259	78AC	0649		ADD SIZE	DECT	SOFT	11525
15260	78AE	C646		MOV	R6,*SOFT		11526
15261	78B0	C849	0012	MOV	SOFT,18(R13)	RETURN SOFTSTACK POINTER	11527
15262	78B4	0380		RTWP		RETURN TO CALLER	11528
15263	78B6	2028	337C	ADDSPCS	CDC	CH400,R0	CHECK IF NUMBER SHOULD BE LEFT JUSTIFIED
15264	78BA	1305		JEQ	LEFTJST		11529
15265	78BC	0649		DECT	SOFT	MOVE A BLANK INTO BUFFER	11531
15266	78BE	C660	34E0	MOV	BLANK,*SOFT		11532
15267	78C2	0586		INC	R6	INCREMENT FILLED COUNT	11533
15268	78C4	10EF		JMP	CHKBUF		11534
15269	78C6	C106		LEFTJST	MOV	R6,R4	GET COUNT OF # OF WORDS TO SHIFT IN R4
15270	78C8	C0D9		MOV	*SOFT,R3	MOVE NEXT DIGIT TO R3	11536
15271	78CA	0649		DECT	SOFT		11537
15272	78CC	CE43		MOV	R3,*SOFT+	MOVE THIS DIGIT TO NEXT AVAILABLE SPOT	11538
15273	78CE	05C9		INCT	SOFT	PUT POINT TO NEXT DIGIT	11539
15274	78D0	0604		DEC	R4		11540
15275	78D2	15FA		JGT	LEFTJST+2	SHIFT ALL DIGITS	11541
15276	78D4	0649		DECT	SOFT		11542
15277	78D6	C660	34E0	MOV	BLANK,*SOFT	IF ALL DIGITS HAVE BEEN SHIFTED ADD A	11543
15278	78DA	0A16		SLA	R6,1	BLANK TO BUFFER	11544
15279	78DC	6246		S	R6,SOFT	REPOINT TO MOST SIGNIFICANT DIGIT	11545
15280	78DE	0916		SRL	R6,1		11546
15281	78E0	0586		INC	R6	INCREMENT FILL COUNT	11547
15282	78E2	10E0		JMP	CHKBUF		11548

15343	7942	C1F9		MOV	*SOFT+,R7	R7,R8 = EXTENT	11585
15344	7944	C239		MOV	*SOFT+,R8		11586
15345	7946	1004		JMP	*+10		11587
15346	7948	C1C1		MOV	R1,R7		11588
15347	794A	C202		MOV	R2,R8		11589
15348	794C	0229	0004	AI	SOFT,4		11590
15349	7950	C1C7		MOV	R7,R7	IF EXTENT = 0, THEN WAVEFORM IS ZERO	11591
15350	7952	1602		JNE	*+6		11592
15351	7954	0460	7AAE	B	ZEROSCL		11593
15352	7958	0288	7FFF	CI	R8,\$7FFF		11594
15353	795C	1602		JNE	*+6		11595
15354	795E	0460	7ADA	B	SOVRFLW		11596
15355	7962	C060	33FB	MOV	FP204,R1	LOAD R1,R2 WITH A FLOATING POINT 20	11597
15356	7966	C0A0	33FA	MOV	FP20E,R2		11598
15357	796A	0649		DECT	SOFT	PUSH FP (2*SMAX) ONTO SOFTSTACK	11599
15358	796C	04D9		CLR	*SOFT		11600
15359	796E	0649		DECT	SOFT		11601
15360	7970	C660	3416	MOV	SMAX,*SOFT		11602
15361	7974	A650	3416	A	SMAX,*SOFT		11603
15362	7978	0420	70AA	BLWP	FPMPY	MULTIPLY BY 20 TO GET DIVISION VALUE	11604
15363	797C	0649		DECT	SOFT		11605
15364	797E	04D9		CLR	*SOFT	PUSH FP (ZMAX) ONTO SOFTSTACK	11606
15365	7980	0649		DECT	SOFT		11607
15366	7982	C660	3418	MOV	ZMAX,*SOFT		11608
15367	7986	0420	70AA	BLWP	FPMPY	MULTIPLY BY 20 TO GET DIVISION VALUE	11609
15368	798A	C079		MOV	*SOFT+,R1	MOVE ZMAX TO R1,R2	11610
15369	798C	C099		MOV	*SOFT,R2		11611
15370	798E	C648		MOV	R8,*SOFT		11612
15371	7990	0649		DECT	SOFT		11613
15372	7992	C647		MOV	R7,*SOFT	PUSH EXTENT ONTO SOFTSTACK	11614
15373	7994	0420	7050	BLWP	FPDIV	EXTENT / ZMAX	11615
15374	7998	C2F9		MOV	*SOFT+,R11		11616
15375	799A	C339		MOV	*SOFT+,R12		11617
15376	799C	C079		MOV	*SOFT+,R1	MOVE (2*SMAX-1) TO R1,R2	11618
15377	799E	C099		MOV	*SOFT,R2		11619
15378	79A0	C644		MOV	R4,*SOFT		11620
15379	79A2	0649		DECT	SOFT		11621
15380	79A4	C643		MOV	R3,*SOFT	PUSH PK-PK ONTO SOFTSTACK	11622
15381	79A6	0420	7050	BLWP	FPDIV	PK-PK / (2*SMAX-1)	11623
15382	79AA	C04B		MOV	R11,R1		11624
15383	79AC	C08C		MOV	R12,R2		11625
15384	79AE	0420	7524	BLWP	FPCMPR	TEMPSCL = MAX(EXTENT/ZMAX, PK-PK/(2*SMAX-1))	11626
15385	79B2	1103		JLT	*+8		11627
15386	79B4	CE41		MOV	R1,*SOFT+		11628
15387	79B6	C642		MOV	R2,*SOFT		11629
15388	79B8	0649		DECT	SOFT		11630
15389	79BA	C807	09AA	MOV	R7,MINART1	SAVE EXTENT	11631
15390	79BE	C808	09AC	MOV	R8,MINART2		11632
15391	79C2	0420	7298	BLWP	FPNL0G	LOGTEMP = LOG(TEMPSCL)	11633
15392	79C6	C060	340E	MOV	LOGEM,R1		11634
15393	79CA	C0A0	3336	MOV	LOGEE,R2		11635
15394	79CE	0420	70AA	BLWP	FPMPY	LOG(X) = LOG(E) * LN(X)	11636
15395	79D2	C079		MOV	*SOFT+,R1		11637
15396	79D4	C099		MOV	*SOFT,R2	POP LOGTEMP OFF STACK BUT KEEP	11638
15397	79D6	0649		DECT	SOFT	IT ON STACK	11639
15398	79D8	06A0	75A5	BL	FPTRNC		11640
15399	79DC	0581		INC	R1	EXP = INT(LOGTEMP) + 1	11641
15400	79DE	06A0	757E	BL	INT2FP		11642
15401	79E2	C2C1		MOV	R1,R11	SAVE EXP FOR FUTURE USE	11643

15402	79E4	C302		MOV	R2,R12		11544
15403	79E6	0420	6F60	BLWP	FPSU3	NLOG = LOGTE4P - EXP	11645
15404	79EA	0203	4000	LI	R3,\$4000		11646
15405	79EE	0204	0001	LI	R4,\$0001	MANT = 1	11647
15406	79F2	C050	3420	MOV	LOG0.5M,R1		11648
15407	79F6	C0A0	3336	MOV	LOG0.5E,R2		11649
15408	79FA	0420	7524	BLWP	FPCM>R	IF LOG(0.5) < NLOG THEN MANT = 1	11650
15409	79FE	110C		JLT	NEATFCT		11651
15410	7A00	04C4		CLR	R4	MANT = .5	11652
15411	7A02	C050	3422	MOV	LOG0.2M,R1		11653
15412	7A06	C0A0	3338	MOV	LOG0.2E,R2		11654
15413	7A0A	0420	7524	BLWP	FPCM>R		11655
15414	7A0E	1104		JLT	NEATFCT	IF LOG(0.2) < NLOG THEN MANT = 0.5	11656
15415	7A10	0203	6666	LI	R3,\$6666	MANT = 0.2	11657
15416	7A14	0204	FFFE	LI	R4,\$FFFE		11658
15417	7A18	CE43		NEATFCT MOV	R11,*SOFT+		11659
15418	7A1A	C64C		MOV	R12,*SOFT		11660
15419	7A1C	0649		DECT	SOFT	PUT EXP ON STACK	11661
15420	7A1E	C060	3410	MOV	LN10M,R1		11662
15421	7A22	C0A0	3332	MOV	LN10E,R2		11663
15422	7A26	0420	70AA	BLWP	FPMPY	10**X = E**(X*LN(10))	11664
15423	7A2A	0420	7332	BLWP	FPNEXP		11665
15424	7A2E	C043		MOV	R3,R1		11666
15425	7A30	C084		MOV	R4,R2		11667
15426	7A32	0420	70AA	BLWP	FPMPY	NEATFACT = MANT * 10**EXP	11668
15427	7A36	C060	33F8	MOV	FP20M,R1		11669
15428	7A3A	C0A0	33FA	MOV	FP20E,R2		11670
15429	7A3E	0420	70AA	BLWP	FPMPY	VEXP = VERTICAL SCALE * 20	11671
15430	7A42	C079		MOV	*SOFT+,R1		11672
15431	7A44	C099		MOV	*SOFT,R2	SAVE NEW VEXP	11673
15432	7A46	C660	333A	MOV	FP1E,*SOFT		11674
15433	7A4A	0649		DECT	SOFT	PUT A 1 ON STACK	11675
15434	7A4C	C660	3404	MOV	FP1M,*SOFT		11676
15435	7A50	0420	7050	BLWP	FPOIV	INVERT VERTICAL SCALE	11677
15436	7A54	C839	09AE	MOV	*SOFT+,MAXART1		11678
15437	7A58	C819	0980	MOV	*SOFT,MAXART2	PASS NEW SCALE TO CALLER	11679
15438							
15439				*****	CALCULATE NEW ZERO REFERENCE POSITION	*****	
15440							
15441	7A5C	C660	09AC	MOV	MINART2,*SOFT	VZR = -SGN(WMEAN)*INT(MAX(EXTENT/SCLFACT-	11680
15442	7A60	0649		DECT	SOFT	SMAX,0)	11681
15443	7A62	C660	09AA	MOV	MINART1,*SOFT		11682
15444	7A66	0420	7046	BLWP	FPOIVZ	EXTENT/SCLFACT	11683
15445	7A6A	C139		MOV	*SOFT+,R4		11684
15446	7A6C	6120	3416	S	SMAX,R4	EXTENT/SCLFACT-SMAX	11685
15447	7A70	1501		JST	**4		11686
15448	7A72	04C4		CLR	R4		11687
15449	7A74	0206	0666	LI	R6,\$0666	LOAD R5 WITH ONE DIVISION	11688
15450	7A78	04C3		CLR	R3		11689
15451	7A7A	3CC6		DIV	R6,R3	DIVIDE DIVISIONS BY ONE DIVISION	11690
15452	7A7C	0A14		SLA	R4,1	ROUND TO NEAREST DIVISION	11691
15453	7A7E	8184		C	R4,R5		11692
15454	7A80	1101		JLT	**4		11693
15455	7A82	0583		INC	R3		11694
15456	7A84	0206	6666	LI	R6,\$6666	LOAD DIVISION VALUE * 16	11695
15457	7A88	3CC6		MPY	R6,R3	MULTIPLY TO GET DIVISIONS * 16	11696
15458	7A8A	0207	0003	LI	R7,3		11697
15459	7A8E	0914		SFTVZR SRL	R4,1	DIVIDE RESULT BY 16 TO GET DIVISION VALUE	11698
15460	7A90	0913		SRL	R3,1		11699

15461	7A92	1702		JNC	*+6		11700
15462	7A94	0224	8000	AI	R4,\$8000	DOUBLE-WORD SHIFT	11701
15463	7A98	0607		DEC	R7		11702
15464	7A9A	15F9		JGT	SHFTVZR		11703
15465	7A9C	0914		SRL	R4,1	ROUND ON LAST SHIFT	11704
15466	7A9E	1703		JNC	*+8		11705
15467	7AA0	0584		INC	R4		11706
15468	7AA2	1901		JNO	*+4		11707
15469	7AA4	0604		DEC	R4		11708
15470	7AA6	C145		MOV	R5,R5	MULIPLY TIMES -SIGN(WMEAN)	11709
15471	7AA8	1508		JST	NEWVZR	NOTE--- VOFFAB IN RAM IS NEGATED	11710
15472	7AAA	0504		NEG	R4		11711
15473	7AAC	1009		JMP	NEWVZR		11712
15474	7AAE	0201	5000	ZEROSCL LI	R1,\$5000	VERTICAL SCALE OF 1V	11713
15475	7AB2	0202	0005	LI	R2,\$0005		11714
15476	7AB6	04C4		CLR	R4	VERTICAL ZERO OF 0V	11715
15477	7AB8	04E0	D9AE	CLR	MAXART1		11716
15478	7ABC	04E0	D9B3	CLR	MAXART2		11717
15479	7AC0	C804	D9AA	NEWVZR MOV	R4,MINART1	PASS OFFSET TO CALLER	11718
15480	7AC4	C1A0	DA34	MOV	WOHEAD,R6		11719
15481	7AC8	C146		MOV	R6,R5	SETUP HEADER FOR THIS WAVEFORM	11720
15482	7ACA	A160	3338	A	VEXP,R5	(VEXP & VOFFAB)	11721
15483	7ACE	C041		MOV	R1,*R5+		11722
15484	7AD0	C542		MOV	R2,*R5		11723
15485	7AD2	A1A8	3348	A	VOFFAB,R6		11724
15486	7AD6	C584		MOV	R4,*R6		11725
15487	7AD8	0380		RTWP			11726
15488	7ADA	0206	1333	SOVRFLW LI	R6,\$1333	PICK SCALE FACTOR TO FIT THE MAXIMUM	11727
15489	7ADE	0207	8001	LI	R7,\$8001	REPRESENTABLE VALUES (POS AND NEG)	11728
15490	7AE2	C806	D9AE	MOV	R6,MAXART1	WITHIN + OR - 3 DIVISIONS ON SCREEN	11729
15491	7AE6	C807	D9B0	MOV	R7,MAXART2		11730
15492	7AEA	0201	7FFF	LI	R1,\$7FFF	SET SCALE FACTOR TO MAX	11731
15493	7AEE	0202	7FFF	LI	R2,\$7FFF		11732
15494	7AF2	04C4		CLR	R4	SET VERTICAL OFFSET TO ZERO	11733
15495	7AF4	10E5		JMP	NEWVZR		11734

***** ROM HEADER FOR HIGH ROMS *****

11735

15497			*				
15498			*	DEFINE ROM HEADER FOR HIGH ORDER ROMS			
15499			*				
15500		7FEC		ORG \$7FE0	ROM TAILER FOR \$4000		11736
15501							
15502	7FE0	0404		WORD \$0404,\$1011	ROM PART NO. XXXX FROM 150-XXXX-00		11737
	7FE2	1011					
15503	7FE4	34		FCC '44'	HIGH ORDER ADDRESS NYBBLE (I.E. 4XXX)		11738
	7FE5	34					
15504	7FE6	4D		FCC 'ML'	M = MOST SIGNIFICANT, L = LEAST SIGNIFICANT		11739
	7FE7	4C					
15505	7FE8	30		FCC '0101'	ROM VERSION '01'		11740
	7FE9	31					
	7FEA	30					
	7FEB	31					
15506	7FEC	0000		WORD 280	SPARE		11741
15507							
15508	7FF0	0000		WORD \$0000,\$0000	PROM PART NO. XXXX FROM 160-XXXX-00		11742
	7FF2	0000					
15509	7FF4	34		FCC '44'	HIGH ORDER ADDRESS NYBBLE (I.E. 4XXX)		11743
	7FF5	34					
15510	7FF6	4D		FCC 'ML'	M = MOST SIGNIFICANT, L = LEAST SIGNIFICANT		11744
	7FF7	4C					
15511	7FF8	30		FCC '0000'	PROM VERSION '00'		11745
	7FF9	30					
	7FFA	30					
	7FFB	30					
15512	7FFC	0000		WORD 280	CHECK SUMS		11746
15513							


```

15515 *****
15516 *
15517 * FOLLOWING VARIABLES MUST BE IN RAM *
15518 * *
15519 *****
15520 *
15521 * FOLLOWING VARIABLES ARE RUN-TIME SET *
15522 *
15523 D900 D900 ORG $0900 11748
15524 D900 0004 VCRS BSS 4 VERTICAL DATA POINTS FOR CURSORS 11749
15525
15526 D904 0002 DIAGINT BSS 2 DIAGNOSTICS CONTROL FOR INTERRUPTS 1-8,14,15 11750
15527 D906 0002 INT7854 BSS 2 DIAGNOSTICS CONTROL FOR INTERRUPTS 9-13 11751
15528
15529 D908 0002 BACKUP1 BSS 2 BATTERY BACKUP VARIABLES 11752
15530 D90A 0002 BACKJ2 BSS 2 BACKJ2 = INV(BACKUP1) => BATTERY BACKUP 11753
15531 *
15532 * GPIB VARIABLES *
15533 *
15534 D90C 000C CMDBUF BSS 12 GPIB INPUT COMMAND BUFFER 11754
15535 000A CMDSZ EQU 10 MAXIMUM COMMAND LENGTH 11755
15536 D918 0002 CMDINDX BSS 2 CMDBUF INDEX 11756
15537 D91A 0002 GPI3ADR BSS 2 GPIB ADDRESS & MODE SWITCH STORAGE 11757
15538 D91C 0002 CNSWFH BSS 2 READX TYPE (-=CONSTANT, 0=NONE, +=WAVEFORM) 11758
15539 D91E 0002 RSVENBL BSS 2 RSV ENABLE FLLAGS 11759
15540 D920 0002 GFORMAT BSS 2 GPIB CONVERSION FORMAT FOR 'FP2ASC' 11760
15541 D922 0002 ONOFF BSS 2 GPIB ONLINE STATUS (-=OFFLINE, 0+=ONLINE) 11761
15542 D924 0002 TRMTPRE BSS 2 TERMINATOR TYPE (0=<CR>^EOI, <>0=<CR><.F>^EOI) 11762
15543 D926 0002 TLTLLJ BSS 2 TALK-LISTEN STATUS (-=TO, 0=TL, +=LO) 11763
15544 D928 0002 REMKEY BSS 2 REMOTE KEY EXECUTING (-=YES, 0=NO) 11764
15545 D92A 0002 SEOI BSS 2 SET <EOI> WITH LAST BYTE 11765
15546 D92C 0002 GPI3R12 BSS 2 TEMPORARY STORAGE FOR GPIB'S R12 11766
15547 D92E 0002 STOPDCL BSS 2 DCL STOP FLAG (-=DCL, 0=STOP KEY) 11767
15548 D930 0002 INTGPIB BSS 2 CURRENT GPIB INTERRUPT STATUS 11768
15549 D932 0002 INTGPIB1 BSS 2 CURRENT GPIB INTERRUPT STATUS 11769
15550 D934 0004 MNUGPIB BSS 4 BUFFER FOR 3AD GPIB KEY MNEMONIC 11770
15551 D938 0002 INTFLAG BSS 2 FLAGS FOR SYSTEM INTERRUPTS 11771
15552 D93A 0002 ACPICMD BSS 2 FLAG INDICATING GPIB ACCEPTING COMMANDS 11772
15553 D93C 0002 STPGPIB BSS 2 FLAG INDICATING 'STOP' FROM GPIB 11773
15554 D93E 0002 IGNPRE BSS 2 IGNORE REMAINDER OF GPIB RECORD 11774
15555 D940 0002 INOJT BSS 2 GPIB I/O FLAG (-=INPUT, 0=OUTPUT) 11775
15556 D942 0002 GPI3OPT BSS 2 GPIB BOARD FLAG (-=NO BOARD, 0=BOARD) IN) 11776
15557 D944 0002 GPI3KEY BSS 2 FLAG INDICATING GPIB KEY EXECUTING 11777
15558 D946 0002 KEYXFR BSS 2 POINTER TO BLWP THROUGH FOR KEY HANDLER 11778
15559 D948 0002 KEYXFR2 BSS 2 CONTIGUOUS TO ABOVE --- REST ON POWER UP 11779
15560 D94A 0002 FATAL BSS 2 ERROR NUMBER TO DISPLAY ON CRT -- INT 11780
15561 D94C 0002 WARNING BSS 2 NON-FATAL ERROR TO DISPLAY ON CRT--- INT 11781
15562 D94E 0002 GOLD BSS 2 GOLD KEY FLAG 11782
15563 D950 0004 CURSPLT BSS 4 TWO WORD BFFER FOR CURSOR VECTOR 11783
15564 D954 0002 CJRSJR BSS 2 #CJRSJR DISPL.--NEXT 2 CONTIGUOUS 11784
15565 D956 0002 CJRS1 BSS 2 ELEMENT # OF CURSOR 1 11785
15566 D958 0002 CURS2 BSS 2 ELEMENT # OF CURSOR 2 11786
15567 D95A 0002 OPWFM BSS 2 OPWFM # 11787
15568 D95C 0002 OPWFMD BSS 2 OPWFM DATA ADDRESS 11788
15569 D95E 0002 OPWFMH BSS 2 OPWFM HEADER ADDRESS 11789
15570 D960 0002 CRSOPWD BSS 2 @> OPWFM DATA ADDRESS FOR DISPLAYED OPW 11790
15571 D962 0002 FKEY BSS 2 FRONT PANEL KEY 11791
15572 D964 0002 BJFAVL BSS 2 # OF AVAILA3LE KEY BUFFER WORDS 11792
15573 D966 0001 KEY BSS 1 KEY EXECUTING 11793
    
```


15574	D967	0001	KEYB	BSS 1	CONTIGUOUS TO ABOVE	11794
15575	D968	0002	LASTKEY	BSS 2	LAST KEY EXECUTED/B15=1-NO KEY DISP/B14=1-NO BUSY D11795	11795
15576			*		B13=1-STOP ALLOWED DURING KEY EXECUTION	
15577	D96A	0002	TRUE	BSS 2	TRUE/FALSE FLAG FOR DISPLAY OF WRD (-1=NONE ; 0=F11796	11796
15578	D96C	0002	PRODSPRO	BSS 2	DISPLAY R/O STATUS HOLDER FOR PROGRAM	11797
15579	D96E	0002	PRODSPRT	BSS 2	DISPLLAT REALTIME STATUS HOLDER FOR PROGRAM MODE	11798
15580	D970	0002	RESOLV	BSS 2	RESOLUTION OF DIGITIZED DATA	11799
15581	D972	0002	RJTFLAG	BSS 2	FLAG TO INDICATE NEEDED CHANGES IN READOUT	11800
15582	D974	0002	WFMAXN	BSS 2	MAXIMUM NUMBER OF WAVEFORMS IN MEMORY	11801
15583	D976	0002	IVUSE	BSS 2	KEYBRD BUSY FLAG=0 IF IDLE AND >0 IF KEYS NJST	11802
15584	D978	0002	XYWFM	BSS 2	ZERO = Y-T, POSITIVE = VS # AND ADDRESS	11803
15585	D97A	0004	YTWFM	BSS 4	Y-T PRELOADS FOR WAVEFORMS	11804
15586	D97E	0014	DSP3UF	BSS 20	BUFFER FOR 9 WFM #S + ENDOKEY	11805
15587	D992	0002	SUSGPIB	BSS 2	FLAG INDICATING SUSPENDED GPIB (-=SUSPENDED)	11806
15588	D994	0002	A4REG	BSS 2	ADDRESS MODE REGISTER COPY FOR GPIB	11807
15589	D996	0002	WFM1	BSS 2	WFM #1 FOR WFM ARITH	11808
15590	D998	0002	WFM2	BSS 2	WFM #2 FOR WFM ARITH	11809
15591	D99A	0002	W1HEAD	BSS 2	WFM #1 HEADER ADDRESS FOR WFM ARITH	11810
15592	D99C	0002	W2HEAD	BSS 2	WFM #2 HEADER ADDRESS FOR WFM ARITH	11811
15593	D99E	0002	HVFLAG	BSS 2	HORIZ/VERT FLAG TO CHANGE HD HEADER	11812
15594	D9A0	0002	UNITFLG	BSS 2	= 0 IF BOTH SCALE AND UNITS TO BE CHANGED	11813
15595			*		<> 0 IF ONLY UNITS BLANKED	
15596	D9A2	0002	EXP11	BSS 2	EXP MANT (1) FOR WFM ARITH	11814
15597	D9A4	0002	EXP12	BSS 2	EXP EXP (1) FOR WFM ARITH	11815
15598	D9A6	0002	EXP21	BSS 2	EXP MANT (2) FOR WFM ARITH	11816
15599	D9A8	0002	EXP22	BSS 2	EXP EXP (2) FOR WFM ARITH	11817
15600	D9AA	0002	MINART1	BSS 2	DP MINIMUM POINT FOR WFM ARITH	11818
15601	D9AC	0002	MINART2	BSS 2	SEE ABOVE	11819
15602	D9AE	0002	MAXART1	BSS 2	DP MAXIMUM POINT FOR WFM ARITH	11820
15603	D9B0	0002	MAXART2	BSS 2	SEE ABOVE	11821
15604	D9B2	0002	SWEEPS	BSS 2	SOFTWARE SWEEPS COUNT	11822
15605	D9B4	0002	NEGFLG	BSS 2	NEGATIVE FLAG FOR WFM MATH	11823
15606	D9B6	0002	DATA1	BSS 2	POINTER TO START OF WFM DATA(1) IN WFM ARITH	11824
15607	D9B8	0002	DATA2	BSS 2	POINTER TO START OF WFM DATA(2) IN WFM ARITH	11825
15608	D9BA	0002	PROGRS	BSS 2	>0 IF NJMERIC IN PROGRESS	11826
15609	D9BC	0002	NEWEXPM	BSS 2	TEMP VEXP MANTISSA DURING MATH	11827
15610	D9BE	0002	NEWEXPE	BSS 2	TEMP VEXP EXPONENT DURING MATH	11828
15611	D9C0	0002	HALTIT	BSS 2	=1IF 'STOP' KEY HIT/ 0 OTHERWISE	11829
15612	D9C2	0002	FPTMP	BSS 2	COPY OF FRONT PANEL BUTTON SETTINGS	11830
15613	D9C4	0002	REFRESH	BSS 2	FLAG INDICATING STATUS OF DISPLAY REFRESH	11831
15614	D9C6	0002	REALTIME	BSS 2	FLAG INDICATING DISPLAY TYPE	11832
15615	D9C8	0002	DSPRLT	BSS 2		11833
15616	D9CA	0002	DSPRO	BSS 2		11834
15617	D9CC	0002	DSPCRS	BSS 2	ACQUIRE/DISPLAY CONTROL WORDS	11835
15618	D9CE	0002	DSPWF4	BSS 2		11836
15619	D9D0	0002	ACQWFM	BSS 2		11837
15620	D9D2	0002	ACQWFM1	BSS 2		11838
15621			*			
15622			*		THE NEXT THREE VARIABLES DEFINE THE STARTING ADDRESSES FOR	
15623			*		EACH OF THE 16 LINES IN THE READOUT DISPLAY. THESE MUST BE	
15624			*		KEPT CONTIGUOUS.	
15625			*			
15626	D9D4	001E	LINESTR	BSS 30	STARTING ADDRESSES OF LINES 1-15	11839
15627	D9F2	0002	LINE16	BSS 2	STARTING ADDRESS OF LINE 16	11840
15628	D9F4	0002	ENDTEXT	BSS 2	END ADDRESS OF READOUT BUFFER	11841
15629		DA00		ORG \$DA00		11842
15630	DA00	0004	HCRS	BSS 4	HORIZONTAL DATA POINTS FOR CURSORS	11843
15631	DA04	0008	VRSNUM	BSS 8	PRJM AND RJM VERSION NUMBERS (FOR ID KEY)	11844
15632	DA0C	0002	YZFLAG	BSS 2	FLAG FOR YZERO WAVEFORM HEADER INPUT	11845

15633	DA0E	0002	YZ1 BSS 2	MANTISSA OF YZERO OF WAVEFORM HEADER DURING INPUT	11846
15634	DA10	0002	YZ2 BSS 2	EXPONENT OF YZERO OF WAVEFORM HEADER DURING INPUT	11847
15635	DA12	0002	CYCLE BSS 2	FLAG INDICATING DISPLAY NOT YES DONE WITH CYCLE	11848
15636	DA14	0002	CYCFLAG BSS 2	POWER-UP TEST CYCLE (0=NO, <>0=YES)	11849
15637	DA16	0002	FLICKER BSS 2		11850
15638	DA18	0002	RJUPDT BSS 2		11851
15639	DA1A	0002	TIMER20 BSS 2		11852
15640	DA1C	0002	CROSSFLG BSS 2		11853
15641	DA1E	0002	TEXT0 BSS 2		11854
15642	DA20	0002	TEXT1 BSS 2		11855
15643	DA22	0002	TEXT2 BSS 2		11856
15644	DA24	0002	TEXT3 BSS 2		11857
15645	DA26	0002	VERT4 BSS 2		11858
15646	DA28	0002	HCRZ4 BSS 2		11859
15647	DA2A	0002	GROUND0 BSS 2		11860
15648	DA2C	0002	GROUND1 BSS 2		11861
15649	DA2E	0002	MAXLINE BSS 2	MAX LINE NUMBER FOR USER PROGRAM	11862
15650	DA30	0002	POINT BSS 2	NEEDED BY SOME LEVEL 5 ROUTINES	11863
15651	DA32	0002	POINTVAR BSS 2	POINTER VARIABLE TO DETERMINE INNER PATH THROUGH	11864
15652	DA34	0002	HOLD7 BSS 2	HOLD REG 7 TEMP	11865
15653	DA36	0002	HOLD8 BSS 2	HOLD REG 8 TEMP	11866
15654	DA38	0002	HOLD12 BSS 2	HOLD REG 12 TEMP	11867
15655			*	OTHER LEVEL 5 ROUTINES	
15656	DA3A	0014	KEYBUFR BSS 20	BUFFERED KEYS	11868
15657	DA4E	0014	GOSUBUF BSS 20	RETURN POINTERS FOR GOSUB	11869
15658			*		
15659			*	NOTE: TXTBUFR AND CNVBUFR MUST REMAIN CONTIGUOUS IN	
15660			*	THE ORDER SPECIFIED AS TXTBUFR CONTENTS WILL	
15661			*	SPILL OVER INTO CNVBUFR OCCASIONALLY.	
15662	DA62	0028	TXTBUFR BSS 40	BUFFER FOR PROGRAM TEXT LINE	11870
15663	DA8A	0014	CNVBUF BSS 20		11871
15664	DA9E	0002	NJH3 BSS 2	HOLDER FOR LAST DIGIT OF LINE NUMBER IN PROGRAM	11872
15665	DAA0	0002	PROGRAM BSS 2	-1=REALTIME CALCULATOR USE/0=PROG1=PROGRUN/2=SING	11873
15666	DAA2	0002	PROMODE BSS 2	-1=NON-PROGRAM/ 0=EDIT MODE/ >=1 = ENTRY MODE	11874
15667	DAA4	0002	LINENUM BSS 2	LINE NUMBER TO LOOK FOR IN PROGRAM TEXT	11875
15668	DAA6	0002	LINEPNT BSS 2	ADDRESS FOUND FOR ABOVE LINE NUMBER	11876
15669	DAA8	0002	PROGLN BSS 2	'CURRENT LINE' NUMBER	11877
15670	DAAA	0002	EDITPNT BSS 2	'NEXT COMMAND' POINTER	11878
15671	DAAC	0002	CLPFLAG BSS 2	=1 IF CLEAR PROGRAM UNDER CONSIDERATION	11879
15672	DAAE	0002	LINE99 BSS 2	MAXIMUM LINE # CURRENTLY STORED	11880
15673	DAB0	0002	MOVFLAG BSS 2	=0 IF EDITPOINTER SHOULD BE INCREMENTED BEFORE	11881
15674	DAB2	0002	LNNFLAG BSS 2	=1 IF LNN BEING INPUT IN PROGRAM MODE	11882
15675	DAB4	0002	PROGSTEP BSS 2	LINE NUMBER SINGLE STEP STARTED ON	11883
15676	DAB6	0002	SCRNFLAG BSS 2	FLAG IF SCREEN SHOULD BE UPDATED	11884
15677	DAB8	0002	STPROGF BSS 2	FLAG TO INDICATE PROGRAM WAS STOPPED	11885
15678	DABA	0002	STPFLG BSS 2	FLAG FOR 'STOP IN' STATUS MESSAGE (=-V0)	11886
15679	DABC	0002	PROREALT BSS 2	REALTIME STATUS HOLDER FOR PROGRAM MODE	11887
15680	DABE	0002	CROSSFLAG BSS 2	FLAG =0 TO SEARCH FROM MID-WAY OR =1 TO SEARCH F	11888
15681	DAC0	0002	CROSSNJM BSS 2	NUMBER OF CROSSES REQUESTED IN FWDCROSS/3AKCROSS	11889
15682	DAC2	0002	EJPFLG BSS 2	FLAG <>0 IF AT E-0-P, =0 IF NOT AT E-0-P	11890
15683	DAC4	0002	FALFLG BSS 2	FLAG = 0 IF RISE, -1 FOR FALL	11891
15684	DAC6	0002	BLNKFLG BSS 2	<> 0 IF BLANK STATUS ALREADY UP, 0, OTHERWISE	11892
15685	DAC8	0002	OBUSY BSS 2	FLAG INDICATING STATUS OF 'BUSY' DISPLAY	11893
15686	DACA	0002	OKEY BSS 2	LAST KEY DISPLAYED	11894
15687	DACC	0002	OPROGLN BSS 2	LAST PROGRAM LINE # DISPLAYED	11895
15688	DACE	0002	STEWARN BSS 2	FLAG TO STOP 'STEP' ON WARNING	11896

15689
 15690
 15691

* THE FOLLOWING 11 VARIABLES ARE USED TO MAP RAM
 *

RAM VARIABLES

11747

Address	Label	Value	Symbol	Description	Address
15692			*	THEY MUST REMAIN CONTIGUOUS AND IN THIS ORDER	
15693			*		
15694		DAD0	RAMMAP EQU *		11897
15695	DAD0	0002	KRAM BSS 2	NUMBER OF 1K BLOCK OF RAM IN SYSTEM	11898
15696	DAD2	0002	WDADD BSS 2	ADDRESS OF W0'S DATA	11899
15697	DAD4	0002	WDHEAD BSS 2	ADDRESS OF W0'S HEADER	11900
15698	DAD6	0002	WFMBAS BSS 2	BASE ADDRESS OF WAVEFORM DATA	11901
15699	DAD8	0002	TOPHE0 BSS 2	ADDRESS + 2 OF TOP OF HEADER SPACE	11902
15700	DADA	0002	CONSTR BSS 2	BASE ADDRESS OF CONSTANT STORAGE	11903
15701	DAOC	0002	MAXCNS BSS 2	MAXIMUM NUMBER OF CONSTANTS	11904
15702	DADE	0002	DISMEM BSS 2	BASE ADDRESS OF READOUT DISPLAY MEMORY	11905
15703	DAE0	0002	PRGMEM BSS 2	BASE ADDRESS FOR PROGRAM STORAGE	11906
15704	DAE2	0002	MAXPROG BSS 2	TOP ADDRESS FOR PROGRAM STORAGE	11907
15705	DAE4	0002	RAMOPT BSS 2	ADDRESS OF MAX WFM # TABLE BY RESOLUTION	11908
15706		0016	MAPSIZE EQU *-RAMMAP		11909
15707			*		
15708	DAE6	0002	RQSNUM BSS 2	SERVICE REQUEST NUMBER	11910
15709			*		
15710		A000	RAMBASE EQU \$A000		11911
15711		D000	RAMTOP EQU \$D000		11912
15712		0008	MAXKRAM EQU 8		11913
15713		0000	R0MBASE EQU \$0000		11914
15714		4000	R0MTOP EQU \$4000		11915
15715		E500	DIAG EQU \$E500	TEST WORD IN DIAGNOSTICS RAM	11916
15716		D800	ORG \$D800		11917
15717	DB00	0020	WPKYTR BSS 32		11918
15718	DB20	0020	WPKYS BSS 32		11919
15719	DB40	0020	WPKBFR BSS 32		11920
15720	DB60	0020	WPKVL1 BSS 32		11921
15721	DB80	0020	WPKVL2 BSS 32		11922
15722	DBA0	0020	WPKVL3 BSS 32		11923
15723	DBC0	0020	WPKVL4 BSS 32		11924
15724	DBE0	0020	WPKVL5 BSS 32		11925
15725	DC00	0020	WPK_K BSS 32		11926
15726	DC20	0020	WPKSP BSS 32		11927
15727	DC40	0020	WPKSWP BSS 32		11928
15728	DC60	0020	WPKPI3 BSS 32		11929
15729		DC70	WPKINT EQU *-16		11930
15730	DC80	0020	WPKINT1 BSS 32		11931
15731			*		
15732			*		

```

15734          9600          ORG    $9600
15735          *****
15736          *
15737          * PROBLEM #1 - PATCH #1 (1 OF 2)
15738          *
15739          * WITH GPIB TERMINATOR MODE OF EOI/LF, COMMANDS ARE NOT CORRECTLY
15740          * TERMINATED. THE CHECK FOR LF BEING A TERMINATOR WAS NOT MADE.
15741          *
15742  9600  D320  E05C  PATCH1  MOVB  R7R,R12      READ GPIB DATA BYTE
15743  9604  C820  D924  D924      MOV   TRMYPE,TRMYPE  CHECK TERMINATOR TYPE
15744  960A  1306                JEQ   ENDP1          IF EOI ONLY, DON'T LOOK FOR LF
15745  960C  9320  331F          CB    LFB,R12       IF EOI/LF, IS THE BYTE A LF?
15746  9610  1603                JNE  ENDP1          NO, THEN NOT END YET
15747  9612  E820  337A  D930      SOC   END,INTGPIB   YES, SET END STATUS BIT
15748  9618  0460  2498          ENDP1  B    BACK1    RETURN FROM PATCH #1
15749          *
15750          * END OF PATCH #1 OF PROBLEM #1
15751          *
15752          *****
15753          *
15754          * PROBLEM #1 - PATCH #2 (2 OF 2)
15755          *
15756          * WITH GPIB TERMINATOR MODE OF EOI/LF, COMMANDS ARE NOT CORRECTLY
15757          * TERMINATED. THE CHECK FOR LF BEING A TERMINATOR WAS NOT MADE.
15758          *
15759  961C  D050  E05C  PATCH2  MOVB  R7R,R1      READ GPIB DATA BYTE
15760  9620  C820  D924  D924      MOV   TRMYPE,TRMYPE  CHECK TERMINATOR TYPE
15761  9626  1306                JEQ   ENDP2          IF EOI ONLY, DON'T LOOK FOR LF
15762  9628  9060  331F          CB    LFB,R1       IF EOI/LF, IS THE BYTE A LF?
15763  962C  1605                JNE  ENDP2          NO, THEN NOT END YET
15764  962E  E820  337A  D930      SOC   END,INTGPIB   YES, SET END STATUS BIT
15765  9634  C320  D930          MOV   INTGPI3,R12   LOAD NEW INTERRUPT STATUS INTO R12
15766  9638  0450  24EE          ENDP2  B    BACK2    RETURN FROM PATCH #2
15767          *
15768          * END OF PATCH #2 OF PROBLEM #2
15769          *
15770          *****
15771          *
15772          * PROBLEM #2 - PATCH #3 (1 OF 1)
15773          *
15774          * 'REAJX' DOESN'T TERMINATE WAVEFORM INPUT CORRECTLY. ALL
15775          * TRAILING DELIMITERS SHOULD BE IGNORED.
15776          *
15777  963C  06A0  3124  PATCH3  BL    GPIBIN      INPUT NEXT DATA BYTE FROM GPIB
15778  9640  06A0  3206          BL    CHKEOI      DID <EOI> COME WITH THIS DATA BYTE?
15779  9644  1307                JEQ   ENDP3          YES, EVERYTHING IS OK
15780  9646  06A0  3112          BL    CHKDLN      NO, THEN DATA BYTE MUST BE A DELIMITER
15781  964A  13F8                JEQ   PATCH3        IF BYTE IS A DELIMITER, CONTINUE LOOKING
15782  964C  0720  D93E          SETJ IGNORE      IF NOT, IGNORE THE REST
15783  9650  0460  2A08          B    BACK3        ERROR RETURN FROM PATCH #3
15784  9654  0450  2A1D          ENDP3  B    RESIN   NORMAL RETURN FROM PATCH #3
15785          *
15786          * END OF PROBLEM #2
15787          *
15788          *****
15789          *
15790          * PROBLEM #3 - PATCH #4 (1 OF 1)
15791          *
15792          * '>TEXT' DOESN'T TERMINATE TEXT INPUT CORRECTLY. ALL
    
```

```

15793          * TRAILING DELIMITERS SHOULD BE IGNORED.
15794          *
15795 9658 06A0 3124 PATCH4 BL GPIBIN INPUT NEXT DATA BYTE FROM GPIB
15796 965C 06A0 3206          BL CHKEJI DID <EJI> COME WITH THIS DATA BYTE?
15797 9660 1307          JEQ ENDP4 YES, EVERYTHING IS OK
15798 9662 06A0 3112          BL CHKDL4 NO, THEN DATA BYTE MUST BE A DELIMITER
15799 9666 13F8          JEQ PATCH4 IF BYTE IS A DELIMITER, CONTINUE LOOKING
15800 9668 0720 093E          SETJ IGNORE IF NOT, IGNORE THE REST
15801 966C 0460 2ABE          B BACK4 ERROR RETURN FROM PATCH #4
15802 9670 0460 2A7A          ENDP4 B RTN2TXT NORMAL RETURN FROM PATCH #4
15803          *
15804          * END OF PROBLEM #3
15805          *
15806          *
15807          *
15808          * PROBLEM #5 - PATCH #6 (1 OF 1)
15809          *
15810          * 'F STOP' FROM GPIB SUSPENDS COMMAND INPUT; IT SHOULDN'T.
15811          *
15812 9674 130D PATCH46 JEQ ISSTOP IF NO SHIFT KEY, THEN NOT 'F STOP'
15813 9676 C060 093C          MOV STPGPIB,R1 IS GPIB INPUT SUSPENDED?
15814 967A 1308          JEQ ENDP6 NO, EVERYTHING IS OK
15815 967C 0820 0994 E068          MOVB AMREG,R2W YES, RESET RFD HOLDOFF
15816 9682 0820 3386 E06C          MOVB RFDR,R3W COMPLETE HANDSHAKE
15817 9688 04E0 093C          CLR STPGPIB RESET SUSPENDED FLAG
15818 968C 0460 030C          ENDP6 B BUFIN RETURN
15819 9690 C820 333A 09C0          ISSTOP MOV C1,HALTIT SET HALT FLAG
15820 9696 0460 03A6          B BACK6 RETURN
15821          *
15822          * END OF PROBLEM #5
15823          *
15824          *
15825          *
15826          * PROBLEM #6 - PATCH #7 (1 OF 1)
15827          *
15828          * 'GND' DOES NOT CALCULATE SMALL POSITIVE GROUND CORRECTLY
15829          *
15830 969A C104 PATCH7 MOV R4,R4 CHECK SIGN OF RESULT
15831 969C 1102          JLT ENDP7 IF NEGATIVE, NEGATE FOR DIVISION
15832 969E 0460 1090          B GNDPCS IF POSITIVE, DIVISION WILL WORK
15833 96A2 0460 1084          ENDP7 B BACK7
15834          *
15835          * END OF PROBLEM #7
15836          *
15837          *
15838          *
15839          * PROBLEM #10 - PATCH #11 (1 OF 1)
15840          *
15841          * 'SENJX' SHOULD ADD <CR><LF> AFTER PREAMBLE IN TALK-LISTEN EJI/LF MODE
15842          *
15843 96A6 0050 331E PATCH11 MOVB CRB,R1 ADD A <CR> AFTER THE ";"
15844 96AA C0A0 0926          MOV TOTLLQ,R2 CHECK GPIB MODE
15845 96AE 1607          JNE ENDP11 NO <LF> IN TALK-ONLY
15846 96B0 C0A0 0924          MOV TRMTYPE,R2 CHECK TERMINATOR TYPE
15847 96B4 1304          JEQ ENDP11 NO <LF> IN EOI ONLY MODE
15848 96B6 06A0 3154          BL GPIBOUT
15849 96BA 0050 331F          MOV3 LFB,R1 ADD <LF> AFTER <CR>
15850 96BE 0460 2C3E          ENDP11 B BACK11 RETURN FROM PATCH #11
15851          *
    
```

```

15852 * END OF PROBLEM #10
15853 *
15854 *****
15855 *
15856 * PROBLEM #12 - PATCH #13 (1 OF 3)
15857 *
15858 * 7854 WON'T ACCEPT MULTIPLE OUTPUT COMMANDS IN ONE MESSAGE
15859 *
15860 96C2 C060 D992 PATCH13 MOV SUSGP1B,R1 IS GPIB SUSPENDED IN COMMAND INPUT?
15861 96C6 1302 JEQ *+6 NO, CONTINUE WAITING
15862 96C8 0460 2740 B READERR YES, ERROR (PREVENT GPIB FROM HANGING)
15863 96CC C050 D93A MOV ACPTCMD,R1 IS A COMMAND MESSAGE STILL COMING?
15864 96D0 11F8 JLT PATCH13 YES, WAIT FOR IT TO COMPLETE
15865 96D2 C820 333A D944 MOV C1,GPIBKEY FLAG INDICATING 'TEXT' KEY EXECUTING
15866 96D8 0460 2AD2 B BACK13 RETURN FROM PATCH #13
15867 *
15868 * END OF PATCH #13 OF PROBLEM #12
15869 *
15870 *****
15871 *
15872 * PROBLEM #12 - PATCH #14 (2 OF 3)
15873 *
15874 * 7854 WON'T ACCEPT MULTIPLE OUTPUT COMMANDS IN ONE MESSAGE
15875 *
15876 96DC C060 D992 PATCH14 MOV SUSGP1B,R1 IS GPIB SUSPENDED IN COMMAND INPUT?
15877 96E0 1302 JEQ *+6 NO, CONTINUE WAITING
15878 96E2 0460 2740 B READERR YES, ERROR (PREVENT GPIB FROM HANGING)
15879 96E6 C050 D93A MOV ACPTCMD,R1 IS A COMMAND MESSAGE STILL COMING?
15880 96EA 11F8 JLT PATCH14 YES, WAIT FOR IT TO COMPLETE
15881 96EC C820 333C D944 MOV C2,GPIBKEY FLAG INDICATING 'SENDX' KEY EXECUTING
15882 96F2 0460 2840 B BACK14 RETURN FROM PATCH #14
15883 *
15884 * END OF PATCH #14 OF PROBLEM #12
15885 *
15886 *****
15887 *
15888 * PROBLEM #12 - PATCH #15 (3 OF 3)
15889 *
15890 * 7854 WON'T ACCEPT MULTIPLE OUTPUT COMMANDS IN ONE MESSAGE
15891 *
15892 96F6 C060 D992 PATCH15 MOV SUSGP1B,R1 IS GPIB SUSPENDED IN COMMAND INPUT?
15893 96FA 1302 JEQ *+6 NO, CONTINUE WAITING
15894 96FC 0460 2740 B READERR YES, ERROR (PREVENT GPIB FROM HANGING)
15895 9700 C050 D93A MOV ACPTCMD,R1 IS A COMMAND MESSAGE STILL COMING?
15896 9704 11F8 JLT PATCH15 YES, WAIT FOR IT TO COMPLETE
15897 9706 C820 333E D944 MOV C3,GPIBKEY FLAG INDICATING 'TEXT' KEY EXECUTING
15898 970C 0460 2C92 B BACK15 RETURN FROM PATCH #15
15899 *
15900 * END OF PATCH #15 OF PROBLEM #12
15901 *
15902 *****
15903 *
15904 * PROBLEM #13 - PATCH #16 (1 OF 1)
15905 *
15906 * 'READX' WILL NOT ACCEPT ONLY A WAVEFORM CURVE (NO PREAMBLE)
15907 *
15908 9710 0582 PATCH16 INC R2 INCREMENT BYTE COUNT OF STRING
15909 9712 8802 2F76 C R2,WFMPRE DID ANY CHARACTERS MATCH 'WFMPRE' ?
15910 9715 0460 276E B BACK16 RETURN FROM PATCH #16
    
```



```

15911 *
15912 * END OF PROBLEM #13
15913 *
15914 *****
15915 *
15916 * PROBLEM #15 - PATCH #18 (1 OF 2)
15917 *
15918 * 'STOP' AFTER 'SAVE' IN PROGRAM ENTRY MODE DOESN'T CLEAR
15919 * COMMAND BUFFER
15920 *
15921 971A C050 DAA0 PATCH18 MOV PROGRAM,R1
15922 971E 1607 JNE ENDP18 DON'T WIPE BUFFER DURING PROGRAM INPUT
15923 9720 C050 D976 MOV INUSE,R1 IS 7854 BUSY?
15924 9724 1602 JNE *+6
15925 9726 0450 01CE B BUFRCHK
15926 972A 0450 00DE B HIPBUFR
15927 972E 0460 0078 ENDP18 B BUFRCHK+4
15928 *
15929 * END OF PATCH #18 OF PROBLEM #15
15930 *
15931 *****
15932 *
15933 * PROBLEMS #14 & #15 - PATCH #19 (2 OF 2)
15934 *
15935 * 'RJM' AFTER ERROR DOESN'T INCREMENT TO NEXT COMMAND
15936 *
15937 * 'STOP' AFTER 'SAVE' IN PROGRAM ENTRY MODE DOESN'T CLEAR
15938 * COMMAND BUFFER
15939 *
15940 9732 C050 DAA0 PATCH19 MOV PROGRAM,R1 CURRENTLY IN PROGRAM INPUT MODE?
15941 9736 1302 JEQ *+6 YES, NO PROGRAM TO STOP
15942 9738 0720 DAA0 SETO PROGRAM NO, STOP PROGRAM
15943 973C C820 33E6 D954 MOV BUFRLEN,BUFRVAL RESET INDICATOR FOR CLEAR BUFFER
15944 9742 0720 D972 SETO RDTFLAG UPDATE ALL READOUT LINES
15945 9746 C020 D94A MOV FATAL,R0 HAS IT AN ERROR?
15946 974A 1607 JNE ENDP19 NO, DON'T CHECK BUFFER
15947 974C 0601 DEC R1 IS 7854 IN STEP MODE?
15948 974E 1303 JEQ *+8 NO, ADVANCE TO NEXT COMMAND
15949 9750 C820 333A DAB0 MOV C1,MOVFLAG YES, ADVANCE TO NEXT COMMAND
15950 9756 0450 01CE B BUFRCHK
15951 975A 0450 0100 ENDP19 B RSTGPI3
15952 *
15953 * END OF PATCH #19 OF PROBLEMS #14 & #15
15954 *
15955 *****
15956 *
15957 * PROBLEM #16 - PATCH #20 (1 OF 1)
15958 *
15959 * BUFFER OVERFLOW IS NOT CHECKED FOR IN VALUE INPUT
15960 *
15961 975E 130B PATCH20 JEQ VALIN IF DELIMITER VALUE MAY BE IN
15962 9760 DC01 MOVB R1,*R0+ SAVE DIGIT IN BUFFER
15963 9762 8600 33E2 C R0,GND1BAS CHECK FOR BUFFER OVERFLOW
15964 9766 1202 JLE DLHVAL
15965 9768 C020 33E2 MOV GND1BAS,R0 IF OVERFLOW, PUT REMAINDER IN LAST BYTE
15966 976C 06A0 3205 DLHVAL BL CHKEJI DID <EJI> COME WITH THIS BYTE?
15967 9770 1302 JEQ VALIN YES, VALUE IS IN
15968 9772 0450 323A MCRVA B NXTVAL NO, GET NEXT DIGIT
15969 9776 8600 33E0 VALIN C R0,GPIBBUF WAS ONLY A DELIMITER FOUND?
    
```

15970	977A	1905		J4	ENDP20	NO, VALUE IS IN	
15971	977C	06A0	3206	BL	CHKEJI	YES, DID <EOI> COME WITH IT?	
15972	9780	16F8		JNE	MORVAL	NO, SET NEXT DATA BYTE	
15973	9782	DC20	3797	MOV3	ASCII0,*R0+	YES, DEFAULT VALUE IS ZERO	
15974	9786	0450	3246	ENDP20	B	BACK20	
15975				*			
15976				*	END OF PROBLEM #15		
15977				*			
15978				*****			
15979				*			
15980				*	PROBLEM #17 - PATCH #21 (1 OF 1)		
15981				*			
15982				*	SWITCH FROM DATA INPUT TO COMMAND INPUT DOESN'T OCCUR CORRECTLY		
15983				*			
15984	978A	C050	D940	PATCH21	MOV	INOUT,R1	CHECK GPIB I/O STATUS
15985	978E	13FD		JEQ	PATCH21		IF IDLE, WAIT
15986	9790	1102		JLT	*+6		
15987	9792	0460	303C	B	ERRLSTN		IF OUTPUT, ERROR
15988	9796	C820	D930	D932	MOV	INTGPIB,INTGPIB1	SAVE INTERRUPT STATUS FOR THIS BYTE
15989	979C	04E0	D940	CLR	INOUT		FLAG GPIB I/O IDLE
15990	97A0	C060	D994	MOV	AMREG,R1		
15991	97A4	E060	337C	SOC	HDLA,R1		HOLD OFF RFD ON DATA BYTE
15992	97A8	D801	E068	MOV3	R1,R2W		
15993	97AC	D060	E05C	MOV3	R7R,R1		READ DATA BYTE
15994	97B0	C820	D924	D924	MOV	TRMTYPE,TRMTYPE	CHECK TERMINATOR TYPE
15995	97B6	1305		JEQ	ENDP21		IF <EOI> ONLY THEN DONE
15996	97B8	9801	331F	CB	R1,LFB		CHECK FOR <EOI> OR <LF> TERMINATOR
15997	97BC	1603		JNE	ENDP21		IF NOT <LF> THEN DONE
15998	97BE	E820	337A	D932	SOC	END,INTGPIB1	IF <LF>, SET END STATUS BIT
15999	97C4	0981		ENDP21	SRL	R1,8	SAVE DATA BYTE IN LOW BYTE
16000	97C6	D060	D932	MOV3	INTGPIB1,R1		CHECK <EOI> STATUS
16001	97CA	2068	337A	CDC	END,R1		
16002	97CE	1602		JNE	*+6		IF NO <EOI> THEN MORE TO COME
16003	97D0	04E0	D944	CLR	GPIBKEY		FLAG END OF DATA INPUT (COMMANDS NEXT)
16004	97D4	D820	D994	E058	MOV3	AMREG,R2W	FINISH HANDSHAKE
16005	97DA	D820	3386	E06C	MOV3	RFOR,R3W	
16006	97E0	0A81		SLA	R1,8		PUT DATA BYTE BACK INTO HIGH BYTE
16007	97E2	045B		B	*R11		
16008				*			
16009				*	END OF PROBLEM #17		
16010				*			
16011				*****			
16012				*			
16013				*	PROBLEM 18 - PATCH #22 & #23		
16014				*			
16015				*	CORRECT INCOMPATIBILITIES BETWEEN 7854 AND 4924		
16016				*			
16017	97E4	000A		DLMT3L1	WORD	10	TEN DELIMITERS
16018	97E6	202C			WORD	\$202C,\$0D0A,\$3A3B,\$40FF,\$DF00	
	97E8	000A					
	97EA	3A3B					
	97EC	40FF					
	97EE	DF00					
16019				*			
16020				*	END OF PROBLEM #18		
16021				*			
16022				*****			
16023				*			
16024				*	PROBLEM 19 - PATCH #24		


```
16025 *  
16026 * CORRECT PROM VERSION DISPLAY  
16027 *  
16028 97F0 DC72 PATCH24 MOVB *R2+,*R1+ STORE FIRST DIGIT  
16029 97F2 0502 INC R2 SKIP TO NEXT DIGIT  
16030 97F4 DC52 MOV3 *R2,*R1+ STORE SECOND DIGIT  
16031 97F6 0460 0800 B BACK24  
16032 *  
16033 * END OF PROBLEM 19  
16034 *  
16035 *  
16036 *  
16037 * PROBLEM 21 - PATCH #27  
16038 *  
16039 * DISPLAY DONE INTERRUPT IS NOT ALWAYS GENERATED WHEN P/W IS CHANGED  
16040 *  
16041 97FA 0300 0007 PATCH27 LIM1 7  
16042 97FE 4802 D9C8 SZC R2,DSPRLT SET P/W BITS LOW  
16043 9802 4802 D9CE SZC R2,DSPWFM  
16044 9806 4802 D9CC SZC R2,DSPCRS  
16045 980A E803 D9C8 SOC R3,DSPRLT SET NEW P/W  
16046 980E E803 D9CE SOC R3,DSPWFM  
16047 9812 E803 D9CC SOC R3,DSPCRS  
16048 9816 0300 000F LIM1 $0F ALLOW ALL INTERRUPTS  
16049 981A 0460 6378 B BACK27  
16050 *  
16051 * END OF PROBLEM 21  
16052 *  
16053 *  
16054 *
```

SYM30L TABLE

ABORTQ	2694	ABSCON	4EFC	ACPTCHJ	093A	ACQERR	1988	ACQINT	3386
ACQNJLL	1878	ACQWFM1	09D2	ACQWFM	09D0	ACTGPI3	00A0	ADDCR	2CF0
ADDFP2	4AEC	ADDLBL	2000	ADDOP	48BE	ADDRMDE	3338	ADDSIZE	78AC
ADDSPCS	7886	ADDSUB	6F94	ADDVZR	886E	ADDZADJ	6F8C	ADITION	6F80
ADJEXP	717A	ADJSTR	13D6	ADRWFM	6984	ADVLIN	12DE	ADXD	ED12
ADYO	E014	AEQB	7578	AGTB	7572	ALLDON2	6798	ALLDON	6786
ALTB	757C	ANDWRD	E00A	AMNTVES	7566	AMREG	D994	ANDBJZZ	6CF8
APTE	3378	APT	337E	AQRFILL	18C8	AQRFULL	1D44	AQRMORE	1CFE
AQRSHPS	1CE4	AQRSWP	1CEA	AQRWFM	1CC2	AQR1WFM	1CD8	AQR1.8	1D04
AQR2WFM	1CD4	AQR.GND	1CCA	AQSDJNE	1F16	AQSERR	1A9E	AQSSHFT	1F04
AQSWFM	1EBA	AQS1WFM	1EDC	AQS2WFM	1ED0	ARGDONE	4D7A	ASCADR	2E68
ASCII0	3797	ASCII1	3795	ASCII2	3795	ASCII3	3794	ASCII4	3793
ASCII5	3792	ASCII6	3791	ASCII7	3790	ASCII8	379F	ASCII9	379E
ASCII	36EC	ASCLF	2EA8	ASCLJ	2E80	ASCOFF	2EB0	ASCOFL	2EA0
ASCON	2EAC	ASCRO	231E	ASCRIS	2EB4	ASCTL	2E9D	ASCTJ	2E7D
ASC	3329	ASNRMLZ	6FDE	ASTERISK	34DE	ASTFILL	76D2	ASTRTN	76DC
AT	3324	AUDIO	0D13	AUTOORJ	5D32	AVGERR	195D	AVGSTOP	1980
AVGWFM	1DD0	AVG1WFM	1DD8	AWORD	E002	AWRD	E002	BACKJ?1	D908
BACKUP2	D90A	BACKUP	48DE	BACK11	2C3E	BACK13	2AD2	BACK14	2B40
BACK15	2C92	BACK16	276E	BACK1	2498	BACK20	3246	BACK24	08D0
BACK27	6378	BACK2	24EE	BACK3	2A08	BACK4	2A8E	BACK5	03A6
BACK7	1D84	BADGKEY	25EC	BADHSCL	4B5A	BADLBL	2896	BADPNT	3390
BADPOP1	5486	BADPOP2	5490	BADPOP3	5496	BADVAL	2888	BAD15	5A3D
BAD17	59EA	BAD18	585D	BAD6	0B9C	BAD7	519A	BAKCRJS	6A28
BATSET1	085D	BATSET2	0864	BD	370C	BEEP	030E	BEGNSCRN	4318
BELLB	3326	BELLIN	2AC8	BIGLOP	5854	BI	3378	BLANKST	368A
BLANK	34E0	BLKFLI	15C6	BLNKJON	435D	BLNKFLG	0AC6	BLNKJVR	36C8
BLNK	4342	BO	3386	BP	E210	BUFAVL	0964	BUFCHK1	0074
BUFIN	03DC	BUFRCHK	01CE	BUFRIT	338A	BUFLEN	33E6	BUSYLED	33C4
BUSYUP	1718	BUSY1	3666	BUSY	0811	BUTTONHIT	0336	BUZZIT	6CF4
BWRD	E004	BWRD	E004	BYTEIN	248A	BYTEOUT	2474	CALCJ?	145A
CALCWFM	4FFE	CARROT	34D0	CD	378D	CERENBL	32AA	CHARSTART	33F4
CHE001	3392	CHE002	3394	CHE014	3396	CHE008	3398	CHE011	339A
CHE022	339C	CHE044	339E	CHE088	33A0	CHFFFF	3336	CHFF	3336
CHGSCL	4C8A	CHKBUF	78A4	CHKCKEY	5082	CHKDIV	1EAA	CHKD.M	3112
CHKDMAX	574C	CHKDMIN	5746	CHKDJNE	4E94	CHKEOIT	0EC2	CHKEJI	3206
CHKEXE	4178	CHKEXT	1AB0	CHKFACT	482A	CHKFDLM	284E	CHKFFFF	715E
CHKFLG	4CAC	CHKFPRG	49E6	CHKFST	413A	CHKIMAX	550C	CHKIMIN	54FC
CHKLBL	287D	CHKLO	0EAC	CHKMODE	1444	CHKNXTM	18FC	CHKNXT	444A
CHKOPNT	5CC4	CHKOP	489A	CHKPROG	0436	CHKRATE	1A74	CHKRAT	1A7A
CHKRO	1FE2	CHKSF	2048	CHKSIGN	660C	CHKSLEW	4822	CHKSTOP	1976
CHKSTP	00A6	CHKTEOI	2A72	CHKTDN	0190	CHKVKEY	5124	CHKVSCL	4880
CHKWFM	18A4	CHKWRN	0308	CHK.PNT	213D	CHSCHK	65C0	CHSCYS	64D4
CHSKEY	382A	CHSWFM	64DC	CH1000	3380	CH100	3378	CH10	3358
CH2000	3382	CH200	337A	CH20	3362	CH3FFF	338C	CH3000	3384
CH30	336A	CH4000	3386	CH400	337C	CH40	336E	CH5000	3388
CH6000	338A	CH7FFF	338E	CH8000	3390	CH800	337E	CH80	3376
CH8	3348	CKDECP	120A	CLER4FM	078A	CLINE15	3340	CLINE16	3348
CLINE1	333A	CLINE2	333C	CLKFAIL	06E2	CLKFLG	0F94	CLKGOOD	06EC
CLKINT	337A	CLKRST	338C	CLKWAIT	094A	CLLKEY	37DE	CLOCK	0E26
CLPFLAG	DAAC	CLPKEY	37E2	CLRDSP	63C0	CLRLAST	48C8	CLRLIN	48AC
CLRPTR	48D6	CLRSTK	63D4	CLRST	179C	CLRTXT	128E	CLWERR	0CE4
CLWRO	0CF0	CLWRN	0CDE	CMDBJF	D90C	CMODLM	251E	CMOINDX	D918
CMOIN	255D	CMDSRCH	2576	CMDSZ	008A	CMD	337C	CMPEND	4C78
CMPGT	4C74	CMPLOP	4C64	CMPOYAX	5CFA	CMPOMIN	5D16	CMPRCMD	2586
CMPRQUE	25CE	CMPWFM	4C34	CNSWFM	091C	CNTCROS	6A12	CNTWFMS	198E
CNTWFM	1F7A	CNVBUF	DA8A	CNVOVER	76AC	CNVWRD	1174	CNVZERO	769E
CN16	333D	CN1	3336	CN2	3334	CN4	3332	CJLOU?	04E2

SYMBOL TABLE

COLONB	3323	COMMAB	3321	COMMAND	24D2	COMOUT	299C	CONCOM	6218
CONJSP	2E44	CONSTR	DA4A	CONST2	4ED8	COV	3408	CRB	331E
CRERR0	2D3E	CRERR1	2D40	CRIN	2AA4	CRLFOUT	2B2A	CRLF	33B1
CROSFLAG	DABE	CROSFLG	DA1C	CROS_OP	6A74	CROSNUM	DAC0	CRSKEY	5FBA
CRSDPWJ	D960	CRSRO	608E	CRSTEP	5FE8	CRSWRN	608A	CRS1KEY	5F92
CRS1R	6008	CRS1	5FF8	CRS2KEY	5FAC	CRS2R	6034	CRS2	601C
CR	33AF	CTRLIN	2A84	CTRLJUT	2AFE	CURSORS	1094	CURSJR	D954
CURSPLT	D950	CURS1	D956	CURS2	D958	CURVE	2FF2	CHERR	1FD6
CYCFLAG	DA14	CYCLEUP	04EA	CYCLE	DA12	CYCTEST	0968	CYCMAT	0970
C0	3338	C1UP	5EB0	C100	3374	C1024	337C	C10	334C
C11	334E	C128	3376	C12	3350	C13	3352	C14	3354
C15	3356	C16	3358	C17	335A	C18	335C	C1	333A
C2TOOBIG	5F7C	C2UP	5EE0	C20	335E	C256	3378	C2	333C
C31	3360	C32	3362	C33	3354	C3	333E	C40	3366
C41	3368	C48	336A	C4	3340	C50	336C	C512	337A
C5	3342	C64	336E	C6	3344	C7	3346	C8	3348
C98	3370	C99	3372	C9	334A	DACR	3380	DADJST	1358
DAL	3386	DATADRS	6C08	DATAL	29B6	DATA2	29B8	DAT	3382
DCAS	337A	DCLSTOP	23E5	DCMPRSS	134A	DCNST	340C	DD	378E
DECCHK	6584	DECKEY	3828	DECMNU	33EB	DECPNT	372F	DELIN	1332
DELTA00	6C8C	DESTROY	4916	DHCR	368A	DIAGINT	D904	DIAG	E500
DIFFOUT	57E8	DIFFPNT	57DE	DIFF4FM	5724	DIGIT	7776	DINTFP	7582
DISHM	DADE	DISPLAY	0FBC	DISP_A	3354	DIVBY0	711C	DIVFP2	4AE0
DIVZDIV	7058	DLMTBL1	97E4	DLMT3L	331C	DLMVAL	976C	DLYCRS	6048
DLYMODE	3340	DLYVPOS	6004	DMDWRD	E000	DMINMAX	573C	DONCHK	6866
DONE3	695A	DPA0D	5188	DROPLIN2	4C82	DROPLIN	4AAE	DSPBUF	D97E
DSPCRSI	33A8	DSPCRS	D9CC	DSPFAIL	D728	DSPINIT	0FA4	DSPINT	3382
DSPLNO	2DE0	DSPMAX	3348	DSPRLTI	33A4	DSPRLT	D9C8	DSPRJI	33AA
DSPRO	D9CA	DSPRST	3382	DSPSTP	3386	DSPSTR	3384	DSPTKO	2DDA
DSPWFMI	33A6	DSPWFM	D9CE	DSWERR	0C96	DSWRO	0CA2	DSHWARN	0C8C
DSW	35C2	DVCR	35FA	DXADR	E012	DYADR	E014	EDITMODE	43E4
EDITPNT	3AAA	EEXCHK	657C	EEXHIT	57E0	EEXKEY	3826	EEXMNU	33EA
EEXMSG	6580	EEXONE	65AE	ENABLE	D936	ENDAQS	1C06	ENDASC	370C
ENDFLG	33F2	ENDGND	1C4C	ENDKEYB	382D	ENDKEY	382C	ENDLINE	33E4
END4NJ	455E	ENDMPY	4AF5	ENJOJT	1744	ENDPNT	3802	ENDPRGM	2D32
ENDP11	968E	ENDP18	972E	ENDP19	975A	ENDP1	9618	ENDP20	9786
ENDP21	97C4	ENDP2	9638	ENDP3	9654	ENDP4	9670	ENDP6	968C
ENDP7	96A2	ENDQTB	3015	ENDSRV	4448	ENDSRQ	2F2A	ENDSTAT	364A
ENDTAB	3C36	ENDTEXT	D9F4	ENDVALS	5A80	END	337A	EOILF	3382
EOIONLY	3152	EOIOUT	31F2	EOPFLG	DAC2	EOVER	7702	EQFP2	4AD2
ERRLED	33C2	ERRLSTN	383C	ERRJR	0012	ERROUT	3184	ERRPNT	624E
ERRPOP	549A	ERRSAVE	2D3C	ERRSTEP	5C72	ERRTALK	308E	ERRUP	1728
ERR00	2F4E	ERR01	2F54	ERR2JR	5D34	ERR2PNT	62D4	ERR2SCL	0800
ERR	3674	ERR.0RJ	5C32	ESCB	3325	ESIGNS	34E6	ESMALL	7496
ETXOUT	281E	ETX	33AD	EVS	7434	EXPAGTB	6FFC	EXPAND	13C0
EXPBGTA	7014	EXPEQL	755E	EXPPF	4F38	EXPNGTV	75D6	EXPNJH1	66DA
EXPNUM	7848	EXPOK	674E	EXPOS	5730	EXPOUT	781C	EXPOV	74A6
EXPWFM	5022	EXP11	D9A2	EXP12	D9A4	EXP21	D9A6	EXP22	D9A8
EXREXBL	32A8	E100FLL	75F2	E1	740E	E2	7418	E	34FC
E.100	76EA	FALFLG	DAC4	FALSES	36AC	FATAL	D94A	FE01	3382
FFEOI	01C4	FILLCHK	219C	FILLPNT	5BF8	FILLWFM	1C16	FILL	6C74
FINDBAD	5C3A	FINDEXSGN	67D6	FINEXCHS	55E4	FINISH	588E	FIRSTF	71F6
FIRST0	720C	FKEY	D952	FLICKER	DA16	FNDCHD	260A	FNDFRST	6C2C
FNDLAST	6C5C	FNDNEXT	6C44	FNDQJE	264C	FND10	762A	FND1	766C
FORMAT	7708	FPACQ	233E	FPADJZ	5F7A	FPADD	6F84	FPBTTN	0E00
FPCMPR	7524	FPDIVZ	7046	FPDIV	7058	FPDON	4AF8	FPFRST	4CF6
FPISIN	27FE	FPKEYON	0F3C	FPKEY	0F26	FPKY	0040	FPMPYZ	70A0
FPMPY	70AA	FPNEXP	7382	FPNLOG	7298	FPSCND	4D04	FPSQRT	74BA
FPSUBZ	6F56	FPSUB	6F60	FPTMP	D9C2	FPTRNC	75A6	FPWAIT	0E12

SYMBOL TABLE

FP1E	333A	FP1M	3404	FP10E	3340	FP10M	33FE	FP1000E	3402
FP10004	3400	FP12E	3340	FP124	33F6	FP2ASC	75E0	FP2ELE	6D1E
FP2INT	75A0	FP2WFM	6B68	FP20E	33FA	FP20M	33F8	FP6E	333E
FP6M	33FC	FP	E010	FRCTOUT	77F4	FRDOUT	1430	FREE	1202
FRMTOUT	2B0C	FRONT	E010	FRSTCLK	06C4	FRSTCRV	2784	FRSTDST	27AA
FRSTDIGT	54E6	FRSTPNT	29CA	FRSTPRE	2760	FRSTPT	1E86	FRSTRES	337A
FSTIPNT	55EE	FSTPT	6B4E	FSTS4FT	56DE	FUZZCHK	6B54	FWDGRJS	6A1C
FWDNXT	40A2	FWDSTOP	40B8	GETEEX	57D2	GETENDS	5A62	GETFRST	6A6A
GETLINE	4534	GETMID	58F2	GETPER	5B42	GETPNT	45AA	GETRJN	245C
GET	3382	GEXEKEY	37EA	GFORMAT	D920	GHRZSF	19F6	GHTIME	59EE
GNDAVE	1D6E	GNDPOS	1D90	GNDSTOP	1D48	GNDWFM	1C52	GND03AS	33E0
GND1BAS	33E2	GND1WFM	1C7C	GND2WFM	1C68	GOLD	D94E	GOODKEY	020C
GOODVZR	29AA	GOOD	6184	GOSUBLEN	33E8	GOSUBUF	DA4E	GOTOERR	46C2
GOTOK	46A2	GOTRATJ	5228	GOTS3N	6552	GPIBADR	D91A	GPIB3UF	33E0
GPIBEOI	338A	GPIBINT	337E	GPIBIN	3124	GPIBKEY	D944	GPIBKY	0044
GPIBOK	24C4	GPIBOPT	D942	GPIBOUT	3154	GPIBR12	D92C	GPIB	234E
GPNTIN	29C4	GPRGKEY	37FE	GRJUNDO	DA2A	GROUND1	DA2C	GTLTMSK	3438
HALTIT	D9C0	HAVWFM	4B18	HCRPOS	58E0	HCRSPOP	5F0A	HCRSR	10EA
HCRSU	1634	HCRS2ON	5F5E	HCRS	DA00	HCRXY	15EA	HCRYT	1624
HCR1	68F0	HCR2	68F6	HCR3	68FE	HCR	35EA	HDLA	337C
HEXDSY	E018	HEXP	3340	HITCRS	5AAC	HITLNN	4720	HITLVL	5ECA
HITPNT	45F8	HITRTN	4624	HITSPOT	4694	HITSYM3	57A8	HKILPNT	532E
HLOPING	533C	HLTKEY	03D0	HOLDLIN2	4C86	HOLDLIN	4AA6	HOLD12	DA38
HOLD7	DA34	HOLD8	DA36	HORDLTA	59F2	HORZALT	3344	HORZA	333E
HORZB	3352	HORZCHP	3354	HORZM	DA28	HORZSF	1A3E	HORZXY	14B8
HORZYT	14EA	HORZ	358C	HOWBIG	66FA	HPERR	569C	HPSHFT	5678
HPD	5654	HP1	5658	HP2	5670	HRZSCL	6844	HSCALD	3350
HSCLU	14FA	HVFLAG	D99E	HWDEVBL	08A0	HXPOON	5358	HXPLT1	535A
H.RO	2302	IACQ	0000	IBP	0002	ICURVE	2980	IDENTFY	1266
IDENTIFY	37B0	IDMSG	2EFE	IDSY	000A	IENCOG	289C	IFXBACK	4804
IGNORE	D93E	IGPIB	0008	IKB	000C	IKRON	0004	INCURVE	298A
INERR	4268	INITCLR	12CA	INITKT	12B2	INLABEL	2838	INNUM	66D6
INOUT	D940	INPUTDON	42F6	INPUTIT	4270	INPUTMOD	429C	INR.PT	28BE
INSERT	13F6	INSTR	3210	INSWFM	18EE	INTDSP	0FCA	INTFIX	771A
INTFLAG	D938	INTGPIB1	D932	INTGPI3	D930	INTGPNT	55C8	INTGWFM	5480
INTMASK	33D0	INTOUT	77E0	INTSTST	75C2	INTU1	2216	INTU	2210
INT2FP	757E	INT7854	D906	INJSE	D976	INVAL	3234	INVS2R2	340A
IOGENBL	32A4	IOLED	33C6	IO	0814	IPT.FMT	28D4	IRTC	0009
ISDECPT	123C	ISDIGIT	2102	ISDLN	3122	ISOMAX	5D0A	ISOMIN	5D26
ISOPNT	5D82	ISPOS	27DA	ISSF	20A4	ISSTOP	9690	IS2K	0586
IS4K	05BC	IS8K	05C2	ITERM	0003	ITIMER	0005	ITRP	542A
IWFMPRE	282A	IXINCR	28FC	IXUNIT	2914	IXYUNIT	291A	IXZERJ	28EA
IYMULT	2940	IYUNIT	2978	IYZERO	2960	I1	0001	I2FP	6F0C
JUMP	37A0	KBCODE	E00E	KBID	0010	KBINT	337C	KBRST	0016
KB	E00E	KCUART	E314	KDUART	E214	KEEPON	6670	KEEP2QS	241A
KEYA3S	4E5C	KEYAQR	1904	KEYAQS	195E	KEYAREA	5AD4	KEYAVG1000	191C
KEYAVG100	1916	KEYAVG10	1910	KEYAVG	1922	KEYBOTH	0C02	KEYBUFR	DA3A
KEYB	D967	KEYCEROFF	26C8	KEYCERON	26D0	KEYCHS	649E	KEYCLD	0CFE
KEYCLF	49D4	KEYCLL	4886	KEYCLP	483C	KEYCLR	6432	KEYCLW	0CAB
KEYCLX	544C	KEYCNS	61E0	KEYCRS1LFT	5FB6	KEYCRS1RGT	5F8C	KEYCRS1	5D08
KEYCRS2LFT	5FA0	KEYCRS2RGT	5FA6	KEYCRS2.1	5D30	KEYCYCLE	099E	KEYDEC	649E
KEYDELAY	5A3A	KEYDIFF	5714	KEYDIV	4A4A	KEYDOTS	0D72	KEYDSH	0C3C
KEYEEX	549E	KEYENERGY	5A3A	KEYENTER	548A	KEYEXP	4F0A	KEYEXROFF	26D8
KEYEXRON	26E0	KEYFALL	595C	KEYFREJ	5B18	KEYF	49C2	KEYGEKE	4000
KEYGND	196C	KEYGNEXT	4786	KEYGOTO	464A	KEYGPROG	484C	KEYGSB	4646
KEYHCRJ	53F8	KEYHMDALT	0DDA	KEYH4DA	0D2E	KEYHMDB	0DF2	KEYHMDCHOP	0DE6
KEYHMD	0DFC	KEYHPLFT	5642	KEYHPRGT	5646	KEYHSCL	0ABB	KEYHXPD	519E
KEYIJ	2D42	KEYIFXEQY	4A60	KEYIFY3TK	4A68	KEYINTG	54A0	KEYI0COFF	26F8
KEYI0CON	2700	KEYITRP	53B8	KEYL3L	46D2	KEYLNN	4874	KEYLN	4F10

SYMBOL TABLE

KEYMATH	4A6E	KEYMAX	5826	KEYMEAN	583C	KEYMID	5830	KEYMINUS	4A58
KEYMIN	582A	KEYMNU	363E	KEYMJLT	4A44	KEYNEXT	4786	KEYNJP	4904
KEYNXT2	472E	KEYOFF	5DEC	KEYOPCJFF	2688	KEYOPCON	26C0	KEYORD	58A6
KEYOUT	178E	KEYPAUSE	4938	KEYPER	5B12	KEYPLUS	4A50	KEYPNT	623C
KEYPREV	478A	KEYPROG	4054	KEYP2P	5836	KEYP.W	641C	KEYRODOUT	1F18
KEYREADX	2708	KEYREMOFF	26E8	KEYREMOJN	26F0	KEYRISE	596A	KEYRMS	58C4
KEYRJLL	6458	KEYRQSOFF	26A8	KEYRQSON	26B0	KEYRQS	2F6A	KEYRTN	4682
KEYRUN	4970	KEYSAVE	2C8E	KEYSCOPE	0BA2	KEYSENDX	2B3C	KEYSGN	4E60
KEYSMOJTH	5072	KEYSQRT	4F16	KEYSTART	498C	KEYSTAT	16F2	KEYSTEP	476A
KEYSTORED	08C8	KEYSMH	2F62	KEYSAL	2F5A	KEYTAB	3836	KEYTEXT	2ACE
KEYTIME	0062	KEYTRE	004C	KEYUP	17A6	KEYVCRD	5E16	KEYVECT	0D80
KEYVMDADD	0DA6	KEYVMDALT	0D9A	KEYVMDCHOP	0DB2	KEYVMDL	0D8E	KEYVMDR	0DBE
KEYVMD	0DC8	KEYVPD4	6096	KEYVPU3	50A0	KEYVSC	0AB6	KEYVS	0D2A
KEYVXPJ	098A	KEYVZR	0AAA	KEYWFM	6162	KEYWIDTH	5A1A	KEYXFR2	0948
KEYXFR	0946	KEYX2Y	646E	KEY0	649E	KEY1	649E	KEY2CNS	61F6
KEY2HCRD	5EEC	KEY2HSCL	0AC8	KEY2JRD	5C44	KEY2PNT	626E	KEY2P.W	62E4
KEY2TEXT	2A1C	KEY2VCRD	5E2E	KEY2VSCL	0AD4	KEY2VZR	0B06	KEY2WFM	6194
KEY2	649E	KEY3	649E	KEY4	649E	KEY5	649E	KEY6	649E
KEY7	649E	KEY8	649E	KEY9	649E	KEY	0966	KEY.	649E
KNTOSWS	0C56	KRAM	0A00	KYBDEX	4008	KYBRST	33B8	LABELIN	2858
LACS	337C	LASTKEY	0968	LASTXJT	4A1E	LASTOUT	2484	LBLCHR	283C
LCLFP	0F6C	LCLKEY	0384	LEDCHK	0EF8	LEFTB	33F8	LEFTJST	78C6
LENOK	139A	LFB	331F	LF	33AE	LINEHUNT	45C4	LINLEN	342A
LINENUM	DAA4	LINEOK	12F4	LINEPNT	DAA6	LINSTR	09D4	LINE15	156A
LINE16	D9F2	LINE2	1508	LINE99	DAAE	LISTENER	338A	LITERR	6CE8
LNC1	3426	LNC2	3428	LNEWMAX	4FB4	LNEWMIN	4FB8	LNEXTPT	4FBA
LNLOOP	4A2A	LNNOIGT	4218	LNNERR	4880	LNNFLAG	0AB2	LNNKEYB	3805
LNNKEY	3804	LNNUMB	4230	LNKERR1	7310	LNKERR	730A	LN10E	333C
LN104	3410	LN2EXP	3338	LN2MANT	3424	LN2	3426	LOADA	782A
LOAD3	7838	LOADSF	2178	LOGEE	3336	LOGEM	340E	LOGPK1	72F2
LOGPK2	7396	LOG0.2E	3338	LOG0.24	3422	LOG0.5E	3336	LOG0.5H	3420
LOG10	7392	LOG13	7374	LOG1	72B8	LOG2	72E2	LOG4	7352
LOG5	7364	LOG6	739E	LOG7	7390	LOG8	73AC	LOG9	738A
L0LSTN	307E	LOOK4LOW	6AA6	LOPSCALX	4D9C	LOTO	0ECC	LO	3382
LSHFTOK	7264	LT1NOCRS	5374	MADJSGN	70EE	MANTEXP	668E	MANTOJT	775E
MAPMEM	05C6	MAPSIZE	0016	MAP2	348A	MAP4	34A0	MAP8	3486
MARGAOK	70CE	MARGBOK	70E0	MATC4	E210	MATHARG	4CCE	MAXART1	09AE
MAXART2	0980	MAXCNS	0ADC	MAXKRAM	0088	MAXLINE	0A2E	MAXPJS	6748
MAXPROG	DAE2	MAXRATE	3414	MAXRAT4	3412	MAX80K	7934	MAX	588A
MA	3390	MDWRD	E000	MEANRMS	58CA	MF.RO	2292	MF.WFM	2236
MID	5834	MINART1	09AA	MINART2	09AC	MINUS	3782	MIN80K	793C
MIN	5880	MM2PNT	62AE	MNRMLZ	70FA	MNTSHFT	7698	MNUA3S	3E6E
MNUAQR	3C6C	MNUAQS	3E8E	MNUAREA	3CE4	MNUAVG1800	3CCA	MNUAVG100	3C3E
MNUAVG10	3C38	MNUAVG	3C68	MNUBOT4	3E12	MNUCEROFF	3EC0	MNUCERON	3EBA
MNUCHS	3CC4	MNUCLD	3EAA	MNUCLF	3F2A	MNUCLL	3D9E	MNUCLP	3F12
MNUCLR	3E84	MNUCLM	3D4A	MNUCLX	3D02	MNUCNS	3D1C	MNUCRS1LFT	3CB4
MNUCRS1RGT	3CBA	MNUCRS1	3F34	MNUCRS2LFT	3DE4	MNUCRS2RGT	3DDE	MNUCRS2.1	3DEE
MNUDELAY	3CA8	MNUDIFF	3EAE	MNUDIV	3CC8	MNUDOTS	3EAD	MNUDSH	3D4E
MNUJEX	3DC2	MNUENERGY	3C02	MNUENTER	3C98	MNUEXECUTE	3C7E	MNUEXP	3E1E
MNUEXROFF	3ECE	MNUEXRON	3EC8	MNUFALL	3E22	MNUFREQ	3E72	MNUF	3C46
MNUGND	3C64	MNUGOTO	3DB2	MNUGPIB	0934	MNUGSB	3DAE	MNUHCRD	3CEE
MNUHMDALT	3E52	MNUHMDA	3E4C	MNUHMD8	3E62	MNUHMDCHOP	3E5A	MNUHPLFT	3D72
MNUHPRGT	3078	MNUHSCL	3D94	MNUHXPJ	3D7E	MNUID	3C8E	MNUIFXEQY	3F24
MNUIFYGTX	308C	MNUINT3	3D5A	MNUICJFF	3EF8	MNUIOCON	3EF2	MNUITRP	3D64
MNUJBL	3088	MNULNN	3F20	MNULN	3C9E	MNUMAX	3D32	MNUMEAN	3D0A
MNUMID	3D10	MNUMINUS	3D58	MNUMIN	3D36	MNUMULT	3D00	MNUMEXT	3DA8
MNUMOP	3C62	MNUOFF	3DEA	MNUOPCJFF	3EEA	MNUOPCON	3EE4	MNUORD	3D60
MNUPAUSE	3C50	MNUPER	3CEA	MNUP.US	3D2A	MNUPNT	3D18	MNUPREV	3DA2
MNUPROG	3C86	MNUP2P	3D3A	MNJP.W	3D14	MNURDOJT	3E08	MNUREADX	3DC6

SYMBOL TABLE

MNURMOFF	3E0C	MNURMON	3E06	MNURISE	3CA2	MNURMS	3006	MNUROLL	3E68
MNURQSOFF	3F06	MNURQSON	3F00	MNURIS	3C4C	MNURTN	3F16	MNURUN	3C48
MNUSAVE	3E02	MNUSCOPE	3C78	MNUSENDX	3F2E	MNUSGN	3C08	MNUSMOOTH	306A
MNUSQRT	302C	MNUSTART	3F1A	MNUSTEP	3C5C	MNUSTOP	3C56	MNUSTORED	3C70
MNUSWH	309A	MNUSWL	3F0E	MNUTABL	3C38	MNUTEXT	3C92	MNUTIME	3044
MNUVCRD	3CF4	MNUVECT	3D3E	MNUVMDADJ	3E36	MNUVMDALT	3E2E	MNUVMDCHOP	3E3E
MNUVMDL	3E28	MNUVMDR	3E46	MNUVPON	3DF6	MNUVPUP	3DFC	MNUVSCL	3D8A
MNUVS	3EA6	MNUVXPJ	3D84	MNUVZR	3D30	MNUVFM	3D20	MNUWJTH	3CAE
MNUX2Y	3CD2	MNU0	3CC0	MNUJ1	3CFA	MNU2GNS	3E94	MNU2ICRD	3E78
MNU2HSCL	3D08	MNU2ORD	3EB4	MNU2PNT	3E8E	MNU2P.W	3E88	MNU2TEXT	3E18
MNU2VCRD	3E7E	MNU2VSCL	3DC0	MNU2VZR	3D02	MNU2WFM	3E9A	MNU2	3CFC
MNU3	3CFE	MNU4	3D24	MNUJ5	3D26	MNU6	3D28	MNU7	3D52
MNU8	3D54	MNU9	3D56	MNJ.	3CC2	MODEEDIT	4282	MODESW	37BA
MODE	E000	MORNUM	6518	MORTXT	4474	MORVAL	9772	NOVBACK	458E
MOVCRS	5FF2	MOVDATA	6104	MOVFLAG	0AB0	MOVIT	481E	MOVPT	5700
MOVFLW	710A	MOVSUB	4566	MOVVPOS	60D0	MPVBY0	712A	MPYDIVO	712C
MPYFP2	4AE6	MPYLOP	4C06	MPYNOMAX	4DBA	MPYNOMIN	4DD4	MPYZMPY	7082
MPY2	4C0A	MSA	337E	MSIGNS	34E2	MWACQR	3382	MWDCLC	337E
MWDRLT	3380	MWEXCLK	3380	MWVCTR	3348	MWXVSY	3362	MW2WFM	337E
MWBKHZ	336E	NCHG2	0286	NEATFCT	7A18	NEGEXP	34F2	NEGFLG	0984
NEGKEY	3830	NEGMNU	33EC	NEWKPE	098E	NEWEXP4	098C	NEWGHT	1E50
NEWHSCCL	6EA8	NEWIMAX	5516	NEWIMIN	5506	NEWOMAX	5C8C	NEWOMIN	5C80
NEWVSCL	6EE0	NEWVZR	7AC0	NEXTCHAR	43FA	NEXTCRS	10A4	NEXTC2	44C8
NEXTDONE	485C	NEXTIN	2A50	NEXTLINE	4380	NEXTMNU	2C8C	NEXTPRV	4866
NEXTQUIT	4844	NEXTRUN	479E	NEXTSF	204E	NEXTWFM	18B8	NOAVERO	1E44
NOBAD	528E	NOBATERY	0540	NOBEEP	0312	NOBUFR	03EE	NOCHG	6CE6
NOCLP	415A	NOCROS	6A94	NOCRS	5272	NOCRT	43C0	NOCYC	0988
NODECPT	1238	NODIV	5808	NOFKEY	494C	NOFPKEY	340A	NOFWD5	593C
NOGCHD	088A	NOGOOD2	0A9C	NOGOOD3	5306	NOGOOD4	5176	NOGOOD5	6238
NOGOOD6	5EE6	NOGOOD7	5064	NOGPIBK	242C	NOHCRS	5F04	NOITRP	5470
NOKEY	180C	NOLABL	4506	NOLINE	45F0	NOLSTR	3164	NONSPCL	7322
NONTR	02F4	NONUMCHG	0432	NONUMLIN	43D4	NONXT	47F0	NOPHER	5E10
NOPKEY	388A	NOPL	16EA	NOPRGNTR	8242	NOPURG	4156	NOQUER	2456
NOQUE	26A6	NORMAL	32D2	NORQS	3310	NORSV	3308	NORUN	498A
NOSET	0334	NOSIG	6698	NOSLAVE	1F08	NOSPEC	41F6	NOSRA	0B40
NOSTPWN	00CA	NOSTST	1776	NOSY4B	67B6	NOTAQRD	1E8E	NOTAQS	1ACC
NOTC4D	2392	NOTCTRL	12AA	NOTDGT	279C	NOTDIGIT	2112	NOTEND	24AE
NOTE	5D04	NOTFAL	599E	NOTFILL	21C0	NOTFST	41AC	NOTFJLL	27F2
NOTGP1B	3182	NOTIOL	03C0	NOTLBL	288A	NOTLNN	42DA	NOTOPNT	5074
NOTQ	2538	NOTRES	2A0C	NOTR4LS	31A0	NOTRML	3106	NOTRJP	2118
NOTSF	209E	NOTSUS	01E8	NOTTRM	2566	NOTXTNUM	4544	NOTYSF	217C
NOUNIT	2938	NOVRFLW	71CC	NOVZRH	1896	NOWARN1	047E	NOWARN	467E
NOWIPE	411C	NOXNEG	66D0	NO.ACQ	0FE4	NO.INTR	1002	NRMCHKZ	71DE
NRMCHK	7134	NRMLA	7074	NR4LIZZ	71E4	NRMLIZ	713C	NRMLZED	7188
NRMLZD	715A	NRMLZ	7186	NR2	332C	NR3	332E	NTRKEY	3834
NT8000	7600	NULLFIL	1310	NULLHEAD	343A	NULLIN	12FA	NULLJUT	2824
NULLWFM	6F26	NULL	33AC	NULWRDB	342F	NULWRD	342E	NUNCHG	0280
NUMDSP	49F0	NUMMNU	342C	NUMRETRN	660A	NUMTAB	3812	NUMWMS	2256
NUM3	0A9E	NXTCK	665E	NXTC4D	2598	NXTCRV	277A	NXTCYCL	181C
NXTDGT	27DE	NXTERR	4856	NXTGND	1062	NXTHUNT	4360	NXTIPNT	551A
NXTKEYB	37EF	NXTKEYG	3806	NXTKEY	37EE	NXTLBL	2868	NXTLIN	2E0C
NXTLOP	4832	NXTOK	47AE	NXTOPNT	5C84	NXTPASS	1E0C	NXTPNT	1E94
NXTPRE	2756	NXTQUE	25D8	NXTR3YT	2738	NXTSF	2192	NXTSTR	3216
NXTTEXT	2AEE	NXTVAL	323A	NXTWFM	153A	NXTWPNT	2C52	NXTWRJ	1276
OBUSY	0AC8	OCURVE	2C44	OENCOG	288E	OFFLINE	0EEC	OFFLIN	2E00
OKEY	0ACA	OKPNT	625C	OLDTKT	131C	OMINMAX	5CAC	ONEPNT	6088
ONESIZE	0028	ONESRCH	684C	ONET3L	3522	ONLINE	3390	ONOFF	0922
ONR.PT	2BA6	OPCENBL	32AC	OPROGLN	0ACC	OPT.FMT	28B6	OPWCHG	6C98
OPWFM0	095C	OPWFMH	095E	OPWFM	095A	OPWH2WD	69A8	OPW	359E

SYMBOL TABLE

ORDLP	5BF2	ORDOK	5BE8	ORDWFM	5D60	OUTFP	3286	OUTLBL	2B72
OUTMNU	23D0	OUTPUT	0004	OUTSTR	3262	OUTVAL	3270	OVFAJJ	6FC0
OVFLW	75CC	OV	3406	OWFMPRE	2B8C	OXINCR	2BCE	OXUNIT	2BDE
OXZERO	28C2	OYMULT	2C1A	OYUNIT	2C22	OYZERO	2BF0	O	3327
PANICB	3833	PANIC	3832	PASSCHD	261A	PASSIT	665A	PATCH11	96A6
PATCH13	96C2	PATCH14	96DC	PATCH15	96F6	PATCH16	9710	PATCH18	971A
PATCH19	9732	PATCH1	9600	PATCH20	975E	PATCH21	978A	PATCH24	97F0
PATCH27	97FA	PATCH2	961C	PATCH3	363C	PATCH4	9658	PATCH6	9674
PATCH7	969A	PAUSE	3658	PAUSKEYB	3811	PAUSKEY	3810	PCDSY	E21C
PERBAD	5B92	PERCENT	3434	PERCHK	7814	PERIOD1	34FA	PI.RJ	22A2
PI.WFM	2246	PLATEAU	6B10	PLBLNK	36D4	PLOUT	16C0	PLUS	3781
PL	36E0	PNTCHK	1E6E	PNTFILL	6BEE	PNTHUNT	45CE	PNTONE	6E5E
PNTSTWJ	6E38	PNTWFM	6BC0	PNT2FP	6BA6	POINTVAR	DA32	POINT	DA30
POPARG	4F1A	POPBUF	040A	POPHSCL	51F0	POPLOP	042A	PJPREG	690E
POPSTK	6918	POP1	53DE	POP2	53C2	POP3	5416	PJSEKHIT	65E0
POSEXP	34EA	POSKEY	382E	POSMNU	33EE	POSNOW	513E	POSRLT	514A
POWERUP	0750	PREVKEYB	37F3	PREVKEY	37F2	PREZERO	7768	PRGRTP	16BA
PRGSTAT	15B0	PRODSPRO	095C	PRODSPRT	096E	PROGCRS	5FCE	PROGJN	42FA
PROGIN	40E6	PROGKEY	37E6	PROGLN	0AA8	PROGMEN	DAE0	PROGON	04BA
PROGRAM	0AA0	PROGRS	09BA	PROGSTP	0AB4	PROGUP	144E	PROGVP	60BC
PROMODE	0AA2	PROMV	3FF8	PRORALT	0ABC	PSHANUM	1940	PSHC0N	6204
PSHDIFF	5808	PSHFREQ	5B40	PSHIF2	5E08	PSHLOP	6944	PSHNJMS	1C3A
PSHREG	6962	PSHRVAL	2818	PSHSTK	6936	PSHWFMS	1C24	PUD	0000
PURGNXT	4120	PWRERR1	34D2	PWRERR2	34D4	PWRERR3	34D6	PWRERR4	34D8
PWRSRQ	0734	QUEERR	2F30	QUEID	2EDE	QUESRC1	25C8	QUESRQ	2F10
QUETABL	3000	QUEWAIT	2686	RAMBAD	0658	RAMBASE	A000	RAMFAIL	05AA
RAMMAP	0AD0	RAMOPT	0AE4	RAMTEST	05FC	RAMTOP	0800	RANGE10	7612
RANGE1	7654	RDIN	20AE	RDOUT	141A	RDSTST	2370	RJSY	0000
RDERR	1F6C	RDIFLAG	0972	RDTSRCH	20E4	READGNS	2798	READERR	2740
READJUT	1138	REALTIME	09C5	RECDJN	5192	REFRESH	09C4	REFRS1	0F70
REHENSBL	32A6	RENKEY	0928	RENLED	33CC	REMOTE	0005	REN	3386
RESET	3390	RESGPIB	2648	RESIN	2A10	RESOLV	0970	RESPDS	6B62
REVERSE	786A	RFDR	3386	RGPIB	0003	RLC	337E	RLOK	338A
RLTACQ	1166	RLTRO	1156	RLTU ²	1454	RNDLOOP	77A2	RNGFNJ	768A
ROACQEN	0017	ROAQR	1162	RODMA	33C0	ROMBAD	0684	ROMBASE	0000
RONFAIL	05EE	ROMSUM	0570	ROMTEST	0668	ROMTOP	4000	ROMV	3FE8
ROOMAVL	03F0	ROSTRE	2128	ROTATE	56AA	ROUND	7796	ROUPDT	0A18
RQSENBL	32A2	RQSKEY	380C	RQSNJM	0AE5	RQSOFF	3304	RQSOK	016E
RQSON	23E0	RQS	32AE	RRTC	0007	RSLTNEG	6FEC	RSTBP	000C
RSTGPIB	0100	RSTINT	100A	RSTK3	0016	RSVBUSY	3380	RSVENBL	091E
RSVOFF	32F0	RSVTBL	3294	RSV	3386	RS	3380	RTIM	0008
RTL	337C	RTN2TXT	2A7A	RTWPI	E218	RUNALL	583A	RUNKEYR	01CE
RUNKEY	380E	RUNSET	583A	RVSXKT	40BA	RVSTOP	4802	ROR	E048
RW	E050	R0	0000	R1R	E044	R1W	E064	R10	000A
R11	0008	R12	000C	R13	0000	R14	000E	R15	000F
R1	0001	R2R	E048	R2W	E068	R2	0002	R3R	E04C
R3W	E06C	R3	0003	R4R	E050	R4W	E078	R4	0004
R5R	E054	R5W	E074	R5	0005	R6R	E058	R6W	E078
R6	0006	R7R	E05C	R7W	E07C	R7	0007	R8	0008
R9	0009	SAMEKEY	0F44	SAMPNT	1E8C	SAVSTST	1B28	SCALADD	4E46
SCALDATA	4E56	SCALDIV	4E40	SCALE	78E4	SCALSUB	4E52	SCALWFM	407E
SCALWORK	40E2	SCALW1	4E00	SCALJ2	4E08	SCALW3	4E2A	SCALW4	4E2E
SCANIN	6618	SCIENG	7738	SCI	773E	SCLOIFF	5760	SCNDCLK	06DC
SCRNFLAG	0AB6	SEMIB	3322	SENDGNS	2B5A	SENDWFM	2B68	SEOI	092A
SETGLOJ	49CC	SETHOON	5342	SETH.OP	5304	SETLAST	0316	SETRES	6370
SETSTOP	1892	SETUP	2ECC	SET1J24	635C	SET128	6314	SET256	632C
SET512	6344	SFALSE	1770	SFNEXT	215A	SHFTVZR	7A8E	SHIFTKEYB	37F8
SHIFTKEY	37FA	SHIFT1	6E70	SHIFT2	6E98	SIGNDON	4ECC	SIGNLOP	4E88
SIGNOK	721E	SINCON	4EF2	SKIPKEY	25B2	SKIPSC1	4CC6	SLWSHFT	56CA

SYMBOL TABLE

SMAK	3416	SMTHCNT	516A	SMTHEND	5170	SMTHIT	510C	SMTHLOP	5124
SMTHQUT	5176	SOFTST	33DE	SOFT	0009	SORTED	18C0	SOVFLW	7ADA
SPACBYT	3433	SPACEB	3320	SPACES	3430	SPACE	3432	SPACOUT	64F6
SPASIN	2386	SPAS	337C	SPCSTR	3226	SPEAKER	33BE	SPEC.FND	41E8
SPEC.LOP	4108	SPEC.PROG	370A	SPEC.TST	41CA	SQRNRH	6E7A	SQRREST	6E1C
SQRTA	341A	SQRTB	341C	SORTC	341E	SQRTD	751A	SRCHLNN	4702
SRQLED	33CA	SRQMSG	2F2C	SRQS	3386	SRQTEND	2ECC	SRQ	0015
STADSY	0001	STARTUSER	DFA0	START	0048	STDSY	E21E	STEPKEYG	3808
STESTOP	0498	STEWARN	DACE	STICKIT	6552	STILNXT	0252	STILPLA	6B18
STILSIG	6696	STOPACQ	189E	STOPJCL	D92E	STOPDON	0146	STOPIN	3690
STOPKEY	3382	STOPLN	4A42	STOPPRJG	04A2	STOPROG	0118	STPDSY	0002
STPFLG	DABA	STPGPIB	D93C	STPNOUT	1750	STPREAD	281C	STPRJGF	DAB8
STPSAVE	2D36	STPSEND	2C7E	STPTXT	2936	STRIN	322E	STRLSTN	3016
STRQUT	2B98	STRTALK	3898	STRTCHK	6A6C	STRTCROS	6A32	STRTDSP	1058
STRTNXT	437E	STRUE	176A	STSTOUT	178E	STSTRQS	0180	STUFF	6564
SUBFP2	4ADA	SUBZSUB	6F68	SUMIPNT	54E2	SUMREST	6D8A	SUMSQR	6DE0
SUM	6D54	SUSGPI3	D992	SUSSTOP	263A	SVREG	1370	SWAP	7876
SWCHIT	6688	SWCNT	E00A	SWEETS	0932	SWEEP	21DA	SWPCNT	E00A
SYMTAB	3F3A	SYS2K	34CC	SYS4K	342E	SYS8K	3400	S5	3380
TABEND	3F6A	TACS	337E	TALKER	3386	TCUART	E312	TDUART	E212
TEKASCR3	232E	TEKASC	3710	TEKCODE	0F00	TEKRO	230E	TENCTR	3572
TENNRH	6774	TENSIZ	0054	TENT3L	354A	TERMINAL	0008	TERMINATE	3196
TESTCLK	0688	TESTDSP	06F6	TESTRAM	1544	TESTROM	05D6	TESTVZR	0B58
TEST1K	0576	TEXTERR	2A80	TEXTIN	2A5A	TEXTOK	13EC	TEXTOUT	2AF8
TEXTS	1362	TEXT0	DA1E	TEXT1	DA20	TEXT2	DA22	TEXT3	DA24
TEXT	136C	TINCLR	000F	TINEN	000E	TIMER20	DA1A	TIMER	E216
TIMESLOT	1288	TIMSPD	000D	TLLSTN	307A	TLTALK	30D0	TOLO	250A
TOOBIG	66F4	TOOMNY	656E	TOPHED	DA08	TOPUSE	33D2	TOTALK	38E2
TOTLLO	0926	TO	3386	TRMMNU	2CDA	TRMYPE	0924	TRNCFE	77AE
TRUES	369E	TRUE	D96A	TRYA2	425A	TRYCRV	276A	TRYPRE	2746
TRYWFM2	4D44	TR.FA	175C	TSTK	330C	TSTVZR	0A22	TS10.7B87	0F88
TS1	11AC	TS1.3	1242	TS23	11D6	TS4	11DE	TS4.10	1250
TS567	11E4	TS8	11EE	TS910	11F8	TTLOUT	33CE	THOPNTS	6D9E
TXTBUFR	DA62	TXTEND	4436	TXTNXT	4378	TXTQUIT	441A	TXTSCRN	42FE
TXT3OUT	4530	ULINE15	1646	UNITFLG	09A0	UPDTRO	0F88	UPDOWN	60A8
USER	000A	VALIN	9776	VALNXT	1854	VALOK	5B06	VALONLY	184A
VALOUT	28B0	VALWRD	1842	VCR0JT	158E	VCRPOS	5880	VCRS	0900
VCR1	589A	VCR2	6880	VCR	35DA	VERSION	098C	VERTADD	334E
VERTALT	334C	VERTCHP	3346	VERTL	3350	VERTH	0A26	VERTR	333C
VERTSF	1806	VERT	3586	VEXP	3338	VFPMOV	68C6	VNRTRN	154E
VOFFAB	3348	VPBACK	6136	VPD0VE	6158	VPOSNXT	6102	VPOSPNT	60EE
VPSTEP	611A	VPSTOP	6130	VRSNJH	DA04	VRTSCL	6814	VSCALJ	334C
VSERR	0D50	VSOUT	1554	VSRC1	5E7C	VSRO	0D5C	VSTK	33DA
VS	35CE	VXPBAD1	0A0C	VXPBAD2	0A12	VXPDRNG	09D0	VXPMAX	338C
VXPHIN	33A2	VXPMPY	0A75	VXPTST	09E4	VXFTSTL	0A00	VZROK1	0A48
VZROK	0886	VZRRO	186C	VZR	35AA	V.RO	22B2	WAIT1	06B4
WAIT2	06CC	WAIT4KEY	014E	WAIT	01C8	WALKONE	062E	WARNING	094C
WARNOUT	1738	WARNPER	3436	WARNSF	2898	WARN	3682	WAVEFORM	1820
WFMAQS	1EE4	WFHARG	4F5A	WFMAXN	D974	WFMBAS	DA06	WFMINW	280E
WFMIN	2812	WFHIO	2F75	WFMLED	33C8	WFMLT	0006	WFHMAX	4F9A
WFNON	18C6	WFMPRE	2F76	WFMSRT	1898	WFHX	16AD	WFHY	1664
WFH1	D996	WF42FP	6888	WFH2K	345A	WFH2	D998	WFH4K	346A
WFH8K	347A	WHCHSF	213A	WIPBJFR	00DE	WIPEDON	4746	WJADD	DA02
WHEAD	DA04	WPBUFR	DB48	WPCLK	DC08	WPDSP	DC20	WPDSY	E21A
WPGPIB	DC60	WPINT1	DC80	WPINT	DC70	WPKEYS	0820	WPKYTRE	0808
WPLVL1	DB60	WPLVL2	DB80	WPLVL3	DBA0	WPLVL4	DBC0	WPLVL5	DBE0
WPSWP	DC40	WRDNXT	1828	WRDONLY	181E	WRDVAL	1816	WRITELIN	43F2
WRV2TXT	2A6E	WSTK	3304	WDSF	2150	W1HEAD	D99A	W2HEAD	D99C
XCUTOON	02D8	XCUTE	0286	XCUTSTCP	02D4	XFRHEAD	698C	XFRHZ	486C

SYMBOL TABLE

XFRH	69C4	XFRIT	4BEE	XFRVRT	4BE4	XOUT	1682	XPSAME	4FDA
XSTCK	3520	XSTK	33D6	XWFM	3632	XYCRS	10D0	XYWFM	0978
YSTCK	361A	YSTK	33D8	YTCRS	10DC	YTWFM	097A	YWFM	3626
YZFLAG	DA0C	YZ1	DA0E	YZ2	DA10	Y	3328	ZCNSWFM	306C
ZEROCNS	304E	ZEROEXP	7222	ZEROSCL	7AAE	ZEROWFM	3052	ZEROW0	6974
ZMAX	3418	ZNRMLSL	7244	ZNRMLSR	7230	ZNRMLZ0	7288	ZNRMLVR	7278
ZROREF	67E6	ZTRAIL	7806						

CROSS REFERENCE LIST

ASCADR	2E68	5524	5600	D						
ASCII0	3797	2203	3862	4803	7129	D15973				
ASCII1	3796	7128	D							
ASCII2	3795	7127	D							
ASCII3	3794	7126	D							
ASCII4	3793	7125	D							
ASCII5	3792	7124	D							
ASCII6	3791	7123	D							
ASCII7	3790	7122	D							
ASCII8	379F	7133	D							
ASCII9	379E	3864	4805	7132	D					
ASCI	35EC	2158	7018	D						
ASCLF	2EA8	5556	5605	D						
ASCLO	2E80	5551	5602	D						
ASCOFF	2EB0	5575	5607	D						
ASCOFL	2EA0	5560	5604	D						
ASC0N	2EAC	5568	5606	D						
ASCR0	231E	3843	3848	4230	D					
ASCRQS	2EB4	5582	5612	D						
ASCTL	2E90	5547	5603	D						
ASCT0	2E70	5549	5601	D						
ASC	3329	4903	5292	6337	D					
ASNRMLZ	6FDE	13950	13957	13968	D					
ASTERISK	34DE	6839	D15006							
ASTFILL	7502	15006	D15009							
ASTRTN	750C	15004	15005	15010	D15024					
AT	3324	6331	D							
AUDIO	0013	6417	D 6604							
AUTOORD	5032	11516	11520	D						
AVGERR	1950	3016	3018	3020	3327	D				
AVGSTOP	1980	3013	3041	D						
AVGWFH	10D0	3023	3201	3260	3434	D				
AVG1WFH	1008	3437	D							
AWORD	E002	6451	D							
AWRD	E002	2040	2098	3323	3530	3534	3549	6347	D 6451	
BACKUP1	D908	712	1129	1248	1264	15529	D			
BACKUP2	D90A	724	1131	15530	D					
BACKUP	480E	9021	9028	D						
BACK11	203E	5363	D15850							
BACK13	2AD2	5179	D15866							
BACK14	2340	5252	D15882							
BACK15	2092	5419	D15898							
BACK16	275E	4774	D15910							
BACK1	2498	4483	D15748							
BACK20	3246	6143	D15974							
BACK24	09D0	1177	D16031							
BACK27	6378	12289	D16849							
BACK2	24EE	4524	D15766							
BACK3	2A08	5069	D15783							
BACK4	2A3E	5152	D15801							
BACK6	03A6	554	D15820							
BACK7	1084	3392	D15833							
BADGKEY	25EC	4609	4636	D						
BADHSC	435A	9430	9432	9439	D					
BADLBL	2896	4870	4886	D						
BADPNT	3390	3208	3242	3246	3296	3302	3476	3938	6751	D10365 10414
		10502	13407							
BADPOP1	5486	10455	10458	10459	10520	D				
BADPOP2	5490	10444	10447	10448	10521	10523	D			

CROSS REFERENCE LIST

BADPOP3	5495	10472	10526	D										
BADVAL	28B8	4906	4910	D	4946	4947	4976	4977						
BAD16	5A30	11101	11138		11162	11165	11158	D11184						
BAD17	59EA	11086	11093		11138	D								
BAD18	5050	10051	D10078											
BAD6	039C	1484	1495		1516	1541	D							
BAD7	519A	10209	D10214		10241	10243	10244							
BAKCROS	6A28	11064	11135		13150	D								
BATSET1	0850	1114	D 1117		1119									
BATSET2	0864	1121	D 1124		1126									
BD	370C	3141	7029	D										
BEEP	030E	479	481		484	D								
BEGNSCRN	4318	8472	D 8657											
BELLB	3326	5130	6334	D										
BELLIN	2AC8	5131	5159	D										
BIGLOP	5854	10950	D10958											
BI	3378	4359	6637	D										
BLANKST	35BA	2819	6997	D										
BLANK	34E0	2661	2724		6840	D15001	15246	15266	15277					
BLKFL	15C6	2660	D 2663											
BLNKDON	4350	8491	8496	D										
BLNKFLG	0AC6	2817	2821		2825	15684	D							
BLNKOVR	35C8	2828	7000	D										
BLNK	4342	8490	D 8495											
BO	3386	4352	6638	D										
BP	E210	6374	D 6375											
BUFAVL	0954	314	579		587	592	509	1062	4644	15572	D15943			
BUFCHK1	0074	257	D 276		15927									
BUFIN	03DC	544	564		571	576	D15818							
BUFRCHK	01CE	257	317		377	D15925	15950							
BUFRIT	038A	226	228		533	539	D							
BUFRLEN	33E6	297	314		585	505	6682	D15943						
BUSYLED	33C4	283	404		458	6507	D							
BUSYUP	1718	2780	2784	D										
BUSY1	3566	2787	6979	D										
BUSY	0011	6415	D 6607											
BUTTONHIT	0336	200	509	D										
BUZZIT	6CF4	582	1113		1120	5159	8158	8183	9149	13540	D			
BWORD	E004	6452	D											
BWRD	E004	2043	2099		3324	3443	3531	3535	6348	D 6452				
BYTEIN	248A	4360	4472	D										
BYTEOUT	2474	4353	4453	D										
CALCUP	145A	2521	2528	D										
CALCWFM	4FFE	9990	D10015											
CARRJT	34DD	6837	D 8543		8570	8575								
CD	37DD	3110	7030	D										
CERENBL	32AA	4704	4706		6277	D								
CHARSTART	33F4	6696	D 9252		12468	12500	12585							
CHE001	3392	6532	D12272											
CHE002	3394	1092	6533	D12267										
CHE004	3396	6534	D12262											
CHE008	3398	3310	6535	D12257										
CHE011	339A	6536	D12273											
CHE022	339C	1093	6537	D12268										
CHE044	339E	6538	D12263											
CHE088	33A0	3312	6539	D12258										
CHFFFF	3336	6541	D											
CHFF	3336	6540	D											
CHGSCL	4C8A	9441	9458		9470	9563	D 9842	9930						

CROSS REFERENCE LIST

CH400	337C	6521 D	6584	6624	6641	6643	6650	6551	15263	
CH40	336E	6517 D	6660							
CH5000	3388	6527 D								
CH6000	338A	6528 D	6628	6631						
CH7FFF	338E	1845	6530 D	14122	14123	14415	14417			
CH8000	3390	6531 D	6629	6630	6534	6751				
CH800	337E	6522 D	6583	6620	6540	6644	6649	6558	6662	15202
CH80	3376	6518 D	9230							
CH8	3348	6513 D	6663							
CKDECPT	120A	2215 D	2232							
CLERMFM	078A	1051 D	1055	1056						
CLINE15	3340	6834 D	11585	11588	11753	11930	11941			
CLINE16	3348	441	1424	1685	5398	5835 D	8855	8912	9348	9915 10949
		11020	11151	11280	11412	11608	11518	11754	12067	12112 12128
		12175	12361	12383	12385	12399	12421	12435	12463	
CLINE1	333A	6832 D								
CLINE2	333C	1684	1702	3189	6833 D					
CLKFAIL	05E2	965	973	985 D						
CLKFLG	0F94	1954	1957 D							
CLKGOOD	05EC	984	987 D							
CLKINT	337A	961	969	1221	1243	1798	1957	6505 D		
CLKRST	338C	956	988	1958	6503 D					
CLKWAIT	094A	1219 D	1222							
CLLKEY	37DE	7161 D	8363							
CLOCK	0E26	194	1835 D							
CLPFLAG	0AAC	8271	8278	8465	8955	9137	15671 D			
CLPKEY	37E2	7163 D								
CLRDSP	63C0	12318 D	12326							
CLRLAST	48C8	9110	9120 D							
CLRLIN	48AC	9108	9111 D							
CLRPTR	40D6	658	8247 D	8319	9215					
CLRSTK	63D4	12320	12327 D							
CLRST	179C	2807	2828 D							
CLRTXT	128E	1058	1578	1590	1606	2314 D	5195	5505	8161	8192
CLWERR	0CE4	1662	1664	1666	1581 D					
CLWRD	0CF0	1678	1680	1684 D						
CLWWRN	0CDE	1673	1679 D							
CMDBUF	090C	4562	4577	4587	4619	4637	4639	15534 D		
CMDLHM	251E	4552 D	4555							
CMDINDX	0918	273	1891	4408	4433	4493	4512	4567	4573	4578 15535 D
CMDIN	2550	4553	4570 D							
CMDSRCH	2576	4582 D	4607							
CMDSZ	000A	4564	15535 D							
CMD	337C	4339	6641 D							
CMPEND	4C78	9536	9540 D							
CMPGT	4C74	9533	9537 D							
CMPLOP	4C64	9519	9526	9538 D	9542					
CMPOMAX	5CFA	11504 D								
CMPOMIN	5D16	11508	11512 D							
CMPRCMD	2586	4588 D	4591							
CMPRQUE	25CE	4620 D	4623							
CMPWFM	4C34	9414	9513 D							
CNSWFM	091C	4751	4807	4840	5845	5853	15538 D			
CNTGROS	6A12	11702	13141 D							
CNTWFMS	198E	3040	3045 D							
CNTWFM	1F7A	3045	3640 D							
CNVBUF	0A8A	2167	2178	15663 D						
CNVOVER	75AC	14939	14968	14993 D						
CNVWRD	1174	2162 D	2260							

CROSS REFERENCE LIST

CNVZERD	759E	14921	14930	14970	14984						
CN16	3330	6474	D								
CN1	3336	859	5094	5843	6477	6540	5541	6714	5734	11859	11978
CN2	3334	1204	4735	6476	D						
CN4	3332	6475	D								
COLDUP	04E2	164	687	D							
COLONB	3323	6330	D								
COMMAB	3321	1181	4830	4962	4965	5295	5394	6117	6153	6328	D
COMMAND	24D2	4504	4505	4510	D						
COMOUT	239C	5295	D	5303	5327						
CONCOM	6218	12107	12120	12137	D						
CONDSP	2E44	5584	D	5597							
CJNSTR	DADA	12146	15700	D							
CONST2	4ED8	9808	9848	D							
COV	3408	6710	D14231								
CR3	331E	5126	5222	5455	6066	6325	D15843				
CRERR0	2J3E	3724	3735	3736	3752	D					
CRERR1	2840	3740	3743	3746	3753	D					
CRIN	2AA4	5127	5136	D							
CRLFOUT	232A	5213	5220	D							
CRLF	33B1	2308	2372	5212	6594	D					
CROSFLAG	DA3E	11052	11061	11124	11133	11332	11335	11583	11700	13164	13266
		15680	D								
CROSFLG	DA1C	13143	13148	13153	13166	13250	15640	D			
CROSLOP	6A74	13183	D13197	13207	13273						
CROSNUM	DAC0	11053	11062	11125	11134	11333	11336	11684	11701	13209	13268
		15681	D								
CRSKEY	5FBA	11840	11851	11858	D						
CRSOPWD	D960	2067	2537	15570	D						
CRSRO	608E	11916	11939	11941	D						
CRSTEP	5FE8	11880	D11883								
CRSWRN	608A	11897	11902	11909	11914	11940	D				
CRS1KEY	5F92	11835	11839	D							
CRS1R	6008	11893	11894	11898	D						
CRS1	5FF8	11892	D11923								
CRS2KEY	5FAC	11846	11850	D							
CRS2R	6034	11905	11906	11910	D						
CRS2	601C	11899	11903	D							
CR	33AF	6592	D								
CTRLIN	2A84	5106	5124	D							
CTRLOUT	2AFE	5202	5205	D							
CURSORS	1094	2023	2056	D							
CURSJR	D954	2056	2065	2079	2629	3184	3187	3224	9518	10211	10284
		10383	10437	10461	10520	10710	11047	11085	11161	11179	11209
		11273	11579	11581	11587	11606	11616	11666	11693	11711	11738
		11773	11839	11850	12897	12931	15564	D			
CURSPLT	D950	15563	D								
CURS1	D956	9520	9528	10216	10286	10385	10451	10475	10716	10724	11181
		11211	11275	11582	11584	11682	11690	11697	11714	11776	11777
		11779	11781	11892	11895	11898	11900	11901	11904	11907	11932
		12308	12900	12912	12934	12937	15565	D			
CURS2	D958	1063	1064	9527	10215	10476	10725	11051	11214	11275	11582
		11584	11679	11717	11777	11779	11784	11787	11903	11904	11907
		11910	11912	11913	11932	12306	12307	12908	12936	15566	D
CURVE	2FF2	4779	5786	D							
CHERR	1FD6	3642	3645	3646	3557	3658	3669	D			
CYCFLAG	DA14	689	750	752	1224	1265	15636	D			
CYCLEUP	04EA	690	D	1249	1266						
CYCLE	DA12	1985	2001	2125	15635	D					

CROSS REFERENCE LIST

CYCTEST	0968	1225	1234	1239 D							
CYCHAIT	0970	1241	1244	1247							
C0	3338	857	1195	2860	4577	4592	4625	4627	4670	4672	4865
		6478 D	6562	6617	6736	6738	8544	9580	11749	11839	12003
C10P	5E30	11695	11698 D								
C100	3374	3236	3349	3356	6509 D	8526					
C1024	337C	2087	6512 D	6521	12271						
C10	334C	1062	5259	5532	6488 D	6551	6565	8529	8914	11232	11234
		12668	12670	12695	12736	14945	15135	15144	15219		
C11	334E	4738	6489 D	6552							
C128	3376	3314	3401	3403	6509 D	6518	12256				
C12	3350	1038	5186	5586	6490 D	6550	6566	12513			
C13	3352	1039	5098	6491 D	6558						
C14	3354	1138	5138	6492 D	6557	6567	14582				
C15	3356	2632	3463	5512	6493 D	11186	12939	13100	14174	14187	
C16	3358	2333	2335	3191	3504	3662	5552	6494 D	6514	8520	9250
		10628	10631	10684	12466	12583	12684	12710	12717	12737	13663
		14190	14285								
C17	335A	6495 D	9258	12591	13760						
C18	335C	1145	6496 D								
C1	333A	292	325	334	352	391	450	1044	1069	1140	1208
		1635	1648	1672	1877	2301	2303	2330	2364	2558	2586
		2598	2634	2653	2598	2726	2886	3193	3263	3291	3321
		3327	3451	3475	3528	3526	4456	4651	4751	5103	5112
		5136	5271	5509	5523	5567	5574	5927	6479 D	6708	6832
		8181	8191	8264	8271	8320	8431	8465	8577	8761	8826
		8883	8996	9137	9190	9214	9217	9331	9368	9378	9784
		9844	10371	10437	10461	10520	10710	11052	11061	11124	11133
		11179	11332	11335	11579	11850	12543	12549	12520	13266	15819
		15865	15949								
C2TOOBIG	5F7C	11783	11786 D								
C2JF	5EE0	11713	11716 D								
C20	335E	6497 D									
C256	3378	6510 D	6519	12261							
C2	333C	327	629	654	1090	1192	1677	1701	2007	2596	5540
		5858	6480 D	6554	6715	6833	8966	9212	9334	9636	10070
		10211	10760	11047	11085	11161	11581	13517	13524	15881	
C31	3360	6498 D	14783								
C32	3362	6499 D	6515								
C33	3364	6500 D	9147								
C3	333E	392	3479	5101	5557	5481 D	6555	6702	9337	11336	15893
		15096	15897								
C40	3366	6501 D	8640								
C41	3368	6502 D									
C48	336A	6503 D	6516								
C4	3340	4556	4677	4682	5569	5575	5591	6482 D	6563	6664	6698
		6704	6834	9340	12646	12659	12664				
C50	336C	6504 D									
C512	337A	6511 D	6520	6754	12265						
C5	3342	480	6483 D	8562							
C64	336E	6505 D	6517								
C6	3344	711	770	4396	6484 D	6556	8572				
C7	3346	1022	3308	4398	6485 D	6553					
C8	3348	5515	5525	5561	5728	6486 D	6513	6564	6753	6835	12636
		12727	12774								
C98	3370	6506 D									
C99	3372	6507 D									
C9	334A	5432	5507	6487 D							
DACR	3380	4344	4431	6625 D							

CROSS REFERENCE LIST

DSPRO	D9CA	1095	1580	1581	1594	1597	1610	1611	2119	5504	8163
		8185	8187	8188	15615						
DSPRST	3332	1003	2012	2037	2048	2095	2104	3063	6598	D	
DSPSTP	3336	2046	2102	2113	3059	6600	D				
DSPSTR	3384	2050	2106	6599	D						
DSPTKO	2DDA	5545	5549	D							
DSPWFMI	33A6	1097	6575	D							
DSPWFM	D9CE	1097	1595	1613	1717	1728	1734	1740	2003	2024	3055
		3183	3185	3186	3225	3308	12346	15618	D16043	16046	
DSWERR	0C96	1622	1624	1626	1553	D					
DSWRD	0CA2	1649	1652	1656	D						
DSWHARN	0C8C	1642	1650	D							
DSW	35C2	2604	6932	D							
DVCR	35FA	2644	6948	D							
DXADR	E012	6453	D								
DYADR	E114	6455	D								
EDITMODE	43E4	8542	8545	8546	8553	D					
EDITPNT	DAAA	330	333	621	555	665	1073	2757	8280	8295	8317
		8321	8328	8332	8423	8433	8434	8498	8648	8649	8652
		8653	8722	8728	8776	8821	8825	8874	8881	8950	8950
		8968	8995	9004	9114	9017	9019	9035	9063	9073	9075
		9089	9113	9191	9205	9374	9376	15670	D		
EEXCHK	657C	12508	12541	D							
EEXHIT	67E0	12800	12803	D							
EEXKEY	3926	7194	D 8673	12507							
EEXMNU	33EA	6687	D12642	12799							
EEXMSG	65B0	12545	12558	D							
EEXONE	65AE	12551	12557	D							
ENABLE	0936	1193	1197	1201	1212	D					
ENDAQ5	1C06	3241	3247	D							
ENDASC	370C	2258	7026	D							
ENDFLG	33F2	5430	5436	5453	6595	D					
ENDGND	1C4C	3259	3274	D							
ENDKEY3	382D	625	1076	2758	7198	D 8165	8208	8410	8419	8538	8565
		8735	8779	8930	8951	8969	9010	9107	9121	9192	9207
		9275									
ENDKEY	382C	7197	D								
ENDLINE	33E4	6681	D 8412	8853	8910	9007					
ENDMNU	455E	8691	8695	D							
ENDMPY	4AF6	9400	D 9512								
ENDOUT	1744	2795	2798	D							
ENDPNT	3802	7177	D 8337								
ENDPRGM	2032	5437	5454	5473	D						
ENDP11	958E	15845	15847	15850	D						
ENDP18	972E	15922	15927	D							
ENDP19	975A	15946	15951	D							
ENDP1	9518	15744	15746	15748	D						
ENDP20	9786	15970	15974	D							
ENDP21	97C4	15995	15997	15999	D						
ENDP2	9638	15761	15763	15766	D						
ENDP3	9654	15779	15784	D							
ENDP4	9570	15797	15802	D							
ENDP6	958C	15814	15818	D							
ENDP7	96A2	15831	15833	D							
ENDQTB	3016	4630	5809	D							
ENDSCRN	4448	8587	8592	D							
ENDSRQ	2F2A	5688	5698	D							
ENDSTAT	354A	2800	6973	D							
ENDTAB	3C36	425	637	7809	D						

CROSS REFERENCE LIST

FPSUB	6F50	9386	9789	10321	11141	11331	13917	D15317	15403		
FPTEMP	D9C2	726	1091	1792	1793	1794	3080	3221	15612	D	
FPTRNC	75A6	10315	11113	11376	14815	D15398					
FPWAIT	0E12	1796	D 1801								
FP1E	333A	3145	6708	D 9576	9851	11304	15432				
FP1M	3404	3144	6707	D 9578	9850	10072	11306	15434			
FP10E	3340	6704	D10086	12881	13808						
FP10M	33FE	6703	D10085	12880	13807						
FP1000E	3402	3116	6706	D							
FP1000M	3400	3115	6705	D							
FP12E	3340	6698	D14603								
FP12M	33F6	6697	D14602								
FP2ASC	75E0	2905	3470	6216	14907	D					
FP2ELE	6D1E	10445	10456	10719	13568	D					
FP2INT	75A0	1623	1663	1709	3017	3584	4917	4937	8851	8908	10093
		10336	10357	10399	11447	11756	12056	12077	12140	12164	12194
		12243	13582	14537	14813	D					
FP2WFM	6368	11324	11659	12204	13315	D					
FP20E	33FA	1493	6700	D13838	15356	15428					
FP20M	33F8	1492	6699	D13837	15355	15427					
FP6E	333E	6702	D14590	14596							
FP6M	33FC	6701	D14592	14598							
FP	E010	726	728	777	821	985	1012	1023	1026	1037	1260
		1794	1916	3092	3221	3647	3664	3725	6352	D 6457	
FRCTOUT	77F4	15177	D15184								
FRDOU	1430	2514	D 3196	4737	5097	5185	5258	5425	9168	11931	
FREE	1202	2182	2212	D							
FRMTOUT	230C	5200	5210	D							
FRONT	E010	6457	D								
FRSTCLK	06C4	962	966	D							
FRSTCRV	2784	4782	4786	D							
FRSTDGT	27AA	4797	4803	D							
FRSTDIGT	64E6	12474	12486	D							
FRSTPNT	29CA	5035	5038	D							
FRSTPRE	2760	4755	4759	D							
FRSTPT	1E86	3492	3499	D							
FRSTRES	337A	1057	6754	D							
FSTIPNT	55EE	10660	10662	10664	10679	10681	D				
FSTPT	694E	13251	13276	D							
FSTSHFT	56DE	10761	10776	D							
FUZZCHK	6954	13196	13206	13262	13294	D					
FWDGROS	6A1C	11054	11126	11334	11339	11685	13145	D			
FWDNXT	4JA2	8207	D 8213	8215	9025	9112	9277	9375			
FWDSTOP	4038	8209	8216	D							
GETEEX	67D2	12519	12570	12796	D						
GETENDS	5A62	10941	11038	11094	11204	D11325	11653	13520	13706		
GETFRST	6A6A	13165	13167	13177	D						
GETLINE	45B4	5428	8316	8763	D 8880	9105					
GETMID	58F2	11035	D11091	11163	11177						
GETPER	5342	11289	11298	11319	D						
GETPNT	45AA	8759	D 8823								
GETRUN	245C	4350	4431	D							
GET	3382	4349	6642	D							
GEXEKEY	37EA	7165	D 8155	8306	8308						
GFJRMAT	D920	5263	5267	5378	6215	15540	D				
GHRZSF	19F6	3074	3079	D							
GHTIME	59EE	11139	D11170	11189	11342						
GNDAVE	105E	3375	D 3382								
GNDPJS	1090	3399	3401	D15832							

CROSS REFERENCE LIST

I1	0001	6431	J								
I2FP	6FDC	13229	13235	13242	13851	D					
JUMP	37A0	2268	7134	D							
KBCODE	E00E	514	1115	1122	6456	D	8987	9051	11863	11880	11915 11937
		11982	12018	12022							
KBID	0010	6414	D								
KBINT	337C	6584	J								
KBRST	0J16	6420	D	6458	6601						
KB	E00E	6351	D	6456							
KCUART	E314	6379	J								
KDUART	E214	6378	D	6379							
KEEPON	6670	12645	12648	D							
KEEPRQS	241A	4397	4399	4401	D						
KEYABS	4E5C	7285	9803	D							
KEYAQR	1904	3001	D	7227							
KEYAQS	195E	3031	D	7483							
KEYAREA	5A04	7289	11262	D							
KEYAVG1000	191C	3009	J	7281							
KEYAVG100	1916	3007	D	7313							
KEYAVG10	1910	3005	D	7345							
KEYAVG	1922	3011	D	7225							
KEYBOTH	0C02	1605	J	7485							
KEYBUFR	DA3A	296	589	603	585	15656	D				
KEYB	0957	666	2779	8399	8424	15574	D				
KEYCEROFF	25C8	4704	D	7375							
KEYCERON	25D0	4706	J	7711							
KEYCHS	649E	7273	12462	D							
KEYCLD	0CFE	1690	D	1697	1698	7577					
KEYCLF	49D4	679	7769	7805	9233	J					
KEYCLL	4886	7162	7411	9099	D						
KEYCLP	48DC	7163	7747	9132	D						
KEYCLR	6432	7611	12376	D							
KEYCLW	0CA8	1661	D	7361							
KEYCLX	644C	7315	12385	D							
KEYCNS	61E0	7329	12107	D							
KEYCRS1LFT5F86		7261	11832	D							
KEYCRS1RGT5F8C		7263	11836	D							
KEYCRS1	5DC8	7791	11579	D							
KEYCRS2LFT5FA0		7451	11843	D							
KEYCRS2RGT5FA6		7449	11847	D							
KEYCRS2.1	5DD0	7455	11581	D							
KEYCYCLE	099E	1260	D	7487							
KEYDEC	649E	12460	J								
KEYDELAY	5A3A	7257	11177	D							
KEYDIFF	5714	7717	10821	D							
KEYDIV	4A4A	7275	9327	D							
KEYDOTS	0D72	1734	D	7673							
KEYDSW	0C3C	1621	D	7363							
KEYEEX	649E	7433	12461	D							
KEYENERGY	5ADA	7287	11264	D							
KEYENTER	648A	7251	12433	D							
KEYEXP	4F0A	7509	9884	D							
KEYEXR0FF	25D8	4708	D	7377							
KEYEXRON	26E0	4710	D	7713							
KEYFALL	595C	7511	11084	D							
KEYFREQ	5318	7567	11297	D							
KEYF	49C2	678	7172	7175	7447	7783	9225	D			
KEYGEXE	4000	7165	7221	8151	D						
KEYGND	196C	3035	D	7223							

CROSS REFERENCE LIST

KEYREADX	2708	4731 D	4734	7435					
KEYREMOFF	25E8	4712 D	7399						
KEYREMON	25F8	4714 D	7735						
KEYRISE	595A	7255	11089 D						
KEYRMS	58C4	7319	11006 D						
KEYROLL	6458	7559	12394 D						
KEYRQSOFF	25A8	4696 D	7485						
KEYRQSON	25B0	4698 D	7741						
KEYRQS	2F6A	5728 D	7249						
KEYRTN	4502	7753	8806 D						
KEYRUN	4970	673	7423	9187 D					
KEYSAVE	2C8E	5418 D	7160	7467					
KEYSCOPE	0BA2	1577 D	7231						
KEYSENDX	233C	5251 D	7771						
KEYSGN	4E60	7561	9805 D						
KEYSMOOTH	5072	7387	10063 D						
KEYSQRT	4F16	7347	9888 D						
KEYSTART	498C	7759	9204 D						
KEYSTAT	15F2	2762	2767	2772 D					
KEYSTEP	476A	674	7237	8965 D					
KEYSTORED	0BC8	1589 D	7229						
KEYSWH	2F62	5724 D	7403						
KEYSWL	2F5A	5720 D	7739						
KEYTAB	3936	420	2845	4585	5441	7216 D	8351	8560	12528
KEYTEXT	2ACE	5178 D	7247						
KEYTIME	0J62	1727 D	7359						
KEYTRE	004C	233	234 D	487	565				
KEYUP	17A6	2831 D	9166	9228	9238				
KEYVCRD	5E16	7295	11616 D						
KEYVECT	0J80	1740 D	7357						
KEYVMDADD	0DA6	1768 D	7427						
KEYVMDALT	0D9A	1765 D	7425						
KEYVMDCHOP	0DB2	1771 D	7429						
KEYVMDL	0DBE	1762 D	7409						
KEYVMDR	0DBE	1774 D	7431						
KEYVMD	0J38	1764	1767	1770	1773	1776 D			
KEYVPDN	6096	7463	11972 D						
KEYVPUP	60A0	7465	11975 D						
KEYVSCL	0AB6	1421 D	7393						
KEYVS	0J2A	1707 D	7675						
KEYVXPD	098A	1295 D	7391						
KEYVZR	0AAA	1417 D	7395						
KEYWFM	6162	7331	12054 D						
KEYWIDTH	5A1A	7259	11160 D						
KEYXFR2	0948	423	8344	8367	15559 D				
KEYXFR	0946	453	573	1060	8345	8358	15558 D		
KEYX2Y	646E	7283	12416 D						
KEY0	649E	677	7269	8663	12449 D				
KEY1	649E	7301	12450 D						
KEY2CNS	61F6	7625	12128 D						
KEY2HCRD	5EEC	7569	11738 D						
KEY2HSCL	0AC8	1438 D	7733						
KEY2ORD	5C44	7719	11431 D						
KEY2PNT	626E	7623	12189 D						
KEY2P.W	62E4	7621	12241 D						
KEY2TEXT	2A1C	5190 D	5093	7503					
KEY2VCRD	5E2E	7571	11551 D						
KEY2VSCL	0AD4	1442 D	7729						
KEY2VZR	0306	1481 D	7731						

CROSS REFERENCE LIST

MNUFREQ	3E72	7568	8047	D							
MNUF	3C46	7448	7784	7825	D						
MNUGND	3C64	7224	7839	D							
MNUGOTJ	3DB2	7420	7983	D							
MNUGPIB	D934	4637	4638	4639	7472	15550	D				
MNUGSB	3DAE	7418	7981	D							
MNUHCRD	3CEE	7294	7891	D							
MNJHMDALT	3E52	7440	8037	D							
MNUHMDA	3E4C	7438	8035	D							
MNUHMDB	3E62	7444	8041	D							
MNUHMDCHOP	3E5A	7446	8039	D							
MNUHPLFT	3D72	7724	7959	D							
MNUHPRGT	3D78	7726	7961	D							
MNUHSCL	3D94	7398	7971	D							
MNUHXP	3D7E	7390	7963	D							
MNUID	3C8E	7442	7853	D							
MNUIFXEQY	3F24	7758	8107	D							
MNUIFYGTX	3DBC	7422	7987	D							
MNUINTG	3D5A	7382	7951	D							
MNUIOCOFF	3EF8	7402	8091	D							
MNUIOCON	3EF2	7738	8089	D							
MNUITRP	3D64	7386	7955	D							
MNULBL	3DB8	7756	7985	D							
MNULNN	3F20	7750	8105	D							
MNULN	3C9E	7254	7859	D							
MNUMAX	3D32	7352	7929	D							
MNUMEAN	3D0A	7322	7907	D							
MNUMID	3D10	7324	7909	D							
MNUMINUS	3D58	7372	7949	D							
MNUMIN	3D36	7354	7931	D							
MNUMULT	3D00	7308	7901	D							
MNUMEXT	3DA8	7234	7416	7979	D						
MNUMOP	3C62	7240	7242	7246	7266	7268	7278	7280	7298	7300	7310
		7312	7318	7342	7344	7350	7380	7408	7458	7460	7462
		7470	7474	7478	7482	7488	7490	7492	7494	7496	7498
		7500	7502	7506	7508	7514	7516	7518	7520	7522	7524
		7544	7546	7548	7552	7554	7556	7558	7564	7566	7574
		7576	7596	7598	7600	7604	7606	7608	7610	7614	7616
		7618	7620	7648	7650	7652	7656	7658	7660	7662	7664
		7666	7668	7670	7672	7680	7700	7702	7704	7708	7716
		7722	7728	7744	7746	7752	7762	7764	7766	7768	7770
		7774	7776	7778	7780	7782	7786	7788	7790	7794	7796
		7798	7800	7802	7804	7808	7837	D			
MNUOFF	3DEA	7454	8003	D							
MNUOPCOFF	3EEA	7374	8087	D							
MNUOPCON	3EE4	7710	8085	D							
MNUORD	3D60	7384	7953	D							
MNUPAUSE	3C50	7476	7831	D							
MNUPER	3CEA	7292	7889	D							
MNUPLUS	3D2A	7340	7925	D							
MNUPNT	3D18	7328	7913	D							
MNJPREV	3DA2	7414	7977	D							
MNUPROG	3C86	7236	7244	7851	D						
MNUP2P	3D3A	7356	7933	D							
MNUP.W	3D14	7326	7911	D							
MNURDJUT	3E08	7480	8013	D							
MNUREADX	3DC6	7436	7991	D							
MNUREMOFF	3EDC	7400	8083	D							
MNUREMON	3ED6	7736	8081	D							

CROSS REFERENCE LIST

MNURISE	3CA2	7256	7861	D
MNURMS	3C06	7320	7905	D
MNUROLL	3E68	7560	8043	D
MNURQSOFF	3F06	7406	8095	D
MNURQSON	3F00	7742	8093	D
MNURQS	3C4C	7250	7829	D
MNURTN	3F15	7754	8101	D
MNURUN	3C48	7424	7827	D
MNUSAVE	3E02	7468	8011	D
MNUSCOPE	3C78	7232	7847	D
MNUSENDX	3F2E	7772	8111	D
MNUSGN	3C08	7562	7883	D
MNUSHOOTH	3C5A	7388	7957	D
MNUSQRT	3D2C	7348	7927	D
MNUSTART	3F1A	7760	8103	D
MNUSTEP	3C5C	7238	7835	D
MNUSTOP	3C56	7218	7220	7833 D
MNUSTORED	3C70	7230	7845	D
MNUSWH	3C9A	7404	7973	D
MNUSWL	3F0E	7740	8097	D
MNUTABL	3C38	7820	D	
MNUTEXT	3C92	7248	7855	D
MNUTIME	3C44	7360	7937	D
MNUVCRD	3CF4	7296	7893	D
MNUVECT	3D3E	7358	7935	D
MNUVNDADD	3E36	7428	8029	D
MNUVNDALT	3E2E	7426	8027	D
MNUVNDCHOP	3E3E	7430	8031	D
MNUVMDL	3E28	7410	8025	D
MNUVMOR	3E46	7432	8033	D
MNUVPDN	3DF6	7464	8007	D
MNUVPUP	3DFC	7466	8009	D
MNUVSCL	3C8A	7394	7967	D
MNUVS	3EA6	7676	8065	D
MNUVXPD	3C84	7392	7965	D
MNUVZR	3D90	7396	7969	D
MNUWFH	3D20	7332	7917	D
MNUWIDTH	3CAE	7260	7865	D
MNUX2Y	3C02	7284	7881	D
MNU0	3CC0	7270	7871	D
MNU1	3CFA	7302	7895	D
MNU2CNS	3E94	7626	8059	D
MNU2HCRD	3E78	7570	8049	D
MNU2HSCL	3D08	7734	7997	D
MNU2ORD	3E84	7720	8071	D
MNU2PNT	3E8E	7624	8057	D
MNU2P.W	3E88	7622	8055	D
MNU2TEXT	3E18	7504	8019	D
MNU2VCRD	3E7E	7572	8051	D
MNU2VSCL	3C0C	7738	7993	D
MNU2VZR	3D02	7732	7995	D
MNU2WFH	3E9A	7628	8061	D
MNU2	3CFC	7304	7897	D
MNU3	3CFE	7306	7899	D
MNU4	3C24	7334	7919	D
MNU5	3D26	7336	7921	D
MNU6	3C28	7338	7923	D
MNU7	3C52	7366	7943	D
MNU8	3C54	7368	7945	D

CROSS REFERENCE LIST

NEXTMNJ	233C	5434	5452	5457	5472					
NEXTPRV	4865	9061	9072	D						
NEXTQUIT	4844	9006	9058	D						
NEXTRUN	479E	8993	D							
NEXTSF	204E	3789	3819	D						
NEXTWFM	1888	2955	2982	D						
NOAVERJ	1E44	3455	3456	3473	D	3474				
NOBAD	528E	10297	10302	D						
NOBATTERY	0540	715	717	720	722	725	728	D		
NOBEEP	0312	483	486	D						
NOBUFR	03EE	578	583	D						
NOCHG	6CE6	13506	13509	13511	13526	D				
NOCLP	415A	8272	8298	D						
NOCRJS	6A94	13157	13185	13186	13199	13259	13260			
NOCRS	5272	10285	10290	D						
NOCRT	43C0	8540	D							
NOCYC	0988	1263	1267	D						
NODECPT	1238	2219	2223	2231	D					
NODIV	5808	11010	11013	D						
NOFKEY	494C	9163	9167	D						
NOFPKEY	340A	1024	1027	6827	D					
NOFWS	593C	11048	11057	D						
NOGMD	008A	265	268	D						
NOG00D2	0A9C	1297	1347	1360	1363	1366	1400	D10209		
NOG00D3	6306	12242	12252	D12315						
NOG00D4	6175	12055	12057	12059	12061	D12076	12078	12080	12083	12149
NOG00D5	6238	12139	12141	12143	12144	12149	D			
NOG00D6	5EE6	11658	11661	11667	11705	11719	D			
NOG00D7	5164	10059	D10066	10068	10074					
NOGPIBK	242C	4402	4404	4406	D					
NOHCRS	5F04	11746	D11750	11752						
NOITRP	5470	10491	10498	10513	D					
NOKEY	180C	2869	D							
NOLABL	4506	8611	8658	D						
NOLINE	45F0	8780	8789	D						
NOLSTNR	3164	6022	D	6028						
NONSPCL	7322	14381	14383	14385	14421	D				
NONTR	02F4	440	463	472	D					
NONUMCHG	04D2	424	572	D						
NONUMLIN	4304	8547	D	8555						
NONXT	47F0	9002	9003	9018	D	9057				
NOPHER	5E10	11607	11613	D11617						
NOPKEY	380A	7182	D	9164						
NOPL	15EA	2759	2768	D						
NOPRGNTR	0242	407	409	D						
NOPURG	4156	8276	8285	8288	8291	8297	D	8303	8369	
NOQUER	2456	4414	4418	D						
NOQUE	26A6	4669	4694	D						
NORMAL	3202	6293	D							
NORQS	3310	6281	6283	6285	6316	D				
NORSV	3308	6304	6314	D						
NORUN	498A	9189	9195	D						
NOSET	0334	497	499	501	503	D				
NOSIG	6598	12656	12664	D						
NOSLAVE	1FD8	3663	3666	3668	3670	D				
NOSPEC	41F5	8334	8338	8348	D					
NOSRA	0340	1504	D							
NOSTPWV	00CA	286	289	D						
NOSTST	1776	2809	2815	D						

CROSS REFERENCE LIST

NOSYMB	67B6	12763	12772	D						
NOTAQRD	1E8E	3490	3503	D						
NOTAQS	1ACC	3105	3135		3148	D				
NOTCMD	2392	4340	4349	D						
NOTCTRL	12AA	2267	2271		2273		2275		2276	D
NOTDGT	279C	4798	4804	D	4805					
NOTDIGIT	2112	3863	3865		3871	D				
NOTEND	24AE	4489	4493	D	4505		4534		4536	4569 4575
NOTE	6D04	13546	013555							
NOTFAL	599E	11099	11110	D						
NOTFILL	21C0	3939	3945	D						
NOTFST	41AC	8313	8314		8321	D				
NOTFULL	27F2	4821	4826	D						
NOTGPB	31B2	6031	D 6034							
NOTIDL	03C0	562	567	D						
NOTLBL	298A	4872	4882	D						
NOTLNN	42DA	8426	8434	D						
NOTOPNT	5D74	11537	11544	D						
NOTQ	2538	4557	4562	D						
NOTRES	2A0C	5037	5074	D						
NOTRMLS	31A0	6059	D 6065							
NOTRML	31D6	6074	D 6080							
NOTRJP	2118	3841	3874	D						
NOTSF	2D9E	3792	3798		3803		3915	D		
NOTSUS	01E8	382	385	D						
NOTTRM	2566	4574	4577	D						
NOTXTNUM	4544	8664	8684	D						
NOTYSF	217C	3909	3917	D						
NOUNIT	2938	4963	4967	D						
NOVRFLW	71CC	14176	14189		14192		14217		14227	D
NOVZRW	1896	2927	2930	D						
NOWARN1	047E	635	642	D						
NOWARN	467E	8857	8862	D						
NOWIPE	411C	8274	8277	D						
NOXNEG	66D0	12683	12686	D						
NO.ACQ	0FE4	1996	1999	D						
NO.INTR	1002	2005	2009	D						
NRMCHKZ	71DE	14258	D							
NRMCHK	7134	14159	D14370		14659		14726		14729	14911
NRMLA	7074	14043	14047		14050	D14052				
NRMLIZZ	71E4	13915	13929		14034		14073		14251	D
NRMLIZ	713C	10633	10637		10685		13668		13919	13933 14038 14077 14163 D14408
		14587	14788							
NRMLZED	71B8	14175	14178		14188		14216		14218	D
NRMLZO	715A	14160	14166		14177	D				
NRMLZ	7186	14162	14180		14194		14195	D		
NR2	332C	5378	6339	D						
NR3	332E	5263	5267		6340	D				
NTRKEY	3934	444	7203	D						
NT8000	7500	14914	14921	D						
NULLFIL	1310	2350	D 2354							
NULLHEAD	343A	6758	D13876							
NULLIN	12FA	2340	D 2599		2700		3194		3507	8492 8583 8588 8603
NULLOUT	2924	5211	5218	D						
NULLWFH	6F26	1053	3264		5853		13870	D		
NULL	33AC	2175	2353		2449		2470		2471	5210 6589 D12638 12754
NULWRD3	342F	6745	D							
NULWRD	342E	6744	D 9583							
NUMCHG	0280	426	430	D						

CROSS REFERENCE LIST

		11329	11393	11466	11532	11554	11578	11589	11708	11992	12094
		12209	12225	12896	12914	13239	13355	13525	13622	13708	15568 D
OPWFMH	D95E	1066	1352	1370	1504	1536	1675	1699	2553	2566	2570
		2574	2575	2581	2648	2676	2681	2685	2686	2693	5316
		5323	5330	5333	5350	9812	9920	10228	10560	10612	10853
		10959	10975	11142	11265	11368	11400	11403	11440	11483	11488
		11520	11525	11674	11755	11986	12831	12950	12871	12905	12917
		13103	13322	13367	13522	13571	13623	13669	13709	13774	15569 D
OPWFM	D95A	1346	1455	1509	2538	12003	12091	13067	13510	13513	13519
		15567	J								
OPWH2W0	69A8	4790	4847	9418	9834	9927	10104	10260	10480	10552	10826
		13061	D								
OPW	359E	2536	6920	D							
ORDLP	53F2	11388	11392	D							
ORDOK	58E8	11384	11389	D							
ORDWFM	5D50	11534	D11558								
OUTFP	3286	6218	D 5222								
OUTLBL	2372	5269	D 5279								
OUTMNU	25D0	5442	D 5445								
OUTPUT	0004	6402	D 5612								
OUTSTR	3262	5270	5294	5662	5665	5686	5712	6178	D 6182		
OUTVAL	3270	5264	5302	5390	6209	J					
OVFADJ	6FC0	13956	D13980								
OVFLW	75CC	14823	14833	D							
OV	3+06	6709	D13964	14123	14324	14664					
OWFMPRE	2B8C	5289	D 5749								
OXINCR	23CE	5315	D 5769								
OXUNIT	23DE	5322	D 5773								
OXZERO	23C2	5310	D 5765								
OYMULT	2C1A	5345	D 5781								
OYUNIT	2C22	5349	D 5785								
OYZERO	23F0	5329	D 5777								
O	3327	5311	6335	D							
PANICB	3833	331	7202	D							
PANIC	3832	288	416	496	543	4406	7201	D			
PASSCMD	251A	4645	4648	D							
PASSIT	655A	12640	D12647	12662	12676						
PATCH11	95A6	5362	15843	D							
PATCH13	95C2	5178	15860	D15864							
PATCH14	95DC	5251	15876	D15880							
PATCH15	95F6	5418	15892	D15896							
PATCH16	9710	4773	15908	D							
PATCH18	971A	252	15921	D							
PATCH19	9732	310	15940	D							
PATCH1	9500	4482	15742	D							
PATCH20	975E	6142	15961	D							
PATCH21	978A	5983	15984	D15985							
PATCH24	97F0	1176	16028	D							
PATCH27	97FA	12288	16041	D							
PATCH2	951C	4523	15759	D							
PATCH3	953C	5068	15777	D15781							
PATCH4	9658	5151	15795	D15799							
PATCH6	9674	553	15812	D							
PATCH7	959A	3391	15830	D							
PAUSE	3558	2782	6976	D							
PAUSKEYB	3811	2779	7186	D							
PAUSKEY	3810	500	7185	D							
PCDSY	E21C	6383	D								
PERBAD	5392	11323	11341	11343	D						

CROSS REFERENCE LIST

		9188	9190	9212	9214	9226	9236	9372	11859	11978	15665	D
		15921	15940	15942								
PROGRS	D93A	293	431	434	462	454	2731	2775	2806	8519	8568	
		8607	8612	8615	8572	8679	8685	8689	9175	9234	12469	
		12471	12484	12506	12513	12535	12538	12547	12549	12566	12620	
		12636	12684	12709	12717	12734	12778	13585	15608	D		
PROGSTEP	DAB4	643	645	8884	8967	9211	15575	D				
PROGUP	144E	2523	2524	D								
PROGVP	60BC	11979	11983	D								
PROMODE	DAA2	1070	8193	8296	8312	8402	8435	8443	8544	8959	9001	
		9043	9859	9118	9124	15556	D					
PRQMV	3FF8	1168	8140	D								
PROREALT	DABC	8162	8184	15679	D							
PSHNUM	1940	3006	3008	3010	3021	D						
PSHCN	6204	12122	12124	D								
PSHDIFF	5808	10911	10917	D								
PSHFREQ	5840	11301	11311	D								
PSHIT2	5E08	11610	011620	11622								
PSHLOP	6944	12989	012993									
PSHNUMS	1C3A	3261	3265	D								
PSHREG	6962	438	1423	1444	1482	3268	3421	3602	4839	5862	8891	
		8937	9403	9846	9861	9857	9914	10019	10052	10056	10181	
		10253	10256	10372	10453	10458	10474	10515	10526	10549	10703	
		10823	10923	10986	11019	11152	11279	11310	11348	11411	11463	
		11562	11566	11570	11511	11656	11743	12066	12088	12111	12174	
		12201	12232	12330	12333	12338	12339	12340	12341	12342	12343	
		12360	12380	12418	12434	12435	12483	13002	D			
PSHRVAL	2818	4834	4839	D								
PSHSTK	6936	1402	1654	1682	1720	3029	3613	10068	10742	11416	12063	
		12127	12169	12254	12420	12488	12985	013004				
PSHWFMS	1C24	3228	3258	D								
PUD	0000	6430	D									
PURGNXT	4120	8279	D	8347	8358	8396						
PWRERR1	3402	777	6822	D								
PWRERR2	3404	821	6823	D								
PWRERR3	3406	985	6824	D								
PWRERR4	3408	1012	6825	D								
PWRSRQ	0734	778	822	986	1113	1022	D					
QUEERR	2F30	5706	D	5808								
QUEID	2E3E	5659	D	5802								
QUESRCH	25C8	4618	D	4631								
QUESRQ	2F10	5683	D	5805								
QUETABL	3000	4617	5800	D								
QUEWAIT	2586	4678	4683	D	4685							
RAMBAD	0658	858	860	873	891	899	D					
RAMBASE	A000	746	750	752	756	15710	D					
RAMFAIL	05AA	749	754	777	D							
RAMMAP	DADD	791	15694	015705								
RAMOPT	DAE4	1046	1080	12309	15705	D						
RAMTEST	05FC	751	760	854	D							
RAMTOP	D800	747	748	15711	D							
RANGE10	7612	14929	014932	14936								
RANGE1	7554	14952	014955	14959								
RDIN	20AE	3103	3169	3592	3836	D						
RDOUT	141A	354	379	457	1245	2506	D					
RDSTST	2370	4338	D	4347	4435							
RDSY	0000	6398	D	6598								
RDERR	1F6C	3583	3585	3587	3511	D						
RDTFLAG	D972	315	441	1398	1424	1457	1539	1593	1609	1656	1684	

		1685	1702	1722	1729	2528	2529	3138	3188	3189	3223
		3609	4736	4842	5095	5184	5257	5398	5424	5502	5910
		8160	8855	8912	9167	9348	9415	9805	9915	10020	10064
		10376	10513	10704	10738	10924	10949	11020	11151	11280	11412
		11559	11585	11588	11608	11618	11652	11753	11754	11930	11941
		12024	12045	12067	12099	12112	12128	12175	12227	12344	12361
		12383	12386	12399	12421	12436	12463	13512	15581	D15944	
ROTSRCH	20E4	3856	D 3873								
READCNS	2798	4778	4787	4795	D						
READERR	2740	4732	4746	D 4757	4784	4801	5122	15862	15878	15894	
READOUT	1138	2020	2057	2058	2119	D					
REALTIME	D9C6	1158	1579	1592	1508	2018	2152	2520	2749	2834	2919
		3454	8162	8184	8191	9245	12580	15514	D		
RECDON	6192	12068	D								
REFRESH	D9C4	954	1223	1945	1984	2015	3065	15613	D		
REFRSH	0F70	1923	1930	1939	1944	D					
REMEMBL	32A6	4712	4714	6275	D						
RENKEY	D928	350	356	4407	4657	15544	D				
RENLED	33CC	1850	1906	6611	D						
REMOTE	0005	6403	D 6611								
REM	3386	527	1937	4531	6548	D					
RESET	3390	1194	1202	1844	6529	D					
RESGPB	2648	384	4661	D							
RESIN	2A10	5060	5075	D15784							
RESOLV	D970	1047	1057	1063	1381	1381	1382	1588	1519	2088	2109
		2971	3124	3207	3234	3243	3244	3313	3314	3343	3347
		3353	3365	3447	3508	3517	3937	4918	5034	5300	5380
		9495	9517	9683	9811	9934	9987	10079	10107	10120	10222
		10281	10302	10340	10382	10408	10483	10558	10654	10734	10765
		10766	10768	10778	10788	10827	10886	11050	11106	11207	11380
		11451	11467	11557	11665	11592	11767	11869	11870	11898	11900
		11910	11912	11985	11991	12032	12095	12166	12196	12208	12302
		12303	12306	12310	12357	12877	13012	13034	13405	13418	13585
		13810	13883	15580	D						
RESPOS	6962	13298	13301	D							
REVERSE	786A	15198	15211	15213	15238	D					
RFDR	3385	271	321	4495	4654	4657	5623	D15816	16005		
RGPIB	0003	6401	D 6682								
RLC	337E	6620	D								
RLOK	338A	1908	6631	D							
RLTACQ	1155	2153	2158	D							
RLTRQ	1156	2150	D 2526								
RLTUP	1454	2522	2526	D							
RNDLOOP	77A2	15143	D15147								
RNGFND	758A	14953	14971	D							
ROACQEN	0017	6421	D 6605								
ROAQR	1152	2156	D 3066	3589							
RODMA	33C0	6605	D								
ROMBAD	0684	934	937	D							
ROMBASE	0000	813	15713	D							
ROMFAIL	05E3	816	821	D							
ROMSUM	0570	927	D 932								
ROMTEST	0568	814	818	924	D						
ROMTOP	4000	817	15714	D							
ROMV	3FE8	1166	8134	D							
ROOMAVL	03FB	580	585	D							
ROSTRE	2128	3870	3878	D							
ROTATE	56AA	3555	10737	10756	D						
ROUND	7795	15122	15133	15138	D						

CROSS REFERENCE LIST

ROUPT	0A18	353	1955	2508	2512	8432	15638	D				
RQSENBL	32A2	4696	4698	5581	6273)						
RQSKEY	380C	530	7183)								
RQSNUM	0AE6	290	327	346	348	352	392	448	478	480	654	
		711	770	1022	4371	4396	4398	4738	5898	5186	5259	
		5432	5728	6286	6318	8255	15708	D				
RQSOFF	3304	4400	5873	5911	5931	5313	D					
RQSOK	016E	344	348	D								
RQSON	23E0	4364	4374	D								
RQS	32AE	291	355	449	5723	5866	5914	6280	D	8266		
RRTC	0007	6405	D	6683								
RSLTNEG	6FEC	13951	13975)								
RSTBP	000C	6410	D									
RSTGPB	0100	282	294	318	D	332	15951					
RSTINT	100A	1998	2812	D								
RSTKB	0016	6458)									
RSVBUSY	3380	6296	5618	D								
RSVENBL	091E	1204	4696	4698	4700	4702	4704	4706	4708	4710	4712	
		4714	4716	4718	5583	6297	15539	D				
RSVJFF	32F0	6298	6299	6301	6303)						
RSVTBL	3294	6258	D	6293								
RSV	3385	4369	6302	5303	6314	5521)					
RS	3330	2305	6593	D								
RTIM	0008	6409	D									
RTL	337C	524	1934	6650	D							
RTN2TXT	2A7A	5119	D	5143	5158	15802						
RTWPI	E218	6381)									
RUNALL	583A	10941	D									
RUNKEYR	01CE	376	D									
RUNKEY	380E	4432	7184	D								
RUNSET	583A	10934	10936	10938	10940)						
RVSXNT	403A	8227)	8232	8235	9132	9074					
RVSTJP	40D2	8230	8237)								
ROR	E040	1889	4338	4389	6355	D						
ROW	E060	1884	6356	D								
RO	0000	141)	242	264	267	268	275	293	296	298	316
		318	330	331	385	387	394	415	416	419	420	
		422	422	423	427	433	437	462	542	563	563	
		570	570	577	577	579	586	587	588	590	618	
		621	624	625	627	640	655	661	665	666	712	
		713	718	723	724	1059	1068	1072	1073	1076	1077	
		1080	1081	1083	1085	1088	1089	1091	1298	1303	1315	
		1306	1308	1309	1346	1370	1371	1372	1374	1378	1379	
		1380	1381	1401	1422	1438	1445	1445	1483	1483	1507	
		1512	1520	1522	1533	1553	1581	1719	1889	1945	2003	
		2004	2024	2033	2033	2060	2097	2119	2159	2171	2257	
		2536	2544	2550	2563	2604	2613	2619	2641	2644	2655	
		2656	2657	2657	2558	2659	2665	2670	2673	2712	2714	
		2719	2735	2737	2742	2757	2758	2764	2768	2782	2787	
		2792	2796	2800	2804	2811	2813	2813	2828	2848	2851	
		2852	2854	2869	2878	2880	2882	2883	2885	2886	2888	
		2896	2898	2904	2923	2928	2961	2978	2979	2980	3001	
		3015	3015	3023	3028	3031	3035	3093	3104	3197	3201	
		3216	3240	3247	3258	3250	3266	3383	3395	3406	3406	
		3418	3481	3482	3485	3485	3487	3496	3548	3548	3594	
		3612	3853	3856	3858	3860	3862	3864	3867	3867	3868	
		3869	3871	3874	3883	4590	4594	4511	4752	4753	4759	
		4779	4780	4785	4808	4809	4812	4819	4830	4831	4833	
		4836	4851	4855	4861	4863	4865	4866	4868	4871	4902	

CROSS REFERENCE LIST

4905	4927	4928	4948	4949	4950	4951	4961	4962	4964			
4965	5013	5268	5269	5271	5273	5274	5275	5276	5278			
5292	5306	5311	5517	5542	5544	5554	5660	5663	5684			
5707	5710	5849	5859	5108	6114	6117	6128	6153	6154			
6179	6287	6289	6291	5292	8152	8165	8165	8177	8208			
8210	8212	8214	8228	8229	8231	8233	8234	8236	8237			
8292	8329	8335	8337	8339	8341	8344	8349	8350	8351			
8352	8421	8437	8468	8474	8476	8478	8479	8482	8484			
8486	8488	8525	8541	8554	8505	8644	8647	8712	8718			
8730	8766	8775	8779	8781	8783	8785	8794	8807	8811			
8813	8815	8821	8822	8849	8865	8869	8875	8890	8906			
8913	8914	8915	8916	8917	8921	8924	8926	8928	8938			
8936	8949	8950	8951	8952	8968	8969	8990	8999	9001			
9004	9005	9012	9014	9015	9019	9129	9030	9035	9073			
9075	9087	9100	9106	9107	9111	9114	9115	9116	9133			
9139	9141	9143	9145	9152	9170	9171	9175	9188	9191			
9192	9205	9206	9207	9226	9234	9236	9255	9256	9269			
9272	9275	9372	9374	9376	9397	9445	9449	9450	9462			
9485	9486	9495	9507	9509	9517	9521	9527	9528	9529			
9541	9569	9573	9522	9528	9645	9549	9650	9651	9661			
9662	9663	9665	9667	9668	9683	9584	9711	9815	9816			
9819	9819	9822	9824	9825	9827	9831	9844	9911	9987			
10014	10016	10055	10059	10118	10127	10129	10138	10133	10162			
10163	10165	10179	10316	10322	10337	10338	10339	10340	10341			
10342	10343	10344	10361	10362	10365	10367	10371	10382	10386			
10405	10443	10454	10454	10466	10514	10525	10560	10561	10562			
10562	10564	10573	10649	10654	10695	10697	10698	10699	10700			
10741	10827	10828	10846	10853	10854	10855	10857	10858	10886			
10887	10909	10917	10919	10919	10920	10942	10943	10944	10963			
10963	10965	10985	11008	11016	11039	11049	11060	11092	11095			
11097	11105	11106	11107	11115	11121	11128	11136	11150	11166			
11182	11209	11212	11215	11218	11219	11220	11220	11225	11240			
11247	11248	11249	11250	11251	11278	11327	11328	11329	11330			
11337	11345	11408	11415	11434	11458	11467	11470	11476	11531			
11536	11556	11557	11560	11569	11606	11610	11616	11657	11657			
11663	11664	11673	11677	11678	11689	11696	11709	11710	11714			
11717	11738	11744	11744	11755	11756	11761	11762	11773	11774			
11858	11863	11880	11915	11937	11977	11982	12080	12018	12022			
12062	12065	12087	12093	12096	12110	12126	12168	12173	12192			
12202	12202	12231	12253	12302	12305	12309	12310	12312	12314			
12328	12335	12359	12376	12396	12419	12489	12490	12495	12504			
12548	12552	12554	12588	12589	12958	12994	13005	13159	13163			
13170	13171	13172	13178	13197	13188	13191	13192	13193	13213			
13217	13221	13255	13257	13261	13275	13276	13277	13322	13323			
13330	13331	13463	13464	13465	13478	13480	13515	13516	13517			
13521	13522	13523	13524	13621	13630	13637	13647	13787	13717			
13725	13739	13940	13941	13945	13982	13985	13986	13988	13989			
13992	13994	13995	13997	13998	14040	14045	14079	14084	14093			
14105	14129	14196	14197	14198	14204	14205	14213	14214	14215			
14218	14227	14286	14293	14294	14298	14302	14305	14309	14315			
14317	14318	14319	14555	14556	14559	14561	14562	14563	14565			
14580	14612	14614	14615	14618	14816	14819	14821	14822	14825			
14828	14833	14910	14987	14995	14997	14999	15000	15012	15016			
15017	15066	15071	15071	15078	15086	15139	15157	15157	15160			
15185	15202	15252	15263	15945	15962	15963	15965	15969	15973			
R1R	EG44	526	1907	1935	4341	4355	4390	4531	6357 D			
R1W	EJ64	6358	D									
R10	000A	151	D	159	882	883	884	889	890	892	926	927
		928		930	933	2056	2054	2065	2079	2116	2508	2528

CROSS REFERENCE LIST

	2528	2530	2592	2626	2705	3794	3795	3799	3800	3804	
	3805	3809	4581	4595	4608	4900	4911	4915	4921	4925	
	4931	4935	4940	4944	4952	4956	4969	4973	4985	4989	
	4995	4999	5291	5297	5299	5305	5310	5315	5322	5329	
	5345	5349	5368	5659	5657	5683	5690	5705	5714	5829	
	5874	5894	5932	6037	11483	11484	11485	11485	11486	11487	
	11548	12667	12670	12672							
R11	000B	152 D	504	697	599	700	702	905	936	938	1029
	1157	1160	1213	1214	1216	1242	1243	1512	1513	1514	
	1522	1523	1524	1527	1531	1533	1897	1900	1901	1907	
	1908	2278	2313	2339	2352	2363	2381	2427	2478	2858	
	2859	2862	2864	2866	2873	2876	2891	2894	3124	3125	
	3127	3454	3457	3459	3451	3467	3467	3468	3469	3473	
	3477	3479	3480	3483	3493	3495	3496	3497	3673	3751	
	3756	3821	3928	4900	4915	4925	4935	4944	4956	4973	
	4989	4999	5289	5291	5299	5305	5310	5315	5322	5329	
	5345	5349	5634	5640	5641	5659	5683	5706	5829	5894	
	5957	5998	6031	6087	6098	6107	6127	6178	6209	6309	
	6319	8216	8239	8253	8333	8359	8537	8538	9618	9674	
	9675	9682	9713	9714	9793	10191	10572	10573	10574	10575	
	10581	10603	10942	11098	11205	11218	11221	11222	11226	11232	
	11236	11239	11241	11243	11252	11253	11330	11337	11522	11534	
	11872	11924	11929	11973	11976	11996	11997	12007	12036	12137	
	12210	12212	12214	12221	12671	12805	12970	13001	13003	13008	
	13016	13048	13110	13164	13166	13222	13223	13224	13238	13239	
	13240	13241	13250	13268	13270	13303	13466	13471	13483	13483	
	13491	13491	13570	13588	13589	13871	13888	13889	14226	14234	
	14336	14781	14791	14792	14832	14836	15374	15382	15401	15417	
	16007										
R12	000C	153 D	283	381	402	404	458	460	478	482	515
	607	608	615	598	701	956	988	1003	1158	1212	
	1215	1220	1221	1224	1239	1246	1583	1599	1615	1846	
	1848	1850	1899	1905	1958	2012	2037	2046	2048	2058	
	2095	2102	2104	2105	2109	2110	2111	2113	2170	2172	
	2174	2176	2179	2180	2181	2187	2188	2190	2191	2215	
	2216	2217	2218	2218	2220	2221	2222	2249	2253	2263	
	2264	2265	2266	2266	2268	2276	2277	2332	2336	2338	
	2338	2371	2373	2374	2377	2378	2407	2412	2466	3011	
	3012	3038	3039	3059	3053	3232	3237	3249	3249	3342	
	3345	3350	3357	3478	3789	3790	3791	3793	3794	3795	
	3796	3797	3800	3801	3802	3805	3806	3807	3809	3810	
	3811	3815	3816	3891	3896	3901	3902	3925	3959	3967	
	3968	3970	3988	3991	3992	3993	3994	3995	4329	4330	
	4334	4336	4338	4339	4341	4342	4345	4349	4352	4359	
	4362	4363	4365	4365	4368	4369	4370	4372	4374	4389	
	4390	4393	4401	4403	4413	4418	4453	4457	4472	4473	
	4487	4488	4495	4503	4511	4533	4568	4570	4640	4675	
	4687	5720	5724	5928	5921	5923	5924	5925	6084	6107	
	6118	6127	6157	6178	6183	6209	6223	6280	6282	6284	
	6286	6289	6293	6293	6296	6302	6303	6305	6306	6313	
	6314	6315	6316	8512	8685	9644	9545	9646	9647	9648	
	9648	9659	9671	9759	9934	9944	9960	10574	10576	10582	
	10604	11222	11223	11224	11228	11230	11233	11234	11331	11338	
	11520	11521	11522	11523	11523	11535	11861	11879	11882	11917	
	11918	11926	11927	11928	11934	11935	11980	12017	12020	12211	
	12215	12217	12221	12471	12631	12632	12634	12641	12642	12644	
	12648	12649	12651	12655	12655	12673	12681	12682	12686	12689	
	12690	12691	12693	12698	12951	12964	12985	12989	12991	12994	
	12996	12997	12998	13000	13035	13036	13037	13252	13265	13272	

		13274	13472	13486	13537	13544	13878	13880	13915	13919	13929
		13933	13972	13973	14034	14038	14073	14077	14115	14116	14196
		14218	14225	14227	14230	14231	14263	14304	14305	14306	14308
		14309	14311	14313	14313	14316	14324	14335	14650	14664	14700
		14912	14923	15010	15245	15375	15383	15402	15418	15742	15745
		15765									
R13	000D	154 D	2160	2511	2515	2532	2594	2528	2707	2756	2833
		2922	2945	3288	3319	3354	3436	3525	3838	3884	3932
		3955	4415	4558	4679	4690	4974	4990	8473	8712	9407
		9565	9616	9663	9664	9665	9665	9742	10098	10103	10151
		10162	10178	10758	10759	11037	11055	11065	11321	12464	12618
		12619	12777	12779	12833	12841	12842	12852	12861	12862	12870
		12883	12884	12892	12921	12922	12930	13063	13071	13088	13158
		13159	13160	13161	13178	13204	13221	13226	13227	13247	13317
		13319	13321	13334	13353	13351	13362	13375	13376	13395	13399
		13404	13504	13619	13521	13674	13705	13707	13780	13802	13816
		13832	13843	13854	13855	13857	13858	13920	13921	13934	13935
		13937	14006	14039	14041	14042	14078	14080	14081	14369	14528
		14658	14721	14723	14725	14909	14993	15013	15261	15310	
R14	000E	155 D	4416	4559	4680	4691	9242	9346	9346	9408	9416
		9494	9506	9614	9519	9624	9630	10100	10106	10108	10118
		10129	10172	10177	13152						
R15	000F	156 D	3969	3970	3991	4417	4560	4581	4692	9345	9355
		9355	9358	9360	9362	9409	9412	9466	9532	9543	9612
		9773	9775	9775	9784	10102	10111	10121	10156	10176	13960
		13963	13964	13973	14003	14004	14008	14009	14116	14117	14698
		14699	14700	14732	14752	14754					
R1	0001	142 D	280	285	297	299	338	340	343	348	358
		357	359	360	362	363	364	367	424	425	427
		522	525	525	526	527	545	561	589	590	591
		622	631	634	646	713	714	716	718	719	721
		747	748	755	765	786	788	790	794	813	817
		854	855	856	857	859	861	865	870	880	887
		900	924	960	961	968	969	1009	1010	1023	1024
		1026	1027	1040	1041	1042	1046	1047	1048	1048	1049
		1050	1052	1054	1074	1075	1077	1115	1116	1118	1127
		1128	1129	1130	1131	1155	1166	1167	1181	1185	1187
		1205	1206	1207	1260	1251	1262	1447	1447	1453	1485
		1486	1487	1491	1492	1502	1514	1524	1528	1528	1538
		1625	1627	1628	1638	1643	1665	1690	1696	1711	1715
		1716	1763	1766	1769	1772	1775	1780	1783	1786	1789
		1793	1797	1798	1836	1840	1852	1853	1854	1855	1857
		1859	1860	1869	1875	1882	1916	1917	1918	1919	1919
		1924	1927	1931	1932	1935	1935	1936	1937	2018	2021
		2022	2029	2031	2035	2040	2092	2099	2120	2171	2172
		2213	2215	2538	2564	2558	2502	2607	2609	2616	2618
		2659	2662	2674	2678	2715	2720	2738	2743	2760	2761
		2763	2775	2777	2784	2790	2794	2798	2802	2806	2808
		2817	2902	2919	2925	2946	2955	2955	2961	2966	2967
		2978	3005	3007	3009	3019	3022	3099	3113	3115	3120
		3122	3131	3151	3152	3153	3158	3160	3161	3167	3175
		3177	3178	3199	3200	3203	3211	3226	3227	3251	3254
		3256	3265	3267	3269	3293	3294	3295	3299	3300	3302
		3305	3306	3307	3320	3323	3324	3334	3337	3359	3415
		3419	3439	3445	3446	3491	3495	3497	3499	3503	3510
		3511	3527	3530	3531	3532	3534	3535	3551	3553	3557
		3586	3591	3595	3597	3600	3839	3840	3842	3843	3847
		3848	3855	3872	3875	3881	4538	4540	4552	4562	4668
		4684	4731	4733	4759	4786	4799	4803	4805	4819	4823

CROSS REFERENCE LIST

4837	4861	4865	4867	4869	4871	4875	4876	4877	4877
4878	4882	4883	4884	4903	4905	4918	4950	4960	4964
4968	4981	4992	5021	5027	5028	5029	5030	5042	5043
5073	5090	5092	5105	5108	5110	5126	5128	5130	5132
5156	5199	5201	5205	5206	5208	5210	5212	5214	5218
5222	5275	5277	5295	5300	5318	5325	5332	5341	5352
5354	5381	5388	5394	5442	5446	5455	5459	5460	5461
5462	5464	5466	5467	5468	5470	5510	5511	5514	5524
5531	5532	5533	5539	5547	5549	5551	5556	5560	5568
5575	5592	5636	5708	5854	5856	5857	5858	5860	5952
5991	5992	5995	6021	6022	6023	6025	6026	6027	6029
6030	6033	6035	6059	6050	6062	6063	6064	6066	6070
6074	6075	6077	6078	6079	6081	6086	6110	6114	6156
6156	6179	6213	6219	6219	8280	8281	8282	8284	8286
8287	8289	8290	8299	8300	8304	8308	8310	8312	8321
8322	8323	8325	8325	8332	8370	8371	8385	8391	8423
8424	8427	8430	8439	8440	8483	8486	8487	8489	8494
8767	8772	8790	8808	8809	8810	8811	8812	8816	8848
8853	8856	8859	8879	8889	8905	8910	8916	8918	8928
8934	8980	9041	9055	9111	9113	9114	9120	9121	9169
9173	9353	9398	9427	9429	9447	9477	9503	9505	9506
9510	9518	9525	9534	9534	9538	9538	9566	9582	9640
9642	9655	9657	9686	9687	9698	9707	9739	9741	9742
9756	9758	9759	9765	9770	9791	9809	9831	9837	9838
9839	9845	9852	9852	9855	9855	9859	9863	9865	9896
9898	9907	9912	9925	9988	9990	10008	10017	10053	10067
10067	10075	10079	10084	10085	10088	10091	10094	10095	10096
10103	10105	10106	10110	10125	10131	10141	10145	10149	10151
10154	10156	10157	10158	10150	10161	10164	10169	10180	10184
10215	10216	10221	10222	10223	10226	10242	10242	10248	10249
10254	10263	10267	10272	10283	10286	10287	10288	10289	10291
10293	10298	10299	10302	10303	10308	10309	10312	10316	10317
10328	10331	10334	10344	10345	10346	10347	10355	10358	10359
10360	10362	10367	10370	10383	10385	10386	10387	10388	10390
10397	10401	10402	10403	10412	10414	10442	10449	10460	10465
10473	10481	10484	10515	10524	10563	10564	10565	10566	10571
10580	10581	10584	10588	10593	10596	10501	10603	10606	10615
10623	10647	10701	10724	10725	10736	10758	10767	10776	10829
10830	10833	10835	10851	10851	10864	10884	10921	10933	10935
10937	10939	10944	10945	10946	10947	10950	10973	10983	11008
11009	11017	11050	11051	11059	11100	11100	11111	11116	11117
11120	11122	11131	11139	11144	11147	11149	11164	11164	11178
11181	11188	11267	11271	11277	11308	11326	11327	11331	11338
11346	11367	11370	11373	11382	11389	11395	11396	11405	11409
11433	11439	11442	11445	11451	11454	11459	11460	11490	11502
11509	11517	11534	11550	11561	11568	11662	11665	11679	11682
11686	11692	11697	11703	11708	11709	11711	11712	11751	11751
11760	11761	11764	11769	11772	11776	11781	11782	11784	11833
11837	11844	11848	11884	11972	11975	11989	12016	12058	12079
12084	12086	12089	12094	12096	12108	12124	12142	12145	12146
12147	12166	12191	12195	12199	12210	12211	12226	12238	12244
12246	12248	12250	12255	12251	12266	12271	12303	12314	12316
12317	12319	12329	12332	12336	12357	12377	12397	12468	12469
12470	12479	12481	12545	12551	12553	12621	12668	12669	12672
12675	12708	12743	12749	12758	12759	12760	12767	12772	12776
12781	12838	12858	12877	12880	12896	12902	12903	12904	12905
12906	12910	12911	12912	12913	12914	12915	12918	12931	12932
12934	12936	12937	12941	12959	13006	13035	13041	13072	13075
13076	13079	13080	13081	13092	13087	13102	13105	13108	13160

CROSS REFERENCE LIST

13184	13227	13228	13233	13234	13241	13258	13325	13353	13354
13355	13356	13356	13361	13355	13367	13368	13369	13370	13371
13372	13400	13475	13477	13478	13482	13485	13487	13505	13508
13513	13518	13519	13576	13577	13580	13583	13583	13585	13626
13627	13628	13629	13636	13643	13646	13558	13661	13667	13671
13712	13735	13738	13755	13758	13761	13761	13767	13770	13771
13773	13776	13784	13803	13806	13807	13810	13833	13836	13837
13855	13857	13873	13874	13887	13921	13922	13925	13935	13949
13959	13962	13965	13991	13994	14002	14042	14046	14048	14051
14053	14058	14062	14064	14065	14068	14081	14085	14087	14104
14107	14110	14114	14122	14132	14371	14378	14382	14384	14386
14387	14394	14394	14399	14401	14402	14404	14421	14422	14433
14434	14435	14437	14438	14448	14465	14468	14469	14470	14473
14476	14480	14492	14532	14535	14540	14541	14544	14545	14546
14551	14570	14573	14576	14579	14586	14588	14602	14608	14620
14623	14625	14627	14661	14669	14673	14677	14689	14693	14694
14695	14697	14727	14733	14733	14742	14778	14779	14787	14789
14817	14817	14825	14827	14829	14830	14831	14834	14835	14837
14848	14915	14915	14919	14925	14935	14942	14948	14958	14963
14973	14974	14977	14984	15000	15003	15008	15017	15018	15019
15023	15082	15125	15129	15131	15134	15135	15175	15181	15182
15183	15196	15200	15208	15208	15216	15232	15233	15234	15235
15240	15315	15337	15340	15346	15355	15368	15376	15382	15386
15392	15395	15399	15401	15406	15411	15420	15424	15427	15430
15474	15483	15492	15759	15762	15813	15843	15849	15860	15863
15876	15879	15892	15895	15921	15923	15940	15947	15962	15984
15990	15991	15992	15993	15996	15999	16000	16001	16006	16028
16030									
R2R	E048	359	1900	5920	6022	6059	6074	6359	D
R2W	E068	270	320	1885	4391	4494	4515	4653	4666
		15992	16004					6360	J15815
R2	0002	143	D 368	406	431	510	514	518	520
		567	568	636	637	639	640	757	758
		794	959	963	965	971	980	1122	1123
		1168	1182	1183	1184	1185	1186	1187	1451
		1504	1505	1506	1508	1517	1519	1534	1535
		1629	1637	1641	1645	1647	1648	1668	1678
		1675	1676	1677	1692	1694	1695	1699	1700
		1790	1792	1795	1800	1855	1866	1867	1873
		2010	2025	2026	2028	2030	2032	2036	2043
		2121	2173	2179	2205	2206	2213	2263	2711
		2716	2716	2720	2731	2734	2735	2738	2739
		2900	2947	2958	2962	2981	3050	3114	3116
		3126	3132	3265	3295	3297	3301	3303	3338
		3442	3443	3444	3489	3493	3494	3499	3500
		3512	3513	3549	3551	3598	3601	3843	3844
		3849	3851	3876	3879	3933	3938	4512	4562
		4566	4567	4573	4577	4586	4618	4644	4753
		4788	4781	4788	4815	4817	4818	4820	4838
		4882	4904	4907	4951	4957	4958	4968	4979
		5019	5020	5021	5022	5022	5040	5044	5188
		5269	5293	5307	5312	5316	5317	5318	5319
		5324	5325	5330	5331	5332	5333	5334	5336
		5342	5350	5351	5352	5382	5389	5429	5430
		5458	5512	5515	5525	5527	5528	5529	5530
		5552	5557	5561	5569	5575	5591	5594	5638
		5685	5687	5711	5861	5181	6211	6217	6221
		8375	8380	8388	8389	8489	8520	8585	8590
		8717	8718	8719	8721	8722	8728	8729	8730
								8731	8732

CROSS REFERENCE LIST

8768	8772	8787	8792	8795	8813	8847	8867	8888	8904
8919	8923	8934	9354	9399	9428	9431	9448	9478	9520
9521	9522	9523	9524	9588	9589	9599	9708	9743	9760
9767	9771	9792	9810	9815	9860	9866	9899	9905	9913
9926	9989	9991	10009	10018	10054	10076	10082	10086	10089
10092	10120	10122	10140	10145	10147	10219	10227	10246	10250
10255	10271	10284	10292	10293	10299	10306	10310	10313	10335
10356	10398	10400	10403	10441	10464	10482	10484	10523	10565
10567	10569	10578	10582	10591	10594	10599	10602	10604	10613
10614	10615	10616	10615	10624	10648	10702	10720	10722	10725
10726	10727	10730	10731	10734	10735	10759	10760	10762	10765
10766	10781	10782	10784	10793	10830	10831	10832	10835	10849
10850	10854	10860	10861	10862	10862	10865	10865	10922	10943
10957	10974	10984	11018	11046	11112	11123	11132	11140	11145
11148	11224	11242	11265	11266	11267	11268	11268	11272	11278
11309	11326	11347	11365	11368	11369	11370	11371	11371	11374
11397	11400	11401	11402	11403	11404	11405	11406	11406	11410
11432	11437	11440	11441	11442	11443	11443	11446	11461	11462
11488	11489	11490	11491	11491	11503	11510	11518	11535	11551
11567	11660	11662	11573	11576	11758	11762	11765	11834	11838
11845	11849	11885	11983	11985	11990	12095	12097	12109	12125
12190	12205	12229	12275	12322	12324	12325	12337	12378	12398
12482	12487	12494	12502	12551	12561	12563	12622	12667	12669
12671	12673	12679	12706	12745	12751	12768	12773	12782	12839
12854	12857	12859	12881	12897	12898	12900	12901	12902	12908
12909	12910	12916	12917	12918	12919	12919	12942	12960	13007
13082	13087	13103	13104	13105	13106	13106	13109	13161	13179
13216	13231	13295	13302	13323	13324	13325	13326	13326	13368
13372	13373	13373	13479	13480	13574	13578	13581	13628	13632
13634	13644	13659	13665	13669	13670	13671	13672	13672	13713
13714	13715	13716	13716	13722	13733	13736	13753	13756	13763
13763	13774	13775	13775	13777	13777	13782	13787	13808	13838
13858	13920	13924	13934	13941	13984	14000	14041	14049	14050
14054	14055	14061	14063	14069	14080	14088	14098	14108	14112
14121	14133	14374	14376	14379	14380	14380	14405	14455	14458
14459	14474	14477	14485	14533	14536	14549	14571	14577	14584
14589	14603	14609	14665	14667	14670	14671	14672	14675	14676
14690	14691	14728	14733	14749	14778	14785	14790	14819	14925
14931	14940	14949	14954	14961	14972	14975	14985	15019	15020
15021	15134	15141	15180	15183	15316	15339	15347	15356	15369
15377	15383	15387	15393	15396	15402	15407	15412	15421	15425
15428	15431	15475	15484	15493	15844	15846	15908	15909	16028
16029	16030	16042	16043	16044					
R3R	E04C	362	5923	6025	6062	6077	6361	D	
R3W	E05C	271	321	524	528	1194	1198	1202	1844
		1938	4344	4392	4431	4495	4654	4667	6362
R3	0003	144	D 759	761	764	758	769	771	773
		793	795	1301	1310	1312	1325	1325	1329
		1382	1396	1486	1497	1497	1499	1500	1506
		1631	1633	1634	1634	1635	1671	1672	1863
		1885	1886	1887	2032	2044	2091	2092	2093
		2122	2152	2158	2163	2155	2258	2437	2439
		2566	2575	2605	2507	2529	2642	2645	2571
		2749	2774	2834	2838	2839	2840	2843	2844
		2847	2847	2857	2860	2949	2951	2952	2952
		2964	2966	2977	2980	2983	2983	3002	3024
		3047	3049	3050	3052	3073	3089	3097	3161
		3199	3227	3230	3242	3243	3244	3245	3246
		3326	3332	3368	3369	3370	3371	3409	3422
								3438	3465

3519	3552	3553	3334	3940	3954	4513	4514	4515	4528		
4530	4531	4579	4582	4594	4596	4597	4599	4601	4603		
4605	4606	4611	4635	4648	4796	4798	4799	4810	4812		
4813	4814	5030	5031	5032	5050	5052	5075	5205	5208		
5276	5278	5379	5385	5434	5435	5436	5438	5439	5440		
5441	5441	5442	5458	5459	5466	5583	5587	8372	8373		
8374	8383	8402	8429	8430	8435	8562	8572	8577	8581		
8601	8615	8638	8540	8666	8678	8682	8688	8693	8697		
8717	8719	8720	8729	8731	8734	8735	8736	8849	8890		
8906	9353	9423	9424	9429	9431	9433	9435	9437	9459		
9466	9567	9571	9581	9626	9642	9770	9811	9832	9850		
9854	9857	9859	9920	9921	9922	9923	9923	9937	9946		
9996	10053	10075	10088	10112	10121	10125	10131	10169	10236		
10249	10254	10272	10309	10353	10395	10443	10466	10483	10485		
10525	10553	10588	10593	10626	10655	10656	10657	10658	10663		
10668	10669	10670	10671	10677	10691	10714	10716	10722	10775		
10777	10781	10783	10798	10791	10792	10799	10832	10833	10834		
10840	10842	10868	10888	10889	10893	10900	10912	10947	10948		
10951	10953	10961	10968	10969	10970	10972	10973	10988	10992		
10993	11102	11103	11132	11142	11143	11144	11145	11240	11241		
11273	11299	11340	11343	11345	11346	11347	11380	11381	11382		
11389	11392	11393	11394	11395	11434	11458	11459	11464	11466		
11468	11469	11471	11473	11475	11479	11481	11569	11693	11694		
11867	11869	11886	11984	11986	11987	11993	11995	11997	11998		
12026	12038	12082	12192	12231	12259	12264	12269	12274	12379		
12381	12490	12498	12503	12511	12511	12517	12517	12541	12541		
12543	12559	12559	12568	12568	12523	12661	12666	12713	12729		
12730	12732	12733	12735	12736	12739	12747	12749	12750	12751		
12831	12832	12837	12850	12851	12854	12955	12871	12872	12874		
12875	13074	13084	13403	13405	13419	13427	13428	13436	13444		
13445	13446	13455	13456	13464	13468	13472	13486	13487	13488		
13507	13508	13510	13518	13639	13540	13641	13642	13643	13649		
13650	13651	13652	13657	13658	13719	13721	13724	13731	13751		
13765	13765	13785	13805	13813	13814	13815	13835	13840	13841		
13842	13938	13949	13953	13971	13976	13979	13981	13985	14056		
14058	14060	14062	14090	14094	14096	14104	14422	14423	14427		
14428	14429	14430	14431	14434	14435	14436	14437	14438	14439		
14444	14445	14447	14448	14453	14454	14456	14477	14485	14488		
14492	14529	14552	14574	14574	14605	14608	14621	14621	14677		
14678	14681	14683	14730	14736	14736	14742	14746	14746	14924		
14928	14929	14931	14934	14935	14937	14938	14940	14941	14942		
14944	14945	14947	14951	14952	14954	14957	14958	14960	14961		
14962	14963	14965	14965	14971	14986	15073	15077	15091	15098		
15162	15167	15197	15204	15206	15207	15210	15212	15217	15218		
15219	15220	15223	15270	15272	15318	15380	15404	15415	15424		
15450	15451	15455	15457	15460	15045	16046	16047				
R4R	E050	1205	1842	1852	6363)					
R4W	E070	1195	1845	1882	6364)					
R4	0004	145	949	1142	1143	1153	1299	1303	1305	1317	1327
		1351	1364	1383	1387	1395	1627	1543	1953	1956	2031
		2045	2068	2069	2070	2071	2073	2074	2075	2076	2078
		2101	2123	2184	2191	2192	2193	2195	2196	2206	2220
		2224	2225	2226	2234	2236	2237	2240	2244	2245	2270
		2553	2554	2555	2567	2573	2574	2576	2581	2582	2584
		2648	2649	2651	2677	2684	2685	2689	2693	2694	2695
		2773	2816	2837	2862	2864	2950	2963	2979	3003	3025
		3033	3037	3051	3154	3057	3071	3078	3112	3119	3134
		3149	3150	3152	3154	3156	3157	3171	3173	3174	3313
		3365	3373	3377	3380	3396	3400	3401	3405	3408	3409

CROSS REFERENCE LIST

3410	3414	3447	3448	3505	3515	3517	3935	3941	3943		
3948	3950	3952	4547	4551	4552	4582	4583	4584	4585		
4585	4588	4592	4617	4620	4625	4627	4629	4630	4671		
4672	4674	4675	5032	5053	5055	5075	5380	5391	5435		
5448	5450	5581	5587	5593	8352	8353	8354	8354	8355		
8355	8356	8357	8375	8378	8382	8383	8384	8385	8389		
8390	8391	8523	8536	8543	8548	8549	8550	8551	8556		
8557	8558	8559	8561	8569	8570	8576	8579	8597	8599		
8614	8618	8635	8635	8537	8648	8552	8665	8676	8677		
8681	8687	8690	8592	8696	8761	8765	8770	8770	8848		
8889	8905	8975	8978	9020	9020	9060	9060	9354	9425		
9426	9427	9428	9434	9436	9437	9444	9445	9446	9447		
9448	9461	9462	9463	9464	9465	9482	9483	9484	9568		
9572	9582	9583	9527	9771	9814	9818	9821	9830	9831		
9851	9858	9860	9921	9924	9925	9926	9935	9936	9945		
9985	9995	10054	10075	10089	10107	10108	10109	10167	10237		
10250	10255	10268	10268	10271	10278	10278	10310	10351	10393		
10442	10465	10524	10554	10591	10594	10627	10657	10659	10661		
10670	10672	10678	10692	10757	10778	10771	10772	10777	10780		
10785	10794	10834	10843	10845	10874	10889	10890	10892	10898		
10948	10954	10955	10962	10968	10975	10976	10978	10979	10988		
10991	10993	11046	11102	11104	11115	11118	11129	11130	11131		
11223	11225	11228	11244	11246	11247	11394	11396	11399	11433		
11460	11461	11465	11468	11471	11479	11486	11500	11525	11527		
11529	11530	11532	11542	11547	11568	11654	11677	11690	11691		
11696	11767	11768	11769	11772	11782	11787	11868	11873	11917		
11926	11928	11989	11993	11995	11996	12001	12010	12826	12838		
12035	12036	12038	12040	12191	12230	12395	12396	12397	12398		
12520	12521	12522	12524	12526	12527	12528	12529	12530	12530		
12531	12546	12547	12553	12567	12571	12572	12573	12575	12578		
12596	12598	12624	12646	12659	12664	12703	12712	12713	12716		
12718	12724	12726	12740	12741	12743	12744	12745	12757	12760		
12762	12764	12766	12767	12768	12779	12780	12781	12782	13078		
13080	13084	13187	13215	13218	13226	13238	13276	13405	13411		
13426	13427	13428	13445	13447	13454	13455	13456	13462	13463		
13467	13488	13641	13644	13651	13653	13655	13659	13727	13728		
13729	13730	13730	13736	13741	13742	13743	13744	13744	13748		
13756	13948	13952	13955	13969	13975	13978	13981	13989	13991		
13998	14057	14387	14388	14389	14390	14391	14393	14396	14397		
14398	14402	14432	14530	14556	14686	14689	14679	14681	14685		
14686	14687	14688	14689	14691	14694	14731	14739	14749	14946		
14966	14967	14969	14971	15073	15074	15075	15080	15080	15083		
15084	15088	15094	15095	15117	15117	15123	15137	15153	15163		
15165	15168	15169	15173	15217	15221	15225	15236	15238	15259		
15274	15319	15378	15405	15410	15416	15425	15445	15446	15448		
15452	15453	15459	15462	15465	15467	15469	15472	15476	15479		
15486	15494	15830	15830								
R5R	E054	1196	1200	4362	4368	5313	6365	D			
R5W	E074	1199	4370	5308	6315	6366	D				
R5	0005	146	D 434	439	439	950	1154	1300	1306	1308	1327
		1328	1361	1921	1926	1928	1929	1986	1999	2021	2124
		2193	2194	2201	2212	2226	2227	2240	2413	2415	2418
		2422	2431	2432	2440	2441	2442	2443	2444	2446	2448
		2449	2450	2457	2460	2461	2474	2477	2611	2873	2892
		2894	2908	2963	2972	3055	3056	3057	3068	3069	3070
		3071	3072	3075	3076	3077	3078	3079	3080	3081	3086
		3088	3091	3092	3184	3224	3309	3366	3374	3378	3397
		3398	3402	3403	3448	3449	3506	3508	3509	3936	3941
		3943	3944	3945	3948	3950	4551	4554	4586	4590	4618

CROSS REFERENCE LIST

4622	5033	5049	5116	5124	5124	5134	5582	5592	5594
5596	8363	8364	8367	8405	8408	8409	8410	8415	8418
8419	8498	8499	8503	8505	8506	8510	8511	8515	8517
8565	8619	8643	8645	8649	8651	8667	8668	8847	8888
8904	8979	9039	9139	9044	9352	9421	9423	9433	9452
9454	9486	9633	9557	9754	9812	9813	9816	9825	9945
9946	9949	9951	9952	9954	9957	9959	9986	10011	10013
10113	10141	10187	10190	10228	10229	10230	10231	10232	10234
10290	10317	10339	10345	10350	10368	10380	10388	10400	10441
10464	10467	10469	10523	10555	10596	10601	10629	10652	10655
10668	10768	10769	10773	10778	10779	10780	10787	10796	10835
10836	10837	10839	10840	10842	10843	10845	10850	10851	10891
10906	10908	10978	10981	11103	11104	11123	11242	11243	11432
11462	11469	11473	11481	11487	11495	11526	11528	11529	11530
11533	11553	11555	11567	11884	11892	11895	11896	11920	11990
12011	12028	12035	12084	12086	12188	12109	12124	12125	12147
12190	12229	12522	12523	12524	12531	12534	12535	12618	12627
12629	12630	12631	12637	12638	12640	12641	12677	12680	12681
12688	12689	12753	12771	12777	13083	13085	13155	13162	13182
13194	13194	13212	13272	13274	13297	13297	13406	13413	13418
13420	13430	13438	13449	13731	13745	13747	13750	13751	14389
14392	14393	14399	14409	14409	14429	14460	14460	14482	14680
14682	14683	15066	15067	15058	15070	15075	15084	15114	15150
15151	15152	15154	15159	15159	15177	15252	15253	15255	15324
15470	15478	15481	15482	15483	15484				

R6R E058 6367 D

R6W E078 6368 D

R6 0005 147 J

951	1155	1323	1328	1340	1342	1351	1352	1353	
1357	1368	1440	1442	1456	1922	1924	1927	2006	2007
2009	2029	2030	2183	2199	2233	2239	2251	2252	2272
2272	2414	2416	2423	2428	2429	2430	2431	2433	2435
2438	2446	2452	2453	2454	2462	2874	2876	2890	2891
2970	2972	3082	3183	3086	3087	3088	3168	3172	3188
3183	3225	3311	3367	3375	3376	3377	3450	3478	3480
3518	3937	3946	4587	4588	4619	4620	5034	5057	5189
5217	5634	5640	5688	8398	8401	8406	8406	8416	8416
8425	8425	8524	8525	8527	8528	8529	8530	8533	8558
8564	8565	8565	8595	9610	8619	8620	8621	8622	8624
8627	8629	8633	8635	8659	8660	8661	8662	8668	8669
8670	8673	8842	8845	8853	8863	8865	8867	8873	8874
8877	8987	9051	9351	9422	9425	9434	9453	9454	9634
9752	9803	9805	9828	9828	9835	9835	9848	9848	9884
9886	9888	9894	9908	9938	9947	9970	9976	9997	10007
10114	10140	10188	10261	10262	10263	10265	10274	10275	10276
10277	10281	10288	10290	10294	10296	10300	10325	10327	10331
10381	10398	10404	10446	10449	10451	10475	10498	10493	10494
10496	10497	10503	10505	10507	10508	10556	10599	10602	10631
10653	10688	10690	10707	10709	10728	10728	10770	10772	10775
10783	10784	10785	10787	10791	10792	10793	10794	10796	10892
10894	10898	10899	10904	10912	10913	10945	11004	11007	11813
11013	11040	11042	11095	11105	11129	11206	11211	11216	11219
11249	11328	11454	11536	11663	11885	11903	11908	11908	11921
11991	12008	12031	12032	12039	12043	12137	12148	12199	12224
12225	12226	12630	12577	13147	13152	13168	13168	13189	13189
13243	13243	13253	13253	13407	13411	13436	13447	13622	13626
13639	13649	13708	13713	13727	13741	14189	14117	14120	14436
14443	14444	14462	14479	14480	14489	14545	14547	14549	14552
14553	14554	14561	14566	14569	14570	14615	15090	15093	15096
15114	15115	15121	15132	15231	15249	15250	15251	15255	15258

CROSS REFERENCE LIST

		15260	15267	15269	15278	15279	15280	15281	15325	15326	15449
		15451	15453	15456	15457	15480	15481	15485	15486	15488	15490
R7R	E05C	267	4393	4823	5073	5156	5991	6369	015742	15759	15993
R7W	E07C	4356	4394	6030	6070	6086	6370	D			
R7	0007	148	605	611	753	753	762	762	815	815	819
		819	865	867	867	870	872	872	880	883	887
		890	897	899	924	928	933	935	937	1509	1989
		1994	1995	1997	2041	2059	2071	2074	2076	2078	2185
		2214	2228	2229	2231	2270	2304	2305	2307	2308	2309
		2312	2340	2341	2347	2350	2353	2355	2356	2360	2362
		2365	2367	2368	2369	2370	2371	2378	2379	2417	2418
		2419	2420	2433	2434	2435	2438	2439	2442	2443	2445
		2448	2450	2452	2456	2457	2465	2466	2468	2470	2471
		2473	2474	2971	2975	3113	3120	3131	3206	3208	3234
		3235	3236	3237	3343	3344	3345	3347	3348	3349	3350
		3353	3354	3355	3355	3357	3372	3381	3640	3641	3643
		3644	3647	3648	3649	3552	3655	3556	3664	3665	3722
		3723	3725	3726	3727	3730	3733	3734	3744	3745	3745
		3747	3748	3787	3789	3816	3817	3820	3892	3893	3898
		3900	3910	3911	3912	4415	4558	4579	4690	5048	5049
		5050	5052	5053	5055	5830	5831	5835	5837	5840	5845
		5867	5870	5895	5896	5908	5902	5905	5915	5928	5947
		5951	5952	5093	6097	6288	6291	6300	8248	8251	8497
		8501	8513	8525	8531	8534	8559	8617	8618	8621	8623
		8625	8630	8632	8534	8636	8661	8553	8679	8710	8711
		8716	8727	8732	8735	9009	9010	9245	9492	9515	9523
		9652	9653	9672	9758	9936	9937	9940	9942	9943	9952
		9957	9974	9988	9990	10158	10164	10183	10186	10187	10282
		10287	10296	10326	10327	10328	10457	10460	10467	10469	10476
		10490	10494	10495	10497	10504	10506	10508	10557	10563	10572
		10612	10613	10617	10519	10620	10771	10775	10790	10799	10893
		10894	10899	10900	10913	10950	10951	10953	10954	10956	11041
		11042	11049	11060	11095	11115	11118	11119	11122	11207	11208
		11214	11217	11221	11248	11464	11502	11541	11654	11886	11922
		11992	12001	12005	12087	12027	12028	12029	12030	12040	12208
		12219	12580	12625	12695	12696	12696	12701	12703	12753	12754
		12756	12758	12770	12771	12797	12799	12801	12804	12963	12965
		12968	12969	12969	12988	12992	13011	13013	13034	13037	13041
		13042	13043	13045	13163	13173	13174	13176	13179	13188	13193
		13213	13215	13216	13219	13224	13228	13261	13294	13467	13468
		13469	13475	13547	13548	13551	13552	13571	13572	13577	13578
		13623	13624	13625	13525	13527	13640	13650	13709	13710	13711
		13711	13714	13728	13742	13870	13874	13876	13879	13882	13884
		14159	14163	14165	14168	14170	14172	14179	14181	14183	14185
		14186	14199	14202	14203	14208	14211	14212	14219	14221	14224
		14229	14233	14258	14261	14254	14266	14268	14271	14275	14277
		14281	14291	14296	14298	14300	14306	14310	14311	14312	14315
		14322	14325	14326	14329	14332	14333	14334	14335	14400	14406
		14481	14493	14554	14565	14567	14571	14618	14813	14815	14827
		14838	14838	15091	15092	15094	15097	15098	15148	15155	15230
		15233	15236	15237	15332	15334	15336	15343	15346	15349	15349
		15372	15389	15458	15463	15489	15491				
R8	0008	149	606	607	513	615	866	868	871	874	881
		885	888	893	900	902	925	931	2065	2066	2067
		2070	2075	2079	2083	2084	2085	2085	2087	2088	2089
		2090	2093	2094	2167	2168	2175	2178	2203	2242	2258
		2252	2274	2275	2305	2307	2310	2312	2342	2344	2345
		2346	2347	2348	2348	2349	2350	2357	2358	2359	2360
		2370	2372	2373	2376	2377	2406	2408	2409	2411	2417

CROSS REFERENCE LIST

SET256	5323	12247	12261	D								
SET512	6344	12249	12266	D								
SFALSE	1770	2810	2813	D								
SFNEXT	215A	3903	3905	D	3926							
SHFTVZR	7A8E	15459	D15464									
SHIFTKEYB	37FB	7174	D 8323		9241							
SHIFTKEY	37FA	7173	D									
SHIFT1	6E70	13762	13764		13767	D13789						
SHIFT2	6E98	13768	13782	D								
SIGNDON	4ECC	9836	9843	D								
SIGNLOP	4E80	9814	D 9833									
SIGNOK	721E	14270	14276		14279	14285	D					
SINCON	4EF2	9853	9856		9859	D						
SKIPKEY	25B2	4598	4600		4602	4604	4605	D				
SKIPSCL	43C6	9574	9582	D								
SLWSHFT	55CA	10763	10764		10767	D						
SMAX	3416	6719	D15360		15361	15445						
SMTHCNT	516A	10170	D10173		10174							
SMTEND	5170	10168	10172	D								
SMT4IT	510C	10119	10125	D								
SMTHLOP	5124	10138	D10171									
SMTHQUT	5176	10124	10175	D								
SOFTST	33DE	237	240		1045	6377	D 9401	9610				
SOFT	0009	158	D 237		240	1045	1051	1052	1137	1138	1139	1140
		1141	1143		1144	1145	1314	1315	1316	1317	1319	1321
		1323	1324		1354	1355	1356	1357	1361	1364	1368	1369
		1373	1374		1375	1376	1379	1381	1384	1385	1387	1389
		1391	1392		1394	1395	1450	1451	1452	1453	1454	1455
		1488	1489		1490	1491	1500	1502	1503	1631	1632	1645
		1646	1668		1669	1592	1593	1714	1715	2160	2161	2162
		2163	2164		2165	2166	2168	2169	2174	2300	2301	2302
		2303	2330		2331	2333	2335	2337	2340	2342	2343	2355
		2357	2361		2362	2364	2355	2366	2406	2407	2408	2409
		2411	2412		2415	2416	2424	2459	2460	2463	2464	2511
		2516	2532		2555	2556	2557	2558	2570	2571	2583	2584
		2585	2586		2594	2595	2596	2597	2598	2600	2614	2628
		2631	2632		2633	2634	2636	2650	2651	2652	2653	2658
		2660	2661		2664	2665	2681	2682	2695	2696	2697	2698
		2707	2723		2724	2725	2726	2756	2833	2851	2852	2853
		2854	2855		2856	2857	2865	2866	2877	2878	2879	2880
		2881	2882		2883	2884	2885	2895	2896	2897	2898	2899
		2900	2901		2902	2903	2904	2906	2922	2945	2949	2958
		2969	2970		3021	3022	3047	3049	3068	3075	3096	3097
		3112	3119		3133	3134	3143	3144	3145	3146	3149	3154
		3155	3157		3158	3164	3155	3171	3172	3174	3175	3182
		3190	3191		3192	3193	3195	3205	3206	3229	3230	3232
		3233	3253		3254	3262	3253	3288	3291	3319	3321	3326
		3336	3337		3342	3354	3364	3368	3411	3412	3413	3414
		3419	3420		3436	3437	3438	3441	3442	3458	3459	3460
		3461	3462		3463	3464	3465	3466	3469	3472	3512	3525
		3528	3556		3557	3598	3591	3593	3600	3601	3603	3604
		3605	3606		3671	3672	3738	3739	3749	3750	3752	3753
		3754	3755		3787	3820	3838	3839	3845	3846	3850	3851
		3853	3854		3878	3879	3880	3881	3882	3883	3884	3891
		3892	3895		3899	3900	3904	3905	3910	3915	3917	3924
		3927	3932		3933	3951	3952	3953	3954	3955	4974	4978
		4979	4980		4981	4982	4983	4990	5015	5016	5017	5018
		5029	5039		5040	5041	5042	5048	5056	5100	5101	5102
		5103	5107		5110	5111	5112	5136	5137	5138	5140	5335

CROSS REFERENCE LIST

5336	5337	5338	5341	5342	5383	5384	5385	5386	5388
5389	5506	5507	5508	5519	5520	5521	5522	5523	5535
5536	5537	5538	5539	5540	5564	5565	5566	5567	5571
5572	5573	5574	5585	5586	5589	5590	5635	5636	5637
5638	5851	5852	5856	6210	6211	6212	6213	6214	6215
6217	6218	8473	8475	8476	8477	8478	8578	8579	8580
8581	8598	8599	8600	8601	9144	9145	9146	9147	9249
9250	9251	9252	9253	9254	9256	9257	9258	9349	9350
9351	9352	9398	9399	9401	9402	9405	9407	9408	9409
9473	9474	9475	9475	9487	9488	9498	9499	9500	9501
9503	9504	9511	9565	9575	9576	9577	9578	9579	9580
9610	9611	9612	9613	9614	9615	9615	9617	9618	9644
9652	9658	9659	9661	9669	9671	9674	9692	9693	9694
9695	9701	9702	9703	9704	9710	9744	9745	9746	9747
9751	9752	9753	9754	9761	9762	9763	9764	9766	9767
9791	9792	9904	9905	9906	9907	9912	9913	9965	9966
9967	9968	9971	9972	9973	9974	9977	9978	9979	9980
9992	9993	9994	9995	9996	9999	10001	10011	10012	10069
10070	10071	10072	10081	10082	10083	10084	10091	10092	10097
10098	10099	10100	10101	10102	10176	10177	10178	10218	10219
10220	10221	10226	10227	10231	10233	10234	10236	10237	10238
10245	10246	10247	10248	10254	10265	10266	10267	10276	10277
10305	10306	10307	10308	10312	10313	10314	10334	10335	10350
10351	10352	10353	10355	10356	10373	10374	10392	10393	10394
10395	10397	10398	10510	10511	10618	10619	10620	10621	10623
10624	10625	10626	10627	10628	10629	10630	10631	10632	10635
10636	10639	10640	10642	10643	10644	10645	10647	10648	10681
10682	10683	10684	10685	10688	10689	10686	10687	10688	10689
10864	10865	10866	10867	10868	10870	10871	10872	10873	10874
10876	10877	10879	10880	10881	10882	10884	10885	10895	10896
10901	10902	10903	10904	10906	10907	10914	10915	10977	10979
10980	10981	10983	10984	11017	11018	11037	11057	11058	11065
11108	11109	11111	11112	11139	11140	11147	11148	11185	11186
11187	11188	11204	11205	11252	11271	11272	11303	11304	11305
11306	11308	11309	11321	11364	11365	11366	11367	11373	11374
11375	11385	11386	11387	11399	11402	11409	11410	11414	11436
11437	11438	11439	11445	11446	11492	11493	11494	11495	11497
11498	11499	11500	11506	11507	11511	11514	11515	11519	11538
11539	11540	11541	11544	11545	11546	11547	11548	11553	11554
11688	11705	11757	11758	11759	11760	11764	11765	12089	12090
12091	12093	12322	12323	12464	12465	12466	12467	12470	12491
12492	12493	12497	12498	12500	12501	12520	12544	12571	12583
12584	12585	12586	12587	12589	12590	12591	12619	12705	12706
12707	12708	12720	12721	12722	12730	12733	12772	12773	12803
12804	12833	12834	12835	12836	12837	12841	12842	12852	12853
12855	12856	12857	12861	12862	12870	12873	12874	12875	12876
12883	12884	12892	12893	12894	12895	12903	12906	12911	12915
12921	12922	12930	12938	12939	12940	12941	13002	13003	13008
13039	13040	13046	13047	13063	13064	13065	13066	13067	13071
13072	13074	13075	13076	13078	13079	13088	13099	13100	13101
13102	13108	13109	13158	13200	13201	13202	13203	13204	13230
13231	13232	13233	13245	13247	13317	13318	13319	13320	13321
13331	13334	13362	13363	13364	13365	13366	13371	13375	13376
13395	13399	13400	13402	13403	13404	13504	13515	13521	13525
13569	13570	13573	13574	13575	13576	13580	13581	13588	13619
13662	13663	13664	13665	13666	13667	13674	13705	13759	13760
13772	13773	13780	13788	13802	13803	13805	13806	13814	13815
13816	13832	13833	13835	13836	13841	13842	13843	13854	13870
13871	13872	13873	13877	13882	13887	13888	13937	13938	13940

CROSS REFERENCE LIST

13956	13958	13965	13966	13968	13969	13970	13971	13984	14000
14001	14002	14005	14007	14007	14039	14078	14090	14097	14098
14101	14111	14112	14113	14114	14123	14126	14126	14131	14134
14159	14163	14165	14173	14174	14177	14181	14187	14190	14195
14215	14222	14223	14224	14228	14232	14233	14258	14261	14262
14274	14284	14285	14286	14330	14331	14332	14369	14371	14374
14375	14403	14404	14405	14406	14407	14411	14415	14416	14417
14418	14421	14463	14468	14471	14471	14483	14490	14490	14528
14529	14530	14531	14535	14536	14581	14582	14583	14584	14585
14586	14588	14589	14590	14591	14592	14595	14595	14597	14598
14599	14605	14606	14611	14612	14624	14625	14626	14627	14658
14661	14666	14671	14696	14697	14721	14722	14723	14724	14725
14727	14728	14730	14731	14780	14781	14782	14783	14784	14785
14786	14787	14789	14790	14791	14909	14910	14913	14915	14917
14918	14919	14925	14925	14927	14948	14949	14958	14972	14978
14988	14993	14994	15001	15012	15006	15007	15010	15011	15012
15013	15021	15022	15115	15119	15120	15128	15130	15131	15138
15143	15144	15145	15148	15149	15151	15155	15170	15171	15172
15175	15176	15179	15180	15182	15187	15189	15190	15192	15201
15210	15212	15214	15215	15216	15222	15223	15224	15225	15230
15242	15244	15245	15246	15248	15249	15259	15260	15261	15265
15266	15270	15271	15272	15273	15276	15277	15279	15310	15311
15312	15313	15314	15318	15319	15320	15321	15322	15324	15325
15327	15328	15329	15330	15332	15333	15335	15336	15343	15344
15348	15357	15358	15359	15360	15361	15363	15364	15365	15365
15368	15369	15370	15371	15372	15374	15375	15376	15377	15378
15379	15380	15386	15387	15388	15395	15396	15397	15417	15418
15419	15438	15431	15432	15433	15434	15436	15437	15441	15442
15443	15445								

SORTED	18C0	2958	D										
SOVRFLW	7ADA	15354	15488	D									
SPACBYT	3433	2468	6748	D	8535	8549	8550	8551	8561	8614	8637	8681	
		8687	8696		12494	12598	12627						
SPACEB	3320	4356	4394		5105	5201	5218	5446	5464	5470	6110	6327	D
SPACES	3430	3143	6746	D	10699	10919							
SPACE	3432	6747	12493										
SPACOUT	64F6	12492	12496	D									
SPASIN	2336	4343	4362	D									
SPAS	337C	4342	4366		6651	D							
SPCSTR	3226	6111	6114	D									
SPEAKER	33BE	6604	13544	D									
SPEC.FND	41E8	8340	8343	D									
SPEC.LOP	4108	8336	8342	D									
SPEC.PROG	37DA	7158	8335	D									
SPEC.TST	41CA	8307	8309		8311	8324	8327	8331	D				
SQRNRM	6E7A	13766	13772	D									
SQRREST	6E1C	13727	13740	D									
SQRTA	341A	6730	14687	D									
SQRTE	341C	6731	14679	D									
SQRTE	341E	6732	14678	D									
SQRTO	751A	14662	14665		14698	D							
SRCHLNN	4702	8920	8925	D	8931								
SRQLED	33CA	1848	4372		6305	6316	6610	D					
SRQMSG	2F2C	5684	5692	D									
SRQS	3386	4363	6647	D									
SRQTEND	2ECC	5596	5618	D									
SRQ	0015	6419	6610	D									
STADSY	0001	6399	6599	D									
STARTUSER	DFA0	6670	6671	D	6672	6673	6674	6675	6676				

CROSS REFERENCE LIST

XSTK	3306	238	241	2734	6573	12957	12962	12967	12995	12999	13507
XWFM	3532	2742	6966	D							
XYGRS	1000	2082	D								
XYWFM	D978	1087	1438	1715	1727	2025	2080	2564	2616	2674	11749
		12345	15584	D							
YSTCK	361A	2714	6958	D							
YSTK	3308	2711	6674	D							
YTCRS	100C	2081	2087	D							
YTWFM	D97A	1092	1093	2028	3309	3310	3311	3312	3366	3367	12257
		12258	12262	12263	12267	12268	12272	12273	15585	D	
YWFM	3526	2719	6962	D							
YZFLAG	DA0C	4740	4994	5013	15632	D					
YZ1	DA0E	4992	5018	15633	D						
YZ2	DA10	4993	5016	15634	D						
Y	3328	4928	5306	6336	D						
ZCNSWFM	305C	5850	5860	D							
ZEROCNS	304E	5846	5849	D							
ZEROEXP	7222	14267	14286	D							
ZEROSCL	7AAE	15351	15474	D							
ZEROWFM	3052	5077	5848	5851	D						
ZEROWD	6974	4791	4848	13011	D						
ZMAX	3418	6726	15366	D							
ZNRNLSL	7244	14287	14302	D							
ZNRHLSR	7230	14288	14293	D							
ZNRMLZO	7288	14280	14290	14292	14297	14299	14301	14323	14328	14330	D
ZNRMOVR	7278	14303	14314	14324	D						
ZROREF	67E6	1417	2545	2924	11521	12829	D				
ZTRAIL	7806	15178	15185	D							

0 LINE(S) WITH ERRORS
2092 SYMBOLS

0 LINE(S) WITH WARNINGS
15494 REFERENCES

54318 MILLISECOUNDS USED