

WVS*FILE(7).V76

```

1 * V75 EMULATION MICROPROGRAM. 001
2 * ENTER AT X'00 IF TESTING IN PAGE 1. 002
3 * 003
4 * REGISTERS USED: 004
5 * 0 = 620 A REGISTER 005
6 * 1 = 620 B REGISTER 006
7 * 2 = 620 X REGISTER 007
8 * 3-7 = GENERAL PURPOSE REGISTERS 008
9 * 8 = USED BY HALT LOOP TO HOLD INSTRUCTION REGISTER. THE 009
10 * CONSOLE IS WIRED TO REQUEST DISPLAY OF REGISTER B WHEN 010
11 * THE INSTRUCTION REGISTER KEY IS PRESSED. 011
12 * 9 = I/O DEVICE SELECTION REGISTER (LOW ORDER 6 BITS) 012
13 * E = USED FREELY AS WORKING SPACE, AND REQUIRED BY MULTIPLY 013
14 * AND DIVIDE INSTRUCTIONS. 014
15 * F = USED FREELY AS WORKING SPACE, AND REQUIRED BY MULTIPLY 015
16 * AND DIVIDE INSTRUCTIONS. 016
17 * 017
18 A EQU X'A HEXADECIMAL 018
19 B EQU X'B DIGITS 019
20 C EQU X'C 020
21 D EQU X'D 021
22 E EQU X'E 022
23 F EQU X'F 023
24 DISMP EQU 0'100745 *EXC DISMP INSTR 024
25 * 025
26 * SPECIAL REGISTERS. 026
27 * 027
28 IH EQU 8 INSTRUCTION REGISTER (HALT LOOP) 028
29 DR EQU 9 I/O DEVICE REGISTER 029
30 S1 EQU E 030
31 S2 EQU F 031
32 * 032
33 * INTERRUPT MASKS. 033
34 * 034
35 M EQU E ALLOW I/O ONLY IF M/P 035
36 I EQU F ALLOW I/O ALWAYS 036
37 N EQU C DO NOT ALLOW I/O 037
38 * 038
39 * PAGE NUMBER. 039
40 * 040
41 P EQU 512*1 041
42 * 042
43 * MACRO TO ORIGIN AT NEXT MULTIPLE OF P(1). 043
44 * 044
45 MORG MAC 045
46 ORG **P(1)-1 046
47 ORG */P(1)*P(1) 047
48 EMAC 048
49 * 049
50 * MACRO FOR ILLEGAL INSTRUCTION INTERRUPT ENTRY. 050
51 * 051
52 ILLAD EQU 2 ILLEGAL INSTRUCTION INTERRUPT 052
53 ILLOP MAC 053
54 GMSK /N(ILLOP1),1(1),IMF,LA1,IB3,RF1, START I/O @ 4, 054
55 CFF9,MK1 P-2->P 055
56 EMAC 056

```

531
723-4

```

57      EJEC                                057
58      *  INTERRUPT SERVICE.                058
59      *  THE STATES WHICH ENABLE THE DECODER ALSO ENABLE INTERRUPTS.  IF 059
60      *  A SINGLE WORD INSTRUCTION WAS AT THE INTERRUPT ADDRESS, THE NEXT 060
61      *  WORD IS IN THE INSTRUCTION BUFFER BUT SHOULD NOT BE PROCESSED. 061
62      *  PROCESSING OF THIS WORD IS INHIBITED BY TRANSFERRING TO INTBAS 062
63      *  IF CINTF=1 AND IM=6.  IF CINTF=0 AND NO INTERRUPT IS PENDING, THE 063
64      *  INSTRUCTION IS DECODED.  IF AN INTERRUPT IS PENDING AND CINTF=0 064
65      *  OR IM NOT =6, CONTROL GOES TO INTBAS+7, WHERE THE INTERRUPT 065
66      *  MICROPROGRAM IN I/OCS IS INITIATED.  IF THE INTERRUPT WASN'T 066
67      *  REALLY AN INTERRUPT, BUT INSTEAD WAS A DMA REQUEST, INTERRUPT 067
68      *  PROCESSING MUST BE ABORTED AND THE PIPELINE RESTARTED.  IF IT 068
69      *  REALLY IS AN INTERRUPT, CONTROL GOES TO INTBAS+1 WHERE INTERRUPT 069
70      *  SERVICING IS COMPLETED.          070
71      *                                     071
72      MORG 16  INTERRUPT HARDWARE USES INTBAS, INTBAS+1, INTBAS+7 072
73      INTBAS EQU *                           073
74      MORG 2   /T(SWAS0,SS2MI)                074
75      SS2MI GEN  /P(SS3MI+P),TF0,SF2,GF4,IM8,RF4  (AA)->ALU,IF(P),INCP 075
76      *                                     PAGE JUMP (NOP IN PG 0) 076
77      ORG   INTBAS+1 INTERRUPT SECOND TRY COMES HERE 077
78      INT1  GEN  /N(INT2),IM2,RF1,FFF,AAE      WAIT(10),RE-1->P 078
79      SWAS0 GEN  /F(STA1-2),2(6),FSF,SF1,IME   FSEL(13-12),OS(MIL) 079
80      ABORT  GMSK /N(ABORT1),IM4,LB2,FFA,MKO   I->IBR 080
81      ABORT1 GMSK /N(SS2MI),IM4,LB2,FFA,MKO   I->IBR 081
82      *      HLT1 AND HLT7 ARE REFERENCED WITH INTERRUPTS ENABLED. 082
83      HLT1  GMSK /N(HLT2),1(3),AB2,IM4,LB3,FFA.  START I/O AT X'E. 083
84      *      CMKEFCO X'103F->ALU 084
85      HLT7  GEN  /N(HLT1),LB1,FFA,MF1,WR1,24(IH) DOR->IH 085
86      ORG   INTBAS+7 INTERRUPT FIRST TRY COMES HERE 086
87      IWAIT GEN  2(ABORT),1(7),MT1,GF4,MR1,IME.  ENABLE INTERRUPTS, 087
88      *      CLAI,RF1,FFF,WR1,AAE  DISABLE DEC, START IO, P-1->P,RE 088
89      ILLOP1 GMSK /N(ILLOP2),IM2,LB3,FFA.      WAIT(10) 089
90      *      CIS(-1-DISMP) EXC DISMP -> ALU 090
91      ILLOP2 GMSK /N(INT2),SF1,IM4,LB3,RF3.    ILLAD->DOR,IF(ALU) 091
92      *      IS(-1-ILLAD) 092
93      INT2  GEN  /N(INT3),SF1,IM4,LB1,RF3,FFA.  IF(ALU),IOR->DOR 093
94      *      CMF1,BB2 094
95      INT3  GEN  /N(INT4),SF1,IM4,LB1,FF9,CF3.  IF(ALU),DOR+1->RF 095
96      *      CWR1,SH1,BB0,AAF 096
97      INT4  GEN  GF5,IM7,FFA,MF1,WR1,BBF,AAE   IBR->I,DECODE, 097
98      *      1->CINTF,S2->S1 098
99      EJEC                                099
100     *  V75 CONDITIONAL JUMP INSTRUCTIONS.  THESE ARE DIFFERENT FROM 100
101     *  VARIAN STANDARD INSTRUCTIONS.  THE JUMP IF EVEN PARITY AND JUMP 101
102     *  IF ODD PARITY INSTRUCTIONS ARE NOT IN VARIANS SET. 102
103     *                                     103
104     JI1  GEN  /F(JISEL),2(7),FS7,GF2,IM1,RF3,  FSEL(5-3),WAIT(M), 104
105     *      C23(*-*)  BB->DOR,SAMPLE 105
106     MORG 2   /T(JP5,*) 106
107     JP4  GEN  /T(JP5,*),TF2,GF2,MR1,RF5,FF9.  DOR(RC),DOR(1)->BB(0), 107
108     *      CWR1,SC1,WF1,BBE,AAF  INCS,RD+(RE OR RF)->RD 108
109     JP5  GEN  /F(JRN),2(1),FS7,GF2,LA3,SH1,AA2 FSEL(3),RD(RC),SAMPLE 109
110     MORG 8   REF BY FSEL(5-3) FROM JI1 110
111     JISEL EQU * 111
112     JEP  GEN  /N(JP1),FF0,WR1,SH1,AAE 0->RE 112
113     JOP  GEN  /N(JP1),FF0,WR1,SH1,AAE 0->RE 113

```

114	JRZ	GEN	/T(JMP1,SS1M),TF2,SF3,GF9,IMC	CND(ALUZ) IF(MIL), TT	114
115	JRNZ	GEN	/T(JMP1,SS1M),TF3,SF2,GF9,IMC	CND(-ALUZ) IF(MIL), TF	115
116	JRN	GEN	/T(JMP1,SS1M),TF2,SF3,GF7,IMC	CND(ALUS) IF(MIL), TT	116
117	JRP	GEN	/T(JMP1,SS1M),TF3,SF2,GF7,IMC	CND(-ALUS) IF(MIL), TF	117
118	JDPZ	GEN	/S(JDPZ1,SS1M),2(1),FS6,TF2,GF9	ALUZ=>FSEL(2)	118
119	JDPNZ	GEN	/S(JDPZ1,JMP1),2(1),FS6,TF2,GF9	ALUZ=>FSEL(2)	119
120	JP1	GEN	/N(JP2),FF0,CF3,WR1,SH1,AAF	1->RF	FREE 120
121	JP2	GMSK	/N(JP3),LB3,RF2,FFA,MKFFF1	-14->SC	FREE 121
122	JP3	GEN	/N(JP4),MR1,FF0,CF3,WR1,SC1,WF1,	DOR(RC),DOR(1)->BB(0),	FREE 122
123			CSH1,BBE,AAD	1->RD	123
124		EJEC			124
125	*		INPUT DATA TRANSFER INSTRUCTIONS.		125
126	*				126
127	IABM2	GEN	/N(IABM3),1(3),IMF,LB1,FFA,MF1,	START I/O AT X'C,	FREE 127
128			CWR1,BB0,24(S1)	DOR->S1	128
129	IABM3	GMSK	/F(IME1),2(7),FSA,IM2,LB2,FFE,	FSEL(8-6),WAIT(10),	FREE 129
130			CMKFE00,16(S1)	IR & X'1FF V S1	130
131	INAB2	GEN	/N(IAB),SF1,IM8,LB1,RF7,FFE,MF1,	IF(P),INCP,	FREE 131
132			CWR1,BB0,AA0	DOR V A -> A,DOR	132
133	IAB	GEN	/N(IWAIT),1(M),GF5,IM6,LB1,FFA,	IBR->I,DEC,INTRPT,	FREE 133
134			CMF1,WR1,BB0,AA1	DOR->B	134
135	IME2	GEN	/N(SS2M),SF1,IM8,RF4,LB1,FFA,MF1	IF(P),INCP,DOR->ALU	FREE 135
136		MORG	8 REF BY FSEL(8-6)		136
137	IME1	GEN	/N(IME2),SF1,IME,LB1,RF3,FFA,MF1,	OS(MIL),IOR->DOR,	137
138			CXF1,BB2	RESET CINTF	138
139	INA	GEN	/N(SS3M),SF1,IM8,LB1,RF4,FFE,MF1,	IF(P),INCP,IOR V A -> A	139
140			CWR1,BB2,AA0		140
141	INB	GEN	/N(SS3M),SF1,IM8,LB1,RF4,FFE,MF1,	IF(P),INCP,IOR V B -> B	141
142			CWR1,BB2,AA1		142
143	INAB1	GEN	/N(INAB2),LB1,RF3,FFE,MF1,BB2,AA1	IOR V B -> DOR	143
144		ILLOP			144
145	CIA	GEN	/N(SS3M),SF1,IM8,LB1,RF4,FFA,MF1,	IF(P),INCP,IOR->A	145
146			CWR1,BB2,AA0		146
147	CIB	GEN	/N(SS3M),SF1,IM8,LB1,RF4,FFA,MF1,	IF(P),INCP,IOR->B	147
148			CWR1,BB2,AA1		148
149	CIAB	GEN	/N(IAB),SF1,IM8,LB1,RF7,FFA,MF1,	IF(P),INCP,IOR->A,DOR	149
150			CWR1,BB2,AA0		150
151		EJEC			151
152	*		DIVIDE INSTRUCTION.		152
153	*				153
154		MORG	2 /T(DIV2,DIV1)		154
155	DIV1	GEN	/N(DIV2),FF6,CF3,WR1,SH1,BBF,AAF	-RF->RF	155
156	DIV2	GEN	/T(DIV4,DIV3),TF2,GFA,FF2,CF3,	DSB->DIV4,-RF->RE	156
157			CWR1,SH1,BBF,AAE		157
158		MORG	2 /T(DIV4,DIV3)		158
159	DIV3	GEN	/N(DIV7),RF3,AA1	B->DOR	159
160	DIV4	GEN	/N(DIV5),LA2,WR1,AA1	B(L)->B,0->DLAD	160
161		MORG	4 BE CAREFUL: /S(DIV13&DIV14,DIV12)	SELECTS DSB	161
162	DIV13	GEN	/S(SS3M,DIV15),2(8),FSF,TF3,SF2,	FSEL(15) IF NO QS	162
163			CGFF,IM8,RF4	IF(P) IF NO QS, INCP	163
164	DIV5	GEN	/N(DIV6),GF2,RF3,FF2,CF3,SH1,BB1	-B->DOR,SAMPLE	FREE 164
165		ORG	DIV13+2 REF BY FSEL(DSB)		165
166	DIV14	GEN	/S(SS3M,DIV15),2(8),FSF,TF2,SF3,	FSEL(15) IF QS,	166
167			CGFF,IM8,RF4,FF6,CF3,WR1,SH1,BB0,	IF(P) IF QS, INCP	167
168			CAAD	-A->A	168
169	DIV6	GEN	/N(DIV7),FF2,CF1,WR1,SC1,WF1,SH1	DOR(RT),CARRY-A->A	FREE 169
170		MORG	4 /T(DIV9,DIV10), /T(DIV11,DIV10)		170

171	DIV10	GEN	/T(DIV11,*),TF2,GFC,MR1,LA2,RF5,	DIVIDE STEP		171
172			CFF9,WR1,SC1,XF2,SH2,BBF			172
173	DIV11	GEN	/S(DIV13,DIV12),2(6),FS2,TF3,GFD,	FSEL(DSB) IF RO(15)=0		173
174			CLB1,FFA,MF1,WR1,BB0,AA1	DOR->B		174
175	DIV9	GEN	/N(DIV10),SF1,GF2	SET OVERFLOW		175
176	DIV7	GEN	/N(DIV8),GF2,FF9,WR1,SC1,XF2,BBE	FIRST DIVIDE STEP	FREE	176
177		MORG	2 /S(DIV13,DIV12)			177
178	DIV12	GEN	/T(DIV14,DIV13),TF2,GFA,FF9,WR1,	DSB=>DIV14,A+RF->A		178
179			CBBF,AA0			179
180	DIV8	GMSK	/T(DIV9,DIV10),TF3,GFD,LB3,RF2,	RO(15)=>DIV10,-14->SC	FREE	180
181			CFFA,MKE			181
182		EJEC				182
183	*		INSTRUCTIONS FORCED TOGETHER BY TEST ADDRESSING			183
184	*					184
185	SRE81	GEN	/N(SS2M),SF1,IM8,RF4,XF2	IF(P),INCP,J=SIG	FREE	185
186	HFPT1	GEN	/N(HMEN),IM2,LB1,FFA,MF1	WAIT(10),DOR->ALU	FREE	186
187		MORG	4 /T(JMP2,SS2M), /T(SRE80,SS2M)			187
188	SS2M	GEN	/N(SS3M),SF1,IM8,RF4,FF0,24(==)	IF(P),INCP,(AA)->ALU		188
189	JMP2	GEN	/N(JMP1),SF1,IMC,RF5	IF(MIL),INC SC		189
190	SRE80	GEN	/N(SRE81),LB1,RF1,FF9,CF3,SH1	DOR+1->P		190
191	HMEN1	GEN	/N(HMEN2),IM2	WAIT(10)	FREE	191
192	HMEN2	GEN	/F(HREN),2(3),FSA,LB1,RF3,FFA,	FSEL(7=6),IOR->DOR	FREE	192
193			CMF1,BB2			193
194	HSTT7	GEN	/N(HSTT8),GF1,24(IH)	IH->ALU,IBR->IR	FREE	194
195	HSTT8	GEN	/N(HSTT9),SF1,IM8	IF(P)	FREE	195
196	HSTT9	GEN	/N(IWAIT),1(4),GF5	IBR->I,DECODE,ETC	FREE	196
197		EJEC				197
198	*		SINGLE WORD ADDRESSING INSTRUCTIONS.			198
199	*		PART OF DOUBLE-WORD EXTENDED INSTRUCTIONS.			199
200	*					200
201		MORG	32 REF BY FSEL(15-13)			201
202	LDA1	GEN	/N(LDB),SF1,AB3,IM8,RF4,BB1,AA0	INCP,IF(P),SET AA&BB		202
203	ELDA1	GEN	/N(IWAIT),1(M),GF5,LB1,FFA,MF1,	IBR->I,DEC,INTRPT,	FREE	203
204			CWR1,BB1,AA0	MIL->A		204
205		MORG	2 LDB,LDX REF BY FSEL(12)			205
206	LDB	GEN	/N(IWAIT),1(I),GF5,IM6,LB1,FFA,	IBR->I,DEC,INTRPT,		206
207			CMF1,WR1,BB1,AA1	SEL&RESET CINTF,MIL->B		207
208	ELDB1	GEN	/N(IWAIT),1(M),GF5,LB1,FFA,MF1,	IBR->I,DEC,INTRPT,	FREE	208
209			CWR1,BB1,AA1	MIL->B		209
210		ORG	LDA1+4			210
211	SWA31	GEN	/F(LDB),2(4),FSE,SF1,IM8,RF4	FSEL(12),IF(P),INCP		211
212	ELDX1	GEN	/N(IWAIT),1(M),GF5,LB1,FFA,MF1,	IBR->I,DEC,INTRPT,	FREE	212
213			CWR1,BB1,AA2	MIL->X		213
214		ORG	LDB+4			214
215	LDX	GEN	/N(IWAIT),1(I),GF5,IM6,LB1,FFA,	IBR->I,DEC,INTRPT,		215
216			CMF1,WR1,BB1,AA2	SEL&RESET CINTF,MIL->X		216
217	EINR1	GEN	/N(INR2),GF8,LB1,FF9,CF3,SH1,BB1	MIL+1->ALU,SAMPLE OVFL	FREE	217
218		ORG	SWA31+4			218
219	INR1	GEN	/N(EINR1),SF1,IM6,LB1,FFA,MF1	OS(ALU),DOR->ALU		219
220	INR2	GEN	/F(SS3M),2(8),FSF,SF1,IM8,RF4,	FSEL(14),MIL+1->ALU,	FREE	220
221			CLB1,FF9,CF3,SH1,BB1	INCP,IF(P)		221
222		MORG	2 SUB,ANA REF BY FSEL(12)			222
223	SUB	GEN	/N(IWAIT),1(I),GFD,IM6,LB1,FF6,	IBR->I,DECODE,EN INT,		223
224			CCF3,WR1,BB1,AA0	SAMPLE OVFL,A=MIL->A		224
225	ESUB1	GEN	/N(IWAIT),1(M),GFD,LB1,FF6,CF3,	IBR->I,DEC,INTRPT,	FREE	225
226			CWR1,BB1,AA0	A=MIL->A,SAMPLE OVFL		226
227		ORG	SUB+2			227

228	ANA	GEN	/N(IWAIT),1(I),GFS,IM6,LB1,FFB.	IBR->I,DECODE,EN INT,	228
229			CMF1,WR1,BB1,AAO	A&MIL->A	229
230	EANA1	GEN	/N(IWAIT),1(M),GFS,LB1,FFB,MF1.	IBR->I,DEC,INTRPT,	FREE 230
231			CWR1,BB1,AAO	A&MIL->A	231
232		MORG	2 /S(LDA1,SWA26)		232
233	SWA26	GEN	/N(SWA22),SF1,IMD,RF5	OF(MIL),INCS	233
234	EORA1	GEN	/N(IWAIT),1(M),GFS,LB1,FFE,MF1.	IBR->I,DEC,INTRPT,	FREE 234
235			CWR1,BB1,AAO	A V MIL -> A	235
236		ORG	INR1+8		236
237	ORA1	GEN	/N(ORA2),SF1,IM8,RF4	IF(P),INCP	237
238	ORA2	GEN	/N(IWAIT),1(I),GFS,IM6,LB1,FFE.	IBR->I,DEC,INTRPT,	FREE 238
239			CMF1,WR1,BB1,AAO	A V MIL -> A	239
240		MORG	2 ADD, ERA REF BY F*SEL(12)		240
241	ADD	GEN	/N(IWAIT),1(I),GFD,IM6,LB1,FF9.	IBR->I,DECODE,EN INT,	241
242			CWR1,BB1,AAO	SAMPLE OVFL,A+MIL->A	242
243	EADD1	GEN	/N(IWAIT),1(M),GFD,LB1,FF9,WR1.	IBR->I,DEC,INTRPT,	FREE 243
244			CBBI,AAO	A+MIL->A,SAMPLE OVFL	244
245		ORG	ORA1+4		245
246	SWA35	GEN	/F(ADD),2(4),FSE,SF1,IM8,RF4	FSEL(12),IF(P),INCP	246
247	SWA22	GEN	/S(LDA1,SWA26),2(C),MT1,FSF,TF3.	C.FSEL(15-13)(MIL15=0),FREE	247
248			CGFB,LB1,RF3,FFA,MF1,BB1	MIL->OPR	248
249		ORG	ADD+4 MORG 16 WITH ADD		249
250	ERA	GEN	/N(IWAIT),1(I),GFS,IM6,LB1,FF6.	IBR->I,DECODE,EN INT,	250
251			CMF1,WR1,BB1,AAO	A XOR MIL -> A	251
252	EERA1	GEN	/N(IWAIT),1(M),GFS,LB1,FF6,MF1.	IBR->I,DEC,INTRPT,	FREE 252
253			CWR1,BB1,AAO	A XOR MIL -> A	253
254		ORG	SWA35+4		254
255	SWA36	GEN	/F(SUB),2(2),FSF,SF1,IM8,RF4	FSEL(12),IF(P),INCP	255
256	SWA21	GEN	/N(SWA22),SF1,IMD	OF(MIL)	FREE 256
257		MORG	2 REF BY FSEL(12)		257
258	MULO	GEN	/N(MUL),LB1,FFA,MF1,WR1,BB1,AAF	MIL->S2	258
259	MUL	GMSK	/T(MUL4,MUL1),TF2,GFD,LB3,RF2.	ROSIGN=>MUL4,15->SC	FREE 259
260			CFFA,MKOOOE		260
261		ORG	SWA36+4		261
262	SWA37	GEN	/F(MULO),2(4),FSE,IM1	FSEL(12),WAIT(M)	262
263	DORTOP	GEN	/N(RESET),RF1,LB1,FFA,MF1,BBD	DOR->P FOR DEBUGGING	FREE 263
264		ORG	MULO+4 MORG 16 WITH MULO		264
265	DIVD	GEN	/N(DIV),LB1,FFA,MF1,WR1,WF1,BB1.	MIL->S2,MIL15->QS	265
266			C24(S2)		266
267	DIV	GEN	/T(DIV2,DIV1),TF3,GFF,VF1,AAO	QS=>DIV1,R0(15)->DSB	FREE 267
268		EJEC			268
269	*		SHIFT INSTRUCTIONS AND INSTRUCTIONS CONSTRAINED TO BE		269
270	*		NEAR SHIFT INSTRUCTIONS BY ADDRESSING.		270
271	*				271
272		MORG	32 SHIFTS REF BY CND FSEL(8-5), FAIL TO SS3MI		272
273	ASLB	GEN	/T(*,SS3MI),TF3,SF3,GFC,IM8,LA2,RF5,WR1,SH4,AA1		273
274	LRLB	GEN	/T(*,SS3MI),TF3,SF3,GFC,IM8,LA2,RF5,WR1,SH1,AA1		274
275	ASRB	GEN	/T(*,SS3MI),TF3,SF3,GFC,IM8,LA3,RF5,WR1,SH2,AA1		275
276	LSRB	GEN	/T(*,SS3MI),TF3,SF3,GFC,IM8,LA3,RF5,WR1,SH4,AA1		276
277	ASLA	GEN	/T(*,SS3MI),TF3,SF3,GFC,IM8,LA2,RF5,WR1,SH4		277
278	LRLA	GEN	/T(*,SS3MI),TF3,SF3,GFC,IM8,LA2,RF5,WR1,SH1		278
279	ASRA	GEN	/T(*,SS3MI),TF3,SF3,GFC,IM8,LA3,RF5,WR1,SH2		279
280	LSRA	GEN	/T(*,SS3MI),TF3,SF3,GFC,IM8,LA3,RF5,WR1,SH4		280
281	LASL	GEN	/N(LASL1),LA2,RF3,VF1,AA1		281
282	LLRL	GEN	/N(LLRL1),RF3,AA1		282
283	LASR	GEN	/N(LASR1),LA2,RF3,VF1,AA1		283
284	LLSR	GEN	/N(LLSR1),RF3,AA1		284

285		MORG	4 REF BY FSEL FROM RR4		285
286	ROF	GEN	/N(SS3MI),SF1,GF4,IM8,RF4	RESET OVFL, IF(P), INCP	286
287	SOF	GEN	/N(SS3MI),SF1,GF2,IM8,RF4	SET OVFL, IF(P), INCP	287
288	TSA	GMSK	/N(TSA1),1(3),AB2,IME,LB3,FFA,	START IO	288
289			CMKDFCO		289
290		ILLOP			290
291		MORG	16 REF BY CND FSEL FROM DIV, FSEL FROM MUL, INR		291
292	SS3M	GEN	/N(IWAIT),1(M),GF5,IM6	IBR->I, DECODE, ETC	292
293	SHFT1	GEN	/S(ASLB,SS3MI),2(F),FS9,TF3,SF3,	CND(NOT ALUZ) FSEL(8-5)FREE	293
294			CGF9,IM8,LB1,RF2,FF6,CF3,SH1,BB0	CND(ALUZ) IF(P),	294
295	*			-DOR->SHIFT COUNTER	295
296		MORG	2 /T(LASL2,*)		296
297	LASR1	GEN	/T(LASL2,*),TF2,GFC,LA3,RF5,WR1,SC1,WF1,XF1,SH2		297
298	LLSR1	GEN	/T(*,LASL3),TF3,SF3,GFC,IM8,LA3,RF5,WR1,SC1,		298
299			CWF1,XF1,SH4		299
300		MORG	2		300
301	LASL1	GEN	/T(LASL2,*),TF2,GFC,LA2,RF5,WR1,	SHFT OVFL => LASL2	301
302			CSC1,XF3,SH6,AA0	(A,DOR)LEFT, INC SC	302
303	LASL2	GEN	/N(LASL3),SF1,IM8,SC1,WF1,XF3	IF(P),DOR RT,DSB->DOR15	303
304		MORG	2 /T(*,LASL3)		304
305	LASL3	GEN	/N(IWAIT),1(I),GF5,IM6,LB1,FFA,	DECODE,IBR->I,EN INT,	305
306			CMF1,WR1,AA1	DOR->B	306
307	LLRL1	GEN	/T(*,LASL3),TF3,SF3,GFC,IM8,LA2,	SHFT OVFL=>LASL3,INC SC	307
308			CRF5,WR1,SC1,XF1,SH2	(A,DOR) LEFT ROTATE	308
309		ORG	SS3M+8 REF BY FSEL, MUST BE MORG 32 WITH SHIFTS		309
310	SS3MI	GEN	/N(IWAIT),1(I),GF5,IM6	IBR->I, DECODE, ETC	310
311	HLT6	GEN	/F(HRDP),2(7),FSA	FSEL(8-6)	FREE 311
312		MORG	2 /S(SS3M,DIV15)		312
313	DIV15	GEN	/F(SS3M),2(8),FSF,SF1,IMA,FF6,	FSEL(14),IF(P),	313
314			CCF3,WR1,SH1,BB1,AA1	-B->B	314
315		EJEC			315
316	*	JUMP	A/D MARK INSTRUCTION.		316
317	*				317
318	JMRK3	GEN	/N(SS2M),SF1,IM4,LB1,RF1,FF9,CF3,	IF(ALU),MIL+1->P	FREE 318
319			CSH1,BB1		319
320		MORG	2 /T(JMRK1,JMRK5)		320
321	JMRK5	GEN	/N(SS2M),SF1,IM8,RF4	IF(P),INCP	321
322	JMRK1	GEN	/T(JMRK2,JMRK4),TF3,SF2,GFB,IM2,	MIL15=>JMRK4,P+1->ALU,	322
323			CLA1,CF3	CND(MIL15=0) OS(MIL)	323
324		MORG	2 /T(JMRK2,JMRK4)		324
325	JMRK4	GEN	/N(JMRK1),SF1,IMD,RF5	OF(MIL),INC SC	325
326	JMRK2	GEN	/N(JMRK3),IM1,LA1,CF3	WAIT(M),P+1->ALU	326
327		EJEC			327
328	*		DOUBLE WORD IMMEDIATE INSTRUCTIONS. NOTE THAT THESE INSTRUCTIONS		328
329	*		ARE DIFFERENT FROM VARIAN'S SINCE AN M-FIELD EQUAL TO ZERO OR 8		329
330	*		DOES NOT CAUSE UNPREDICTABLE RESULTS. IN ADDITION, BITS 7, 1		330
331	*		AND 0 ARE USED SIMILARLY TO THE USAGE FOR DOUBLE-WORD EXTENDED		331
332	*		ADDRESSING INSTRUCTIONS. WHEN BITS 1 AND 0 ARE EQUAL TO 00, 01		332
333	*		OR 10, INDEXING AND INDIRECT ADDRESSING ARE PERFORMED. THUS,		333
334	*		NEGATIVE OPERANDS ARE NOT POSSIBLE. WHEN BITS 1 AND 0 ARE EQUAL		334
335	*		TO 11, THE ACTION IS THE SAME AS VARIAN'S ACTION. NOTE THAT THE		335
336	*		VARIAN ASSEMBLER ASSEMBLES BITS 1 AND 0 AS 00, NOT 11. THIS		336
337	*		PROGRAM MAY BE MADE TO EXECUTE PROGRAMS SO ASSEMBLED BY		337
338	*		EXCHANGING IMMREL(X*168) AND IMMNI(X*16B), AND BY EXCHANGING		338
339	*		IMRL AND IMNI BELOW.		339
340	*				340
341		MORG	32 REF BY CND FSEL, FAIL TO IMIND		341

```

342 IMMBAS EQU * 342
343 * THE FOLLOWING INSTRUCTION IMPLEMENTS AN INDEXED SHIFT CAPABILITY. 343
344 * BITS 8-5 OF THE ADDRESS WORD SHOULD BE SET AS FOR SHIFT 344
345 * INSTRUCTIONS. NOTE THAT ERRATIC RESULTS OCCUR IF THE CONTENTS 345
346 * OF BITS 8-5 > X'B. SINCE THIS INSTRUCTION CANNOT DO 0 => CINTF, 346
347 * ERRATIC RESULTS WILL OCCUR IF INDEXED SHIFTS ARE THE TARGET 347
348 * OF INTERRUPTS. INTERRUPTABILITY IS AS FOR SHIFTS. 348
349 IXSHFT GMSK /N(SHFT1),TF0,SF2,GF3,IM8,RF7, IF(P),INCP,IBR->I 349
350 CFF7,MK001F DOR&X'1F->DOR,SAMPLE 350
351 * NEXT 8 INSTRUCTIONS ALL DO: 0 => CINTF. 351
352 ILDA GEN /N(SS2M),LB1,RF4,FFA,MF1,WR1,XF1, INCP, DOR->A 352
353 CBBO,AAO 353
354 ILDB GEN /N(SS2M),LB1,RF4,FFA,MF1,WR1,XF1, INCP, DOR->B 354
355 CBBO,AAO 355
356 ILDX GEN /N(SS2M),LB1,RF4,FFA,MF1,WR1,XF1, INCP, DOR->X 356
357 CBBO,AAO 357
358 IINR GEN /F(IST2),2(1),FS1,GF8,LB1,RF3, FSEL(CINTF), DOR+1->DOR 358
359 CFF9,CF3,XF1,SH1,BBO SAMPLE OVERFLOW 359
360 ISTA GEN /F(IST2),2(1),FS1,RF3,XF1,AAO FSEL(CINTF),A->DOR 360
361 ISTB GEN /F(IST2),2(1),FS1,RF3,XF1,AA1 FSEL(CINTF),B->DOR 361
362 ISTX GEN /F(IST2),2(1),FS1,RF3,XF1,AA2 FSEL(CINTF),X->DOR 362
363 ISTZ GEN /F(IST2),2(1),FS1,RF3,XF1,SH1 FSEL(CINTF),0->DOR 363
364 * NEXT 7 INSTRUCTIONS ALL DO: 0->CINTF, INCP 364
365 IORA GEN /N(SS2M),LB1,RF4,FFE,MF1,WR1,XF1, A V DOR -> A 365
366 CBBO,AAO 366
367 IADD GEN /N(SS2M),GF8,LB1,RF4,FF9,WR1,XF1, A+DOR->A,SAMPLE OVFL 367
368 CBBO,AAO 368
369 IERA GEN /N(SS2M),LB1,RF4,FF6,MF1,WR1,XF1, A XOR DOR -> A 369
370 CBBO,AAO 370
371 ISUB GEN /N(SS2M),GF8,LB1,RF4,FF6,CF3,WR1, A-DOR->A,SAMPLE OVFL 371
372 CXF1,BBO,AAO 372
373 IANA GEN /N(SS2M),LB1,RF4,FFB,MF1,WR1,XF1, A&DOR->A 373
374 CBBO,AAO 374
375 IMUL GEN /N(MUL),LB1,RF4,FFA,MF1,WR1,XF1, DOR->RF 375
376 CBBO,AAF 376
377 IDIV GEN /N(DIV),LB1,RF4,FFA,MF1,WR1,WF1, DOR->RF, ALU15->QS 377
378 CXF1,BBO,AAF 378
379 MORG 16 REF BY CND FSEL, FAIL TO IMIND 379
380 IMXC EQU * IMMEDIATE INSTRUCTION INDEX COMPUTATION 380
381 * NEXT 3 INSTRUCTIONS DO C.FSEL(6-3) (QS=0), C.OF(A) (QS=1) 381
382 IMRL GEN /S(IMMBAS,IMIND),2(F),FS7,TF3, P+DOR->DOR 382
383 CSF3,GF3,IM5,LA1,LB1,RF3,FF9,BBO 383
384 IMIX GEN /S(IMMBAS,IMIND),2(F),FS7,TF3, X+DOR->DOR 384
385 CSF3,GF3,IM5,LB1,RF3,FF9,BBO,AA2 385
386 IMIB GEN /S(IMMBAS,IMIND),2(F),FS7,TF3, B+DOR->DOR 386
387 CSF3,GF3,IM5,LB1,RF3,FF9,BBO,AA1 387
388 IMNI GEN /F(IMMBAS),2(F),FS7,LB1,RF3,FFA, MIL->DOR, FSEL(6-3) 388
389 CMF1,BB1 389
390 MORG 2 REF BY FSEL(7) 390
391 IMPRE GEN /F(IMXC),2(3),FS4,LB1,RF3, FSEL(1-0), 391
392 CFFA,MF1,WF1,BB1 MIL->DOR,MIL15->QS 392
393 IMPOST GEN /S(IMXC,IMPST1),2(3),FS4,TF3,GF3, C.FSEL(1-0) (MIL15=0), 393
394 CLB1,RF3,FFA,MF1,BB1 MIL->DOR 394
395 MORG 2 /T(IMPST1,IMIND1) 395
396 IMIND1 GEN /S(IMMBAS,IMIND),2(F),FS7,TF3, C.FSEL(6-3) (MIL15=0), 396
397 CSF3,GF3,IM5,LB1,RF3,FFA,MF1,BB1 C.OF(A) (MIL15=1), 397
398 * MIL->DOR 398

```

399	IMPST1	GEN	/S(IMXC,IMIND),2(3),FS4,TF3,SF3,	C.FSEL(1-0) (MIL15=0),	399
400			CGFB,IM5,LB1,RF3,FFA,MF1,BB1	C.OF(A) (MIL15=0),	400
401	*			MIL->DOR	401
402		MORG	2 /S(IMXC,IMIND)		402
403	IMIND	GEN	/T(IMPST1,IMIND1),TF3,GFF,IM1,RF5	WAIT(M),QS=>IMIND1,INCS	403
404	IST4	GEN	/N(SS3M),SF1,IM8,LB1,RF4,FFA,MF1	IF(P),INCP,DOR->ALU	FREE 404
405		MORG	2 /S(IMXC,IMPST1)		405
406	IMPST1	GEN	/N(IMIND),SF1,IM5,LB1,FFA,MF1,BB0	OF(ALU),DOR->ALU	406
407	RR1	GEN	/F(RR2),TS3,2(3),FSA,MRO,AB1,LA3	I(5-3)->BB,FSEL(7-6),	FREE 407
408	*			GET BYTA READY FOR SBT1	408
409		MORG	2 REF BY FSEL(CINTF)		409
410	IST2	GEN	/N(IST3),SF1,IM6,AAF	OS(ALU),RF->ALU	410
411	IST1	GEN	/N(IST3),SF1,IMA	OS(P)	411
412		EJEC			412
413	*		V75 REGISTER - REGISTER AND BYTE INSTRUCTIONS.		413
414	*				414
415	LBT2	GEN	/N(LBT3),GF2,IM1,LA3,SH1,24(*-*)	AA(RC),SAMPLE,WAIT(M)	FREE 415
416	LBT3	GEN	/F(LBT4),2(1),FS2,SF3,GF7,IM8,	FSEL(BYTA),MIL->DOR,	FREE 416
417			CRF7,LB1,BB1	CND(ALUS) IF(P), INCP	417
418		MORG	16 REF BY FSEL WITH REG SEL FROM RR1.		418
419	RR2	GEN	/F(RR5),2(1),FS9,RF3,FF0,24(*-*)	AA->DOR,FSEL(5)	419
420	RRAD	GEN	/N(SS3MI),SF1,GF6,IM8,RF4,FF9,	IF(P),INCP,AA+BB->AA,	420
421			CWR1,23(*-*),24(*-*)	SAMPLE OVERFLOW	421
422	RRSB	GEN	/N(SS3MI),SF1,GF6,IM8,RF4,FF6,	IF(P),INCP,AA-BB->AA,	422
423			CCF3,WR1,23(*-*),24(*-*)	SAMPLE OVERFLOW	423
424	RRT	GEN	/N(SS3MI),SF1,IM8,RF4,FFA,MF1,WR1	IF(P),INCP,BB->AA	424
425		MORG	2 REF BY FSEL(5) FROM RR2		425
426	RR5	GEN	/F(RR4),2(3),TS0,FS7,MRO,AB2,BB0	FSEL(4-3),I(2-0)->AA	426
427	RR3	GEN	/F(LDR1-4),2(7),TS0,FS7,SF1,IM8,	I(2-0)->AA,1->BB,	427
428			CRF7,MRO,AB2,XF1,BB1,AA0	IF(P),INCP,FSEL(5-3),	428
429	*			IS=1,RO->DOR,0->CINTF	429
430		MORG	2 /T(LBT1,LBSBI1),/T(SBT1,LBSBT1)		430
431	LBSBI1	GEN	/N(LBSBI2),SF1,IMD,AB3	OF(MIL),HOLD AA&BB	431
432	LBSBI2	GEN	/F(LBT1),2(1),FS7,AB3,IM1	FSEL(3),WAIT(M),HOLD	432
433		MORG	2 REF BY FSEL(3) FROM LBSBI2		433
434	LBT1	GEN	/N(LBT2),SF1,IM5,AB3,LB1,LA3,FF9,	IF(ALU),MIL+AA(RA)	434
435			CSH2,23(*-*),24(*-*)	HOLD AA&BB	435
436	SBT1	GEN	/F(SBT2),2(1),FS2,SF1,IM7,LB1,	FSEL(BYTA),BS(ALU).	436
437			CLA3,FF9,SH2,23(*-*),24(*-*)	MIL+AA(RA)	437
438		MORG	2 REF BY FSEL(BYTA) FROM SBT1		438
439	SBT2	GEN	/N(IST4),RF7,LB1,FFA,MF1,BB7	ORLS->DOR,INCP	439
440	IST3	GEN	/N(IST4),LB1,RF4,FFA,MF1,BB0	INCP,DOR->ALU	440
441		MORG	2 REF BY FSEL(BYTA) FROM LBT3		441
442	LBT4	GEN	/N(LBT5),SF1,IM8,RF3,LB1,FFA,MF1,	OLSE->DOR,IF(P)	FSEL 442
443			CBB5		443
444	LBT5	GEN	/N(IWAIT),1(M),GF5,IM6,LB1,FFA,	ORZF->RO,IBR->I,DECODE,FSEL	444
445			CMF1,WR1,BB6,AA0	SEL&RST CINTF,INTRPT	445
446	OME2	GEN	/N(OAR1),IM1	WAIT(M)	FREE 446
447	OABM2	GMSK	/F(OME3),2(3),FSA,IM2,LB2,FFE,	FSEL(7-6),WAIT(IO),	FREE 447
448			CMKFEO0,16(S1)	I&X'1FF V S1 -> ALU	448
449		MORG	16 REF BY FSEL WITH REG SEL FROM RR5.		449
450	RR4	GEN	/F(ROF),2(3),FS4	FSEL(1-0)	450
451	INCR	GEN	/N(SS3MI),SF1,GF6,IM8,RF4,LB1,	IF(P),INCP,DOR+1->AA,	451
452			CFF9,WR1,SH2,23(*-*),24(*-*)	SAMPLE OVERFLOW	452
453	DECR	GEN	/N(SS3MI),SF1,GF6,IM8,RF4,LB1,	IF(P),INCP,DOR-1->AA,	453
454			CFF9,CF3,WR1,SH2,23(*-*),24(*-*)	SAMPLE OVERFLOW	454
455	COMR	GEN	/N(SS3MI),SF1,IM8,RF4,FF0,MF1,WR1	IF(P),INCP,NOT AA->AA	455


```

456          ORG      RR4+4      REF BY FSEL(5=3) WITH REG SEL FROM RR3. I(5)=1.      456
457  LDR1      GEN      /N(SS3M),SF1,IM8,RF4,LB1,FFA,MF1, IF(P),INCP,MIL->AA      457
458          CWR1,23(*-*),24(*-*)      458
459  ADR1      GEN      /N(SS3M),SF1,GF6,IM8,RF4,LB1,FF9, IF(P),INCP,MIL+AA->AA,      459
460          CWR1,23(*-*),24(*-*)      SAMPLE OVERFLOW      460
461  LBT       GEN      /T(LBT1,LBSB11),TF3,GF8,AB3      MILS=>LBSBTI,HOLD AA&BB      461
462  SBT       GEN      /T(SBT1,LBSB11),TF3,GF8,AB3,LA1, MILS=>LBSBTI,HOLD AA&BB      462
463          CRF1,FFF      P-1->P (FOR MPLE)      463
464          EJEC      464
465  *          OUTPUT DATA TRANSFER INSTRUCTIONS.      465
466  *          466
467          MORG      4      REF BY FSEL      467
468  OME1      GEN      /N(OME2),SF1,IMD,XF1      OF(MIL),RESET CINTF      468
469  OAR1      GEN      /N(OABM2),1(7),IMF,LB1,FFA,MF1,      START I/O AT X'1C,      469
470          CWR1,BB0,24(S1)      DOR->S1      470
471  OBR1      GEN      /N(OABM2),1(7),IMF,LB1,FFA,MF1,      START I/O AT X'1C,      471
472          CWR1,BB0,24(S1)      DOR->S1      472
473  OAB1      GEN      /N(OABM2),1(7),IMF,LB1,FFA,MF1,      START I/O AT X'1C,      473
474          CWR1,BB0,24(S1)      DOR->S1      474
475          MORG      4      REF BY FSEL      475
476  OME3      GEN      /N(SS1M),IM2,LB1,RF4,FFA,MF1,BB1      WAIT(IO),INCP,MIL->ALU      476
477  OAR2      GEN      /N(SS2M),IM2,AA0      WAIT(IO),A->ALU      477
478  OBR2      GEN      /N(SS2M),IM2,AA1      WAIT(IO),B->ALU      478
479  OAB2      GEN      /N(SS2M),IM2,FFE,MF1,BB1,AA0      WAIT(IO),A V B->ALU      479
480          EJEC      480
481  *          BIT TEST INSTRUCTION.      481
482  *          482
483          MORG      16      REF BY FSEL(3=0)      483
484  BT20      GMSK      /F(BT30),2(1),FS8,LB3,RF3,FFA,MKFFFE      484
485  BT21      GMSK      /F(BT30),2(1),FS8,LB3,RF3,FFA,MKFFFD      485
486  BT22      GMSK      /F(BT30),2(1),FS8,LB3,RF3,FFA,MKFFFB      486
487  BT23      GMSK      /F(BT30),2(1),FS8,LB3,RF3,FFA,MKFFF7      487
488  BT24      GMSK      /F(BT30),2(1),FS8,LB3,RF3,FFA,MKFFEF      488
489  BT25      GMSK      /F(BT30),2(1),FS8,LB3,RF3,FFA,MKFFDF      489
490  BT26      GMSK      /F(BT30),2(1),FS8,LB3,RF3,FFA,MKFFBF      490
491  BT27      GMSK      /F(BT30),2(1),FS8,LB3,RF3,FFA,MKFF7F      491
492  BT28      GMSK      /F(BT30),2(1),FS8,LB3,RF3,FFA,MKFEFF      492
493  BT29      GMSK      /F(BT30),2(1),FS8,LB3,RF3,FFA,MKFDFF      493
494  BT2A      GMSK      /F(BT30),2(1),FS8,LB3,RF3,FFA,MKFBFF      494
495  BT2B      GMSK      /F(BT30),2(1),FS8,LB3,RF3,FFA,MKF7FF      495
496  BT2C      GMSK      /F(BT30),2(1),FS8,LB3,RF3,FFA,MKEFFF      496
497  BT2D      GMSK      /F(BT30),2(1),FS8,LB3,RF3,FFA,MKDFFF      497
498  BT2E      GMSK      /F(BT30),2(1),FS8,LB3,RF3,FFA,MKBFFF      498
499  BT2F      GMSK      /F(BT30),2(1),FS8,LB3,RF3,FFA,MK7FFF      499
500          MORG      2      REF BY FSEL(4)      500
501  BT30      GEN      /F(BT40),2(1),FS9,GF2,LB1,FFB,MF1      501
502  BT31      GEN      /F(BT40),2(1),FS9,GF2,LB1,FFB,MF1,AA1      502
503          MORG      2      REF BY FSEL(5)      503
504  BT40      GEN      /T(BT50,SS1M),TF3,SF2,GF9,IMC      504
505  BT41      GEN      /T(BT50,SS1M),TF2,SF3,GF9,IMC      505
506          EJEC      506
507  *          MULTIPLY INSTRUCTION.      507
508  *          508
509          MORG      2      /T(MUL4,MUL1)      509
510  MUL1      GEN      /N(MUL2),GF2,FF9,BBF,AA1      RF+B->ALU,SAMPLE      510
511  MUL4      GEN      /N(MUL5),LA2,RF3,VF1,AA1      B(L)->DOR,B15->DSB      511
512          MORG      2      /T(MUL3,MUL5)      512

```

513	MUL5	GEN	/N(MUL7),MR1,FF3,MF1,WR1,SC1.	ZERO -> RE, DOR(RT),	513
514			CWF1,XF1,AAE	DOR01 -> BADD	514
515	MUL3	GEN	/N(MUL5),GF8,FF9,BBF,AA1	RF+B->ALU,SAMPLE OVFL	515
516	MUL2	GEN	/T(MUL3,MUL5),TF2,GF9,LA2,RF3.	ALUZ=>MUL3,B(L)->DOR,	FREE 516
517			CVF1,AA1	B15->DSB	517
518	MUL7	GEN	/N(MUL8),MR1,RF5,FF9,WR1,SC1,WF1,	FIRST MULTIPLY STEP	FREE 518
519			CXF1		519
520		MORG	2		520
521	MUL8	GEN	/T(MUL9,*),TF2,GFC,MR1,LA3,RF5,	MULTIPLY STEP	521
522			CFF9,WR1,SC1,WF1,XF1		522
523	MUL9	GEN	/N(MUL11),LA3,WR1,SC1,WF1,XF1,SHO	A,DOR(RT),MULS->DLA15	523
524		MORG	2 /T(MUL12,MUL13)		524
525	MUL13	GEN	/F(SS3M),2(8),FSF,SF1,IM8,RF4.	FSEL(14),IF(P),INCP	525
526			CLA3,WR1,SH4,AA1	SHIFT B	526
527	MUL12	GEN	/N(MUL13),FF6,CF3,WR1,BBF	A=RF->A	527
528	MUL11	GEN	/T(MUL12,MUL13),TF2,GFA,LB1,FFA,	DSB=>MUL12,DOR->B	FREE 528
529			CMF1,WR1,AA1		529
530		GEN		EMPTY WORD1	530
531			CMF1,WR1,AA1		531
532		EJEC			532
533	*		V75 DOUBLE PRECISION AND REGISTER-MEMORY INSTRUCTIONS.		533
534	*				534
535		MORG	16 /T(DPRM2,DPRM1) WITH AB2 FOR	REG SEL	535
536	DPRM1	GEN	/T(DPRM4,DPRM3),TF3,SF3,	CND(-MILS) FSEL(8-6)	TFRS 536
537			CGFB,IM5,LB1,RF3,FFA,MF1,BB1	CND(MILS) OF(ALU),	537
538	*			MIL->DOR	538
539	DPRM2	GEN	/T(DPRM4,DPRM3),TF3,SF3,	CND(-MILS) FSEL(8-6)	TTRS 539
540			CGFB,IM5,LB1,RF3,FF9,BB1,24(*-*)	CND(MILS) OF(ALU),	540
541	*			MIL+AA->DOR	541
542	DST1	GEN	/N(DST2),IM1,FF0,AAE	WAIT(M),RE->ALU	FREE 542
543	DST2	GEN	/N(DPRMST),SF1,AB3,IM6,LR1,FF9,	OS(ALU),DOR+1->ALU,	FREE 543
544			CCF3,BB0,SH1,AAF	F->AA	544
545	DPRMST	GEN	/N(SS2M),AB3,RF4,FF0,24(*-*)	INCP,AA->ALU,HOLD AA	FREE 545
546	DADSB1	GEN	/F(DADD2),2(1),FS7,GF2,IM1,RF3,	RF+RF->DOR,SAMPLE	FREE 546
547			CFF9,BBF,AAF	WAIT(M)	547
548		MORG	2 REF BY FSEL FROM DADSB1		548
549	DADD2	GEN	/F(DPDOR1),2(1),FSA,GF8,LB1,FF9,	RE+MIL+CARRY->RE,	FSEL 549
550			CCF1,WR1,SC1,WF1,BB1,AAE	DOR(RC),SAMPLE OVERFLOW	550
551	DSUB2	GEN	/F(DPDOR1),2(1),FSA,GF8,LB1,FF6,	RE-MIL+CARRY->RE,	FSEL 551
552			CCF1,WR1,SC1,WF1,BB1,AAE	DOR(RC),SAMPLE OVERFLOW	552
553	DLD1	GEN	/F(DPDOR1),2(1),FSA,LB1,FFA,MF1,	MIL->RE	FREE 553
554			CWR1,BB1,AAE		554
555	DAN1	GEN	/F(DPDOR1),2(1),FSA,LB1,FFB,MF1,	RE & MIL->RE	FREE 555
556			CWR1,BB1,AAE		556
557	DOR1	GEN	/F(DPDOR1),2(1),FSA,LB1,FFE,MF1,	RE V MIL->RE	FREE 557
558			CWR1,BB1,AAE		558
559	DER1	GEN	/F(DPDOR1),2(1),FSA,LB1,FF6,MF1,	RE XOR MIL->RE	FREE 559
560			CWR1,BB1,AAE		560
561		MORG	2 REF BY FSEL FROM DLD1,DADD2,DSUB2,DAN1,DOR1,DER1		561
562	DPDOR1	GEN	/N(DPRERO),SF1,IM8,LB1,RF4,FFA,	IF(P),INCP,DOR->R1	FSEL 562
563			CMF1,WR1,BB0,AA1		563
564	DPDOR5	GEN	/N(DPRER4),SF1,IM8,LB1,RF4,FFA,	IF(P),INCP,DOR->R5	FSEL 564
565			CMF1,WR1,BB0,AA5		565
566	DPRERO	GEN	/N(IWAIT),1(M),GF5,IM6,FFA,MF1,	IBR->I,DECODE,INTRPT,	FREE 566
567			CWR1,BBE,AA0	SEL&RST CINTF,RE->R0	567
568	DPRER4	GEN	/N(IWAIT),1(M),GF5,IM6,FFA,MF1,	IBR->I,DECODE,INTRPT,	FREE 568
569			CWR1,BBE,AA4	SEL&RST CINTF,RE->R4	569

570		MORG	16 REF BY FSEL FROM DPRM4			570
571	RMLD	GEN	/N(RM1),SF1,IM4,LB1,RF4,FFA,MF1	OF(ALU),INCP,DOR->ALU	FSEL	571
572	RMST	GEN	/N(DPRMST),TS3,SF1,AB2,IM6,LB1,	OS(ALU),DOR->ALU,	FSEL	572
573			CFFA,MF1,BB0	I(5-3)->AA		573
574	RMAD	GEN	/N(RM1),SF1,IM4,LB1,RF4,FFA,MF1	OF(ALU),INCP,DOR->ALU	FSEL	574
575	RMSB	GEN	/N(RM1),SF1,IM4,LB1,RF4,FFA,MF1	OF(ALU),INCP,DOR->ALU	FSEL	575
576		MORG	2 FAIL OF CND FSEL TO RMLD ETC.			576
577	DPRM3	GEN	/N(DPRM1),IM1,RF5	WAIT(M),INCS	TF	577
578	DPRM4	GEN	/F(RMLD),2(7),FSA,SF1,IM4,LA1,CF3	IF(ALU),P+1->ALU	TT	578
579		ORG	RMLD+6			579
580	DPR0RE	GEN	/N(DPR1RF),FFA,MF1,WR1,BB0,AAE	RD->RE	FSEL	580
581	DPR4RE	GEN	/N(DPR5RF),FFA,MF1,WR1,BB4,AAE	R4->RE	FSEL	581
582		ORG	RMLD+8 REF BY FSEL WITH REG SEL	FROM RM1, 1 IMPOSSIBLE		582
583	RMLD1	GEN	/N(IWAIT),1(M),GF5,IM6,LB1,FFA,	IBR->I,DECODE,INTRPT,	FSRS	583
584			CMF1,WR1,23(*-*),24(*-*)	SEL&RST CINTF,MIL->AA		584
585	*	RM1 IS	NOT REFERENCED BY FIELD SELECTION.			585
586	RM1	GEN	/F(RMLD1-8),TS3,2(B),FSA,SF1,AB2,	IF(P),INCP,I(5-3)->AA,	FREE	586
587			CIM8,RF4,BB1	1->BB,FSEL(7-6)		587
588		ORG	RMLD1+2			588
589	RMAD1	GEN	/N(IWAIT),1(M),GFD,IM6,LB1,FF9,	IBR->I,DECODE,INTRPT,	FSRS	589
590			CWR1,23(*-*),24(*-*)	SEL&RST CINTF,		590
591	*			MIL+AA->AA,SAMPLE OVFL		591
592	RMSB1	GEN	/N(IWAIT),1(M),GFD,IM6,LB1,FF6,	IBR->I,DECODE,INTRPT,	FSRS	592
593			CCF3,WR1,23(*-*),24(*-*)	SEL&RST CINTF,		593
594	*			AA=MIL->AA,SAMPLE OVFL		594
595	DPR1RF	GEN	/N(DP1),FFA,MF1,WR1,BB1,AAF	R1->RF	FREE	595
596	DPR5RF	GEN	/N(DP1),FFA,MF1,WR1,BB5,AAF	R5->RF	FREE	596
597	DP1	GEN	/N(DP2),SF1,IM5,LB1,FF9,CF3,SH1	OF(ALU),DOR+1->ALU	FREE	597
598	DP2	GEN	/F(DLD),2(7),FS7,SF1,IM5,LB1,FFA,	OF(ALU),DOR->ALU,	FREE	598
599			CMF1,BB0	FSEL(5-3)		599
600		MORG	8 REF BY FSEL FROM DP2			600
601	DLD	GEN	/N(DLD1),IM1,LB1,RF7,FFA,MF1,BB1	WAIT(M),INCP,MIL->DOR	FSEL	601
602	DST	GEN	/N(DST1),SF1,IM2,AAE	OVR(OS),RE->ALU	FSEL	602
603	DADD	GEN	/N(DADSB1),LB1,RF4,FF9,WR1,BB1,	RF+MIL->RF,INCP	FSEL	603
604			CAAF			604
605	DSUB	GEN	/N(DADSB1),LB1,RF4,FF6,CF3,WR1,	RF-MIL->RF,INCP	FSEL	605
606			CBB1,AAF			606
607	DAN	GEN	/N(DAN1),IM1,LB1,RF7,FFB,MF1,	RF&MIL->RF,INCP,	FSEL	607
608			CBB1,AAF	WAIT(M)		608
609	DOR	GEN	/N(DOR1),IM1,LB1,RF7,FFE,MF1,	RF V MIL->RF,INCP,	FSEL	609
610			CBB1,AAF	WAIT(M)		610
611	DER	GEN	/N(DER1),IM1,LB1,RF7,FF6,MF1,	RF XOR MIL->RF,INCP,FSEL		611
612			CBB1,AAF	WAIT(M)		612
613		ILLOP			FSEL	613
614	*	DPRM C	NOT DO IF(P) BECAUSE THE MEMORY LATCH ISN'T A TRUE-D			614
615	*	LATCH,	SO THE TESTS AT DPRM1, DPRM2 DO NOT WORK PROPERLY IF THE			615
616	*	MEMORY	IS ACTIVE.			616
617	DPRM	GEN	/T(DPRM2,DPRM1),TS0,TF3,GF9,	ALUZ=>DPRM1,WAIT(M),	FREE	617
618			CAB2,IM1,XF1,BB1	0->CINTF,I(2-0)->AA		618
619		EJEC				619
620	*	EXECUTE	REMOTE INSTRUCTION.			620
621	*					621
622	SS3	GEN	/N(IWAIT),1(N),GF5,IM6	DECODE, EN INTS, ETC	FREE	622
623		MORG	2 /T(XEC1,XEC3)			623
624	XEC3	GEN	/N(SS2),SF1,IM8,RF4	IF(P),INCP		624
625	XEC1	GEN	/T(XEC2,SS2),TF2,GFB	MIL15=>XEC2		625
626		MORG	2 /T(XEC2,SS2), FSEL(STEP)			626

```

627 SS2 GEN /N(SS3),SF1,IM8,RF4 IF(P),INCP 627
628 HSTT4 GEN /N(HSTT5),IM1,RF4 WAIT(M),INCP 628
629 XEC2 GEN /N(XEC1),SF1,IMC,RF5 IF(MIL),INC SC 629
630 EJEC 630
631 * SET I/O DEVICE SELECTION REGISTER. THIS IS NOT A VARIAN 631
632 * INSTRUCTION. THIS INSTRUCTION PLACES 6 BITS IN REGISTER DR, 632
633 * WHICH ARE SUBSEQUENTLY INCLUSIVE-OR'ED WITH ALL DEVICE SELECTION 633
634 * SPECIFICATIONS SENT TO THE I/O BUS. THE OCTAL FORM FOR THIS 634
635 * INSTRUCTION IS: 635
636 * 107TDD WHERE 636
637 * T=0: DD->DR T=4: X->DR 637
638 * T=1: A->DR T=5: A V X -> DR 638
639 * T=2: B->DR T=6: B V X -> DR 639
640 * T=3: A V B -> DR T=7: A V B V X -> DR 640
641 * 641
642 SETXIT GEN /N(IWAIT),1(1),GF5,IM6,LB1,FFA, DECODE,IBR->I,EN INT FREE 642
643 CMF1,WR1,BB0,24(DR) DOR->DR 643
644 MORG 8 REFERENCED BY FSEL(8-6) 644
645 SETI GMSK /N(SETXIT),SF1,IM8,LB2,RF7,FFA, I60'77->DOR 645
646 CMKFFCO IF(P),INCP 646
647 SETA GMSK /N(SETXIT),SF1,IM8,LB3,RF7,FF7, DR60'77->DOR 647
648 CMK3F,16(DR) IF(P),INCP 648
649 SETB GMSK /N(SETXIT),SF1,IM8,LB3,RF7,FF7, B60'77->DOR 649
650 CMK3F,AK1 IF(P),INCP 650
651 SETAB GEN /N(SETA),FFE,MF1,WR1,BB1,24(DR) B V DR -> DR 651
652 SETX GMSK /N(SETXIT),SF1,IM8,LB3,RF7,FF7, X60'77->DOR 652
653 CMK3F,AK2 IF(P),INCP 653
654 SETAX GEN /N(SETA),FFE,MF1,WR1,BB2,24(DR) X V DR -> DR 654
655 SETBX GEN /N(SETAX),FFA,MF1,WR1,BB1,24(DR) B->DR 655
656 SETABX GEN /N(SETAX),FFE,MF1,WR1,BB1,24(DR) B V DR -> DR 656
657 EJEC 657
658 * EXTERNAL CONTROL INSTRUCTION. 658
659 * 659
660 EXC2 GEN /N(EXC3),1(1),IMF,LB1,FFA,MF1, START IO,DOR->S1 FREE 660
661 CWR1,BB0,24(S1) 661
662 EXC3 GMSK /N(SS2),IM2,LB2,FFE,MKFED0,16(S1) WAIT(IO),I60'777 V S1 FREE 662
663 EXC22 GMSK /N(SS2),IM2,LB2,FFE,MK7E00, I6X'81FF V DR, WAIT(IO) FREE 663
664 C16(DR) 664
665 * SENSE INSTRUCTION. 665
666 * 666
667 SEN2 GEN /N(SEN3),1(1),IMF,LB1,FFA,MF1, START I/O AT X'4, FREE 667
668 CWR1,XF1,BB0,24(S1) DOR->S1,RESET CINTF 668
669 SEN3 GMSK /N(SEN4),IM2,LB2,FFE,MKFED0, WAIT(IO), FREE 669
670 C16(S1) I6X'1FF V S1 ->ALU 670
671 SEN4 GEN /T(SEN5,SS1M),TF2,SF3,GF1,IMC,RF4 I/O SENSE=>SEN5,INCP, FREE 671
672 * IF(MIL) IF I/O SENSE 672
673 MORG 2 /T(SEN6,SEN7) 673
674 SEN7 GEN /N(SEN5),SF1,IMC,RF5 IF(MIL),INCSC 674
675 SEN6 GEN /N(SS3M),SF1,IM8 IF(P) 675
676 EJEC 676
677 * DOUBLE-WORD EXTENDED INSTRUCTIONS. NOTE THAT THESE INSTRUCTIONS 677
678 * ARE DIFFERENT FROM VARIAN'S SINCE AN M-FIELD EQUAL TO ZERO OR 8 678
679 * DOES NOT CAUSE UNPREDICTABLE RESULTS. 679
680 * 680
681 MORG 32 ELDA ETC REF BY CND FSEL, FAIL TO DEIND 681
682 EXTBAS EQU * 682
683 DLBX GEN /N(DL1),SF1,IM5,LB1,RF4,FF9,CF3, OF(ALU), DOR+1->ALU, 683

```

684			CSH1,BB0	INCP	684
685	ELDA	GEN	/N(ELDA1),SF1,IM4,RF1,FF0,CF3,AAE	IF(A),S1+1->P	685
686	ELDB	GEN	/N(ELDB1),SF1,IM4,RF1,FF0,CF3,AAE	IF(A),S1+1->P	686
687	ELDX	GEN	/N(ELDX1),SF1,IM4,RF1,FF0,CF3,AAE	IF(A),S1+1->P	687
688	EINR	GEN	/N(EINR1),SF1,IM6,LB1,RF4,FFA,MF1	OS(A),DOR->ALU,INCP	688
689	ESTA	GEN	/N(SS2M),SF1,IM2,AB3,RF4,AA0	OVR(OS),A->AA,INCP	689
690	ESTB	GEN	/N(SS2M),SF1,IM2,AB3,RF4,AA1	OVR(OS),B->AA,INCP	690
691	ESTX	GEN	/N(SS2M),SF1,IM2,AB3,RF4,AA2	OVR(OS),X->AA,INCP	691
692	ESTZ	GEN	/N(SS2M),SF1,IM2,AB2,RF4,SH1,WR1,	OVR(OS),INCP,0->F,F->AA	692
693			CAAF		693
694	EORA	GEN	/N(EORA1),SF1,IM4,RF1,FF0,CF3,AAE	IF(A),S1+1->P	694
695	EADD	GEN	/N(EADD1),SF1,IM4,RF1,FF0,CF3,AAE	IF(A),S1+1->P	695
696	EERA	GEN	/N(EERA1),SF1,IM4,RF1,FF0,CF3,AAE	IF(A),S1+1->P	696
697	ESUB	GEN	/N(ESUB1),SF1,IM4,RF1,FF0,CF3,AAE	IF(A),S1+1->P	697
698	EANA	GEN	/N(EANA1),SF1,IM4,RF1,FF0,CF3,AAE	IF(A),S1+1->P	698
699	EMUL	GEN	/N(MULO),IM1,RF4	WAIT(M),INCP	699
700	EDIV	GEN	/N(DIVO),IM1,RF4	WAIT(M),INCP	700
701		MORG	16 REF BY CND FSEL, FAIL TO DEIND		701
702	DERL	GEN	/S(EXTBAS,DEIND),2(F),FS7,TF3,	C.FSEL(6-3) (QS=0),	702
703			CSF1,GFF,IM5,LA1,LB1,RF3,FF9,BB0	OF(A),P+DOR->DOR	703
704	DEIX	GEN	/S(EXTBAS,DEIND),2(F),FS7,TF3,	C.FSEL(6-3) (QS=0),	704
705			CSF1,GFF,IM5,LB1,RF3,FF9,BB0,AA2	OF(A),X+DOR->DOR	705
706	DEIB	GEN	/S(EXTBAS,DEIND),2(F),FS7,TF3,	C.FSEL(6-3) (QS=0),	706
707			CSF1,GFF,IM5,LB1,RF3,FF9,BB0,AA1	OF(A),B+DOR->DOR	707
708	DENI	GEN	/S(EXTBAS,DEIND),2(F),FS7,TF3,	C.FSEL(6-3) (QS=0),	708
709			CSF1,GFF,IM5,LB1,FFA,MF1,BB0	OF(A),DOR->ALU	709
710		MORG	2 REF BY FSEL(7)		710
711	DEPRE	GEN	/F(DERL),2(3),FS4,LB1,RF3,	FSEL(1,0),	711
712			CFFA,MF1,WF1,BB1	MIL->DOR,MIL15->QS	712
713	DEPOST	GEN	/S(DERL,DEPST1),2(3),FS4,TF3,	C.FSEL(1,0) (MIL15=0),	713
714			CGFB,LB1,RF3,FFA,MF1,BB1	MIL->DOR	714
715		MORG	2 /T(DEPST1,DEIND1)		715
716	DEIND1	GEN	/S(EXTBAS,DEIND),2(F),FS7,TF3,	C.FSEL(6-3) (MIL15=0),	716
717			CSF1,GFB,IM5,LB1,RF3,FFA,MF1,BB1	OF(A),MIL->DOR	717
718	DEPST1	GEN	/S(DERL,DEIND),2(3),FS4,TF3,SF3,	C.FSEL(1,0) (MIL15=0),	718
719			CGFB,IM5,LB1,RF3,FFA,MF1,BB1	CND(MIL15=1) OF(ALU),	719
720	*			MIL->DOR	720
721		MORG	2 /S(EXTBAS,DEIND)		721
722	DEIND	GEN	/T(DEPST1,DEIND1),TF3,GFF,IM1,RF5	WAIT(M),QS=>DEIND1,INCS	722
723	DL1	GEN	/N(ELDX1),SF1,IM8,LB1,RF4,FFA,	IF(P),INCP,MIL->B FREE	723
724			CMF1,WR1,BB1,AA1		724
725		MORG	2 /S(DERL,DEPST1)		725
726	DEPST1	GEN	/N(DEIND),SF1,IM5,LB1,FFA,MF1,BB0	OF(ALU),DOR->ALU	726
727		EJEC			727
728	*		SKIP I= REGISTER EQUAL. THIS CODE IS SOMEWHAT DIFFERENT FROM		728
729	*		VARIAN'S SINCE SETTING OF MORE THAN ONE OF THE A, B OR X BITS		729
730	*		DOES NOT CAUSE UNPREDICTABLE RESULTS.		730
731	*				731
732	SRESX	GEN	/N(SRE60),IM1,LA1,RF1,FF1,BB0	WAIT(M),P V A -> P FREE	732
733		MORG	4 REF BY FSEL(1=0)		733
734	SRE20	GEN	/S(SRE50,SRE30),2(7),FS7,TF3,SF2,	CND(MIL15=0) FSEL(5-3),	734
735			CGFB,IM5,LB1,LA1,FF9,BB1	CND(MIL15=0) OF(ALU),	735
736	*			P+MIL->ALU	736
737	SRE21	GEN	/S(SRE50,SRE30),2(7),FS7,TF3,SF2,	CND(MIL15=0) FSEL(5-3),	737
738			CGFB,IM5,LB1,FF9,BB1,AA2	CND(MIL15=0) OF(ALU),	738
739	*			X+MIL->ALU	739
740	SRE22	GEN	/S(SRE50,SRE30),2(7),FS7,TF3,SF2,	CND(MIL15=0) FSEL(5-3),	740

```

741 CGFB,IM5,LB1,FF9,BB1,AA1 CND(MIL15=0) OF(ALU), 741
742 * B+MIL->ALU 742
743 SRE23 GEN /S(SRE50,SRE30),2(7),FS7,TF3,SF1, CND(MIL15=0) FSEL(5-3), 743
744 CGFB,IMD OF(MIL) 744
745 MORG 16 SRES+ REF BY CND FSEL 745
746 SRE50 GEN /N(SRE60),IM1,RF1,SH1 WAIT(M),0->P 746
747 SRE51 GEN /N(SRE60),IM1,RF1,AA0 WAIT(M),A->P 747
748 SRE52 GEN /N(SRE60),IM1,RF1,AA1 WAIT(M),B->P 748
749 SRE53 GEN /N(SRE60),IM1,RF1,FF1,BB1,AA0 WAIT(M),A V B -> P 749
750 SRE54 GEN /N(SRE60),IM1,RF1,AA2 WAIT(M),X->P 750
751 SRE55 GEN /N(SRE60),IM1,RF1,FF1,BB2,AA0 WAIT(M),A V X -> P 751
752 SRE56 GEN /N(SRE60),IM1,RF1,FF1,BB2,AA1 WAIT(M),B V X -> P 752
753 SRE57 GEN /N(SRE5X),RF1,FF1,BB2,AA1 B V X -> P 753
754 MORG 2 MORG 32 WITH SRE50 754
755 SRE30 GEN /N(SRE40),SF1,IMD OF(MIL) 755
756 SRE60 GEN /N(SRE70),GF2,LB1,LA1,FF6,MF1,BB1 MIL XOR P, SAMPLE FREE 756
757 MORG 2 MORG 32 WITH SRE50 757
758 SRE40 GEN /F(SRE20),2(3),FS4,IM1,RF5 FSEL(1-0),WAIT(M),INCS 758
759 SRE70 GEN /T(SRE80,SS2M),TF2,SF2,GF9,IM4, CND(NO ALUZ) IF(ALU), FREE 759
760 CLB1,RF1,BB0 DOR->P 760
761 EJEC 761
762 * REGISTER - TO - REGISTER TRANSFER INSTRUCTIONS. 762
763 * 763
764 REG30 GEN /F(REG20),2(7),FS4,SF1,IM8,LB1,RF3,FFE,MF1 FREE 764
765 REG41 GEN /N(REG20),SF1,GF6,IM8,LB1,FF9,WR1,SH5 FREE 765
766 REG42 GEN /N(REG20),SF1,GF6,IM8,LB1,FF9,WR1,SH5,AA1 FREE 766
767 REG43 GEN /N(REG21),SF1,GF6,IM8,LB1,FF9,WR1,SH5,AA1 FREE 767
768 MORG 16 REF BY CND FSEL(2-0) 768
769 REG20 GEN /N(IWAIT),1(1),GF5,IM6 769
770 REG21 GEN /N(IWAIT),1(1),GFD,IM6,LB1,FF9,WR1,SH5 770
771 REG22 GEN /N(IWAIT),1(1),GFD,IM6,LB1,FF9,WR1,SH5,AA1 771
772 REG23 GEN /N(REG41),SF1,GF6,IM1,LB1,FF9,WR1,SH5,AA1 772
773 REG24 GEN /N(IWAIT),1(1),GFD,IM6,LB1,FF9,WR1,SH5,AA2 773
774 REG25 GEN /N(REG41),SF1,GF6,IM1,LB1,FF9,WR1,SH5,AA2 774
775 REG26 GEN /N(REG42),SF1,GF6,IM1,LB1,FF9,WR1,SH5,AA2 775
776 REG27 GEN /N(REG43),SF1,GF6,IM1,LB1,FF9,WR1,SH5,AA2 776
777 ***** D E C O D E R ***** 777
778 ORG X*158 DECODER REFERENCES NEXT 8 INSTRUCTIONS * 778
779 REG18 GEN /S(REG20,REG20),2(7),FS4,TF2,SF1,IM8,RF7,SH1 * 779
780 REG1A GEN /S(REG20,REG20),2(7),FS4,TF2,SF1,IM8,RF7,AA1 * 780
781 REG1C GEN /S(REG20,REG20),2(7),FS4,TF2,SF1,IM8,RF7,AA2 * 781
782 REG1E GEN /S(REG20,REG20),2(7),FS4,TF2,SF1,IM8,RF7,FFE, * 782
783 CMF1,BB2,AA1 * 783
784 REG19 GEN /S(REG20,REG20),2(7),FS4,TF2,SF1,IM8,RF7 * 784
785 REG1B GEN /S(REG20,REG20),2(7),FS4,TF2,SF1,IM8,RF7,FFE, * 785
786 CMF1,BB1 * 786
787 REG1D GEN /S(REG20,REG20),2(7),FS4,TF2,SF1,IM8,RF7,FFE, * 787
788 CMF1,BB2 * 788
789 REG1F GEN /T(REG30,REG20),TF2,SF2,IM8,RF7,FFE,MF1,BB2,AA1 * 789
790 ***** 790
791 EJEC 791
792 * HALT LOOP PROCESSING. CONTROL GETS HERE VIA THE HALT 792
793 * INTRUCTIONS, THE RESET KEY AND THE STEP KEY. 793
794 * 794
795 MORG 16 REF BY FSEL(5-4), (5-4)=11 IMPOSSIBLE, REG FLD SELECT 795
796 HREN1 GEN /N(HMEM) NOP (CAN'T CHANGE STAT) 796
797 HREN2 GEN /N(HMEM),LB1,RF1,FFA,MF1 DOR->P 797

```

```

798 HREN3 GEN /N(HMEM),LB1,FFA,MF1,WR1 DOR->R** 798
799 * HMDP1 IS NOT REFERENCED BY FSEL. 799
800 HMDP1 GEN /N(HRDP1),IM1,RF4,AB3,BB1 WAIT(M),INCP,"MIL"->BB FREE 800
801 MORG 4 REF BY FSEL(7-6), (7-6)=10 IMPOSSIBLE 801
802 HREN GEN /F(HREN1),1(0),2(3),FS8,AB2,MR1 FSEL(5-4),(3-0)->AA 802
803 HMEMA GEN /N(HLT1),1(4),GF4,IMC SAMPLE INTERRUPTS 803
804 * HMEN4 IS NOT REFERENCED BY FSEL 804
805 HMEN4 GEN /N(HMEM),LB1,RF4,FFA,MF1 WAIT(M),INCP,DOR->ALU FREE 805
806 HMEN3 GEN /N(HMEN4),SF1,IMA OS(P) 806
807 ***** D E C O D E R ***** 807
808 ORG X'168 DECODER REFERENCES NEXT 8 INSTRUCTIONS. * 808
809 IMMREL GEN /F(IMPRE),2(1),FSB,SF1,IM4,LA1, P+1->S1,IF(A),0->QS, * 809
810 CFFD,CF3,WR1,WF1,24(S1) FSEL(7) * 810
811 IMMIX GEN /F(IMPRE),2(1),FSB,SF1,IM4,LA1, P+1->S1,IF(A),0->QS, * 811
812 CFFD,CF3,WR1,WF1,24(S1) FSEL(7) * 812
813 IMMIB GEN /F(IMPRE),2(1),FSB,SF1,IM4,LA1, P+1>S1,IF(A),0->QS, * 813
814 CFFD,CF3,WR1,WF1,24(S1) FSEL(7) * 814
815 IMMNI GEN /N(IMNI),SF1,IM4,LA1,FFD,CF3, P+1->S1,IF(A) * 815
816 CWR1,24(S1) * 816
817 DE#1 GEN /F(DEPRE),2(1),FSB,SF1,IM4,LA1, P+1->S1,IF(A),0->QS, * 817
818 CFFD,CF3,WR1,WF1,XF1,AAE FSEL(7),0->CINTF * 818
819 DE#2 GEN /F(DEPRE),2(1),FSB,SF1,IM4,LA1, P+1->S1,IF(A),0->QS, * 819
820 CFFD,CF3,WR1,WF1,XF1,AAE FSEL(7),0->CINTF * 820
821 DE#3 GEN /F(DEPRE),2(1),FSB,SF1,IM4,LA1, P+1->S1,IF(A),0->QS, * 821
822 CFFD,CF3,WR1,WF1,XF1,AAE FSEL(7),0->CINTF * 822
823 DE#4 GEN /F(DEPRE),2(1),FSB,SF1,IM4,LA1, P+1->S1,IF(A),0->QS, * 823
824 CFFD,CF3,WR1,WF1,XF1,AAE FSEL(7),0->CINTF * 824
825 ***** 825
826 MORG 16 REF BY FSEL(5-4), (5-4)=11 IMPOSSIBLE, REG FLD SELECT 826
827 HRDP1 GEN /N(HFPT),LB1,RF3,FFA,MF1 STATUS->DOR 827
828 HRDP2 GEN /N(HFPT),LA1,RF3 P->DOR 828
829 HRDP3 GEN /N(HFPT),RF3 R**->DOR 829
830 * HFPT IS NOT REFERENCED BY FSEL. 830
831 HFPT GMSK /N(HFPT1),1(7),AB2,IMF,LB3,FFA, START IO @ X'1E FREE 831
832 CMKBFCO X'403F->ALU (DT0) 832
833 MORG 2 /T(HSTT6,HSTT2) 833
834 HSTT2 GEN /F(SS2),2(1),FS3,SF1,IM8 FSEL(STEP),IF(P) 834
835 HSTT6 GEN /N(HSTT7),IM4,24(IH) IH->IBR,MIL 835
836 HSTT1 GEN /T(HSTT6,HSTT2),TF3,GF9 ALUZ=>HSTT2 FREE 836
837 HSTT5 GEN /N(HLT),GF1 IBR->I FREE 837
838 MORG 8 REF BY FSEL(8-6) 838
839 HRDP GEN /F(HRDP1),1(0),2(3),FS8,MR1,AB2, FSEL(5-4),(3-0)->AA 839
840 CBB3 STATUS->BB 840
841 HMEM GEN /N(HLT1),1(4),GF4,IMC SAMPLE INTERRUPTS 841
842 HSTT GMSK /N(HSTT1),GF2,LB3,FFB,MK01FF, IH&X'FE00,SAMPLE ALU 842
843 C16(IH) (WAS IT A HALT INST?) 843
844 HMEN GMSK /N(HMEN1),1(3),AB2,IMF,LB3,FFA, START IO @ X'E, 844
845 CMKDFCO X'203F->ALU (DT1) 845
846 ABL10 GMSK /N(ABL20),SF1,GF4,IM6,LB3,RF1, OS(ALU),X'80->P, 846
847 CFFA,MKFF7F RESET OVERFLOW 847
848 HMDP GEN /N(HMDP1),SF1,IM9 OF(P) 848
849 ABL11 GMSK /N(ABL21),SF1,GF2,IM6,LB3,RF1, OS(ALU),X'80->P 849
850 CFFA,MKFF7F SET OVERFLOW 850
851 ABL30 GMSK /N(ABL31),SF1,IM6,LB3,RF1,FFA, OS(ALU),X'258->P 851
852 CMKFDA7 852
853 EJEC 853
854 * INDEXE, JUMP INSTRUCTION. 854

```

```

855 * NOTE THAT THIS VERSION OF IJMP DIFFERS FROM VARIAN'S VERSION 855
856 * SINCE BIT 2 MAY BE USED TO INDICATE PRE- OR POST-INDEXING, AS MAY 856
857 * BE DONE USING BIT 7 WITH OTHER EXTENDED ADDRESSING INSTRUCTIONS. 857
858 * ALSO, BITS 4&3 HAVE THE FOLLOWING MEANING: 858
859 * 00 - JUMP 859
860 * 11 - XEC 860
861 * 01 - JSR,X 861
862 * 10 - JSR,B 862
863 * IJRL,IJIX,IJIB,IJNI,IJIND1 ALL DO CND FSEL (6,4,3) BUT BIT 6 = 1. 863
864 MORG 16 REF BY CND FSEL(1=0) 864
865 IJRL GEN /S(IJXIT,IJIND),2(B),FS7,TF3,SF1, QS=>IJIND,P+DOR->DOR, 865
866 CGFF,IM4,LA1,LB1,RF3,FF9,XF2,BB0 IF(A),EN JUMP SIGNAL 866
867 IJIX GEN /S(IJXIT,IJIND),2(B),FS7,TF3,SF1, QS=>IJIND,X+DOR->DOR, 867
868 CGFF,IM4,LB1,RF3,FF9,XF2,BB0,AA2 IF(A),EN JUMP SIGNAL 868
869 IJIB GEN /S(IJXIT,IJIND),2(B),FS7,TF3,SF1, QS=>IJIND,B+DOR->DOR 869
870 CGFF,IM4,LB1,RF3,FF9,XF2,BB0,AA1 IF(A),EN JUMP SIGNAL 870
871 IJNI GEN /S(IJXIT,IJIND),2(B),FS7,TF3,SF1, QS=>IJIND,DOR->ALU, 871
872 CGFF,IM4,LB1,FFA,MF1,XF2,BB0 IF(A),EN JUMP SIGNAL 872
873 MORG 2 /T(IJPOST,IJIND1) 873
874 IJIND1 GEN /S(IJXIT,IJIND),2(B),FS7,TF3,SF1, MIL15=>IJIND,MIL->DOR, 874
875 CGFB,IM4,LB1,RF3,FFA,MF1,WF1,XF2, MIL15->QS,J-SIG,IF(A) 875
876 CBB1 876
877 HLT2 GMSK /N(HLT3),LB3,RF3,FFA,MKFFCD 077->DOR FREE 877
878 MORG 2 /S(IJXIT,IJIND) 878
879 IJIND GEN /T(IJPOST,IJIND1),TF3,GFF,IM1,RF5 WAIT(M),QS=>IJIND1,INCS 879
880 HLT3 GEN /N(HLT4),IM2,LB1,FFB,MF1,WR1,BB0, WAIT(I0),DOR&DR->DR FREE 880
881 C24(DR) 881
882 MORG 8 REF BY CND FSEL(6,4,3) WITH BIT 6 ALWAYS = 1. 882
883 IJXIT GEN /N(SS3M),SF1,IM4,LB1,RF1,FF9,CF3, IF(A),DOR+1->P 883
884 CSH1,BB0 884
885 IJPX GEN /N(IJXIT),LA1,FF0,CF3,WR1,AA2 P+1->X 885
886 IJPB GEN /N(IJXIT),LA1,FF0,CF3,WR1,AA1 P+1->B 886
887 IJXEC GEN /N(SS3),SF1,IM8,RF4 (SAME AS SS2) IF(P),INCP 887
888 MORG 2 REF BY FSEL(2) 888
889 IJPRE GEN /F(IJRL),2(3),FS4,LB1,RF3, FSEL(1=0), 889
890 CFFA,MF1,WF1,BB1 MIL->DOR,MIL15->QS 890
891 IJPOST GEN /S(IJRL,IJIND),2(3),FS4,TF3,SF3, C.FSEL(1=0) (MIL15=0), 891
892 CGFB,IM5,LB1,RF3,FFA,MF1,BB1 C.OF(ALU) (MIL15=1), 892
893 * MIL->DOR 893
894 HLT4 GEN /N(HLT5),IM4,LB1,RF3,FFA,MF1,BB2 IOR->DOR,ALU->IBR FREE 894
895 HLT5 GEN /N(HLT6),GF1,LB1,FFA,MF1,BB0 DOR->ALU,IBR->IR FREE 895
896 EJEC 896
897 * SINGLE WORD ADDRESSING STORE INSTRUCTIONS. 897
898 * AUTOMATIC BOOTSTRAP LOADER FOR TELETYPE AND HIGH-SPEED PAPER TAPE 898
899 * DEPOSITED AT 0200, STARTED WITH X=07000 AT 0212. 899
900 * 900
901 MORG 8 REF BY FSEL(13-12), 00 NOT POSSIBLE. 901
902 STABX1 EQU * 902
903 GEN EMPTY WORD 1 903
904 GEN EMPTY WORD 1 904
905 ORG STABX1+2 905
906 STA1 GEN /T(SWA50,SS2MI),TF2,SF3,GFB,AB3, A->ALU,SET AA,INCS 906
907 CIM1,RF5,AA0 MIL15=>OVR(OF),->SWA50 907
908 ABL20 GMSK /N(ABL3),SF1,IMA,LB3,RF4,FFA,MK7A7E 0200 CIB 001 FREE 908
909 ORG STA1+2 909
910 STB1 GEN /T(SWA50,SS2MI),TF2,SF3,GFB,AB3, B->ALU,SET AA,INCS 910
911 CIM1,RF5,AA1 MIL15=>OVR(OF),->SWA50 911

```



```

912 ABL21 GMSK /N(ABL3),SF1,IMA,LB3,RF4,FFA,MK7A60 0200 CIB 037 FREE 912
913 ORG STB1+2 913
914 STX1 GEN /T(SWA50,SS2MI),TF2,SF3,GFB,AB3, X->ALU,SET AA,INCS 914
915 CIM1,RF5,AA2 MIL15=>OVR(OF),->SWA50 915
916 ABL3 GMSK /N(ABL4),SF1,IMA,LB3,RF4,FFA,MKF7F6 0201 ASLB 9 FREE 916
917 MORG 8 REF BY FSEL(13-12), 00 NOT POSSIBLE. 917
918 STABX2 EQU * 918
919 ABL4 GMSK /N(ABL5),SF1,IMA,LB3,RF4,FFA,MKF7DE 0202 LRLB 1 FREE 919
920 ABL5 GMSK /N(ABL6),SF1,IMA,LB3,RF4,FFA,MKF6D9 0203 LLRL 6 FREE 920
921 ORG STABX2+2 921
922 STA2 GEN /N(SS2MI),AB3,AA0 A->ALU,SET AA 922
923 ABL6 GMSK /N(ABL7),SF1,IMA,LB3,RF4,FFA,MKFDEF 0204 JBZ FREE 923
924 ORG STA2+2 924
925 STB2 GEN /N(SS2MI),AB3,AA1 B->ALU,SET AA 925
926 ABL7 GMSK /N(ABL8),SF1,IMA,LB3,RF4,FFA,MKFF73 0205 0214 FREE 926
927 ORG STB2+2 927
928 STX2 GEN /N(SS2MI),AB3,AA2 X->ALU,SET AA 928
929 ABL8 GMSK /N(ABL9),SF1,IMA,LB3,RF4,FFA,MKA5FF 0206 STA 0,X FREE 929
930 ABL9 GMSK /N(ABLA),SF1,IMA,LB3,RF4,FFA,MKDF7 0207 JAZ FREE 930
931 ABLA GMSK /N(ABLB),SF1,IMA,LB3,RF4,FFA,MKF1FF 0210 07000->DOR FREE 931
932 ABLB GMSK /N(ABLC),SF1,IMA,LB3,RF4,FFA,MKF59B 0211 IXR FREE 932
933 ABLC GMSK /T(ABLD1,ABLD0),TF2,SF1,IMA,LB3,RF4, 0212 INCR 1 FREE 933
934 CFFA,MKF5BE 934
935 MORG 2 /T(ABLD1,ABLD0) 935
936 ABLD0 GMSK /N(ABLE0),SF1,IMA,LB3,RF4,FFA,MK7A7E 0213 CIB 001 936
937 ***** D E C O D E R ***** 937
938 ORG X*1A5 DECODER REFERENCES NEXT 2 INSTRUCTIONS * 938
939 SWA47 GMSK /N(SWA50),SF1,IM5,LB2,FFA,MKFED0 OF(ALU),I&X*1FF->ALU * 939
940 SWA17 GMSK /N(SWA21),SF1,IM5,LB2,FFA,MKFED0 OF(ALU),I&X*1FF->ALU * 940
941 ***** 941
942 ABLD1 GMSK /N(ABLE1),SF1,IMA,LB3,RF4,FFA,MK7EAD 0213 EXC 0537 942
943 ABLE0 GMSK /N(ABLF),SF1,IMA,LB3,RF4,FFA,MK7D7E 0214 SEN 0201 FREE 943
944 ABLE1 GMSK /N(ABLF),SF1,IMA,LB3,RF4,FFA,MK7CA0 0214 SEN 0537 FREE 944
945 ABLF GMSK /N(ABLX0),SF1,IMA,LB3,RF4,FFA,MKFF7F 0215 0200 FREE 945
946 ABLX0 GMSK /N(ABLX1),SF1,IMA,LB3,RF4,FFA,MKFDFF 0216 JMP FREE 946
947 ***** D E C O D E R ***** 947
948 ORG X*1AC DECODER REFERENCES NEXT 2 INSTRUCTIONS * 948
949 RM GMSK /N(DPRM),GF2,LB2,FFA,MKFFF8 I&7, SAMPLE *DECO 949
950 RR GEN /N(RR1),TS0,MRO,AB2 I(2-0)->AA *DECO 950
951 ***** 951
952 ABLX1 GMSK /N(ABLX2),IM1,LB3,FFA,MKFF73 0217 0214 FREE 952
953 ABLX2 GEN /N(ABLX3),LB1,FFA,MF1,WR1,AA2 DOR->X FREE 953
954 ABLX3 GMSK /N(SS2),SF1,GF4,IM4,LB3,RF1,FFA,MKFF75 0212->P FREE 954
955 * 955
956 * AUTOMATIC BOOTSTRAP LOADER FOR DISC. 956
957 * DEPOSITED AT 01130, STARTED AT 01130. 957
958 * 958
959 ABL31 GMSK /N(ABL32),SF1,IMA,LB3,RF4,FFA,MK7EF1 01130 EXC 0416 FREE 959
960 ABL32 GMSK /N(ABL33),SF1,IMA,LB3,RF4,FFA,MK77F1 01131 EXC2 0016 FREE 960
961 ABL33 GMSK /N(ABL34),SF1,IMA,LB3,RF4,FFA,MK7F71 01132 EXC 0216 FREE 961
962 ***** D E C O D E R ***** 962
963 ORG X*1B4 DECODER REFERENCES NEXT 2 INSTRUCTIONS * 963
964 HLT GEN /F(HLT8),2(1),FS1 FSEL(CINTF) * 964
965 SWA40 GMSK /F(STA2-2),2(6),FSF,SF1,IM6,LB2, FSEL(13-12),OS(ALU), * 965
966 CFFA,MKF800 I&X*7FF->ALU * 966
967 SWA10 GMSK /F(LDA1),2(C),MT1,FSF,SF1,IM5, FSEL(15-13),OS(ALU), * 967
968 CLB2,RF3,FFA,MKF800 I&X*7FF->DOR * 968

```

```

969 ***** 969
970 ABL34 GMSK /N(ABL35),SF1,IMA,LB3,RF4,FFA,MKF5FE 01133 TZA FREE 970
971 ABL35 GMSK /N(ABL36),SF1,IMA,LB3,RF4,FFA,MK79B1 01134 OAR 0016 FREE 971
972 ***** D E C O D E R ***** 972
973 ORG X'1B9 DECODER REFERENCES NEXT 3 INSTRUCTIONS * 973
974 JMP GEN /T(JMP1,SS1M),TF2,SF3,GF5,IMC, 620=>JMP1,RESET CINTF,* 974
975 CRF4,XF1 CND(620) IF(MIL), INCP* 975
976 JMRK GEN /T(JMRK1,JMRK5),TF2,SF3,GF5,IMD, 620=>JMRK1, * 976
977 CXF1 CND(620) OF(MIL) * 977
978 XEC GEN /T(XEC1,XEC3),TF2,SF3,GF5,IMC,XF1 620=>XEC1,RESET CINTF,* 978
979 * CND(620) IF(MIL) * 979
980 ***** 980
981 ABL36 GMSK /N(ABL37),SF1,IMA,LB3,RF4,FFA,MK7DF1 01135 SEN 0016 FREE 981
982 ABL37 GMSK /N(ABL38),SF1,IMA,LB3,RF4,FFA,MKFD9E 01136 01141 FREE 982
983 ABL38 GMSK /N(ABL39),SF1,IMA,LB3,RF4,FFA,MKFDFF 01137 JMP FREE 983
984 ABL39 GMSK /N(ABL3A),SF1,IMA,LB3,RF4,FFA,MKFDA2 01140 01133 FREE 984
985 ABL3A GMSK /N(ABL3B),SF1,IMA,LB3,RF4,FFA,MK7AB1 01141 CIA 0016 FREE 985
986 ABL3B GMSK /N(ABL3C),SF1,IMA,LB3,RF4,FFA,MK2D89 01142 ANA 01166 FREE 986
987 ABL3C GMSK /N(ABL3D),SF1,IMA,LB3,RF4,FFA,MKDFD1 01143 JANZ FREE 987
988 ABL3D GMSK /N(ABL3E),SF1,IMA,LB3,RF4,FFA,MKFDA7 01144 01130 FREE 988
989 ABL3E GMSK /N(ABL3F),SF1,IMA,LB3,RF4,FFA,MK7FEE 01145 EXC 0021 FREE 989
990 ***** D E C O D E R ***** 990
991 ORG X'1C5 DECODER REFERENCES NEXT 3 INSTRUCTIONS * 991
992 SWA44 GMSK /F(STA2-2),2(6),FSF,SF1,IM6,LB2, FSEL(13-12),OS(ALU), * 992
993 CLA1,FF9,MKFEE0 I&X'1FF + P -> ALU * 993
994 SWA14 GMSK /F(LDA1),2(C),MT1,FSF,SF1,IM5, FSEL(15-13),OF(ALU), * 994
995 CLB2,LA1,RF3,FF9,MKFEE0 I&X'1FF + P -> DOR * 995
996 SHFT GMSK /N(SHFT1),GF2,LB2,RF7,FFA,MKFFEO I(4-0)->DOR,INCP,SAMPL* 996
997 ***** 997
998 ABL3F GMSK /N(ABL40),SF1,IMA,LB3,RF4,FFA,MK7F31 01146 EXC 0316 FREE 998
999 ABL40 GMSK /N(ABL41),SF1,IMA,LB3,RF4,FFA,MKF5BD 01147 INCR 2 FREE 999
1000 ABL41 GMSK /N(ABL42),SF1,IMA,LB3,RF4,FFA,MK7971 01150 OBR 0016 FREE A00
1001 ABL42 GMSK /N(ABL43),SF1,IMA,LB3,RF4,FFA,MK79AF 01151 OAR 0020 FREE A01
1002 ABL43 GMSK /N(ABL44),SF1,IMA,LB3,RF4,FFA,MKF9EE 01152 OME 0021 FREE A02
1003 ABL44 GMSK /N(ABL45),SF1,IMA,LB3,RF4,FFA,MKFD9B 01153 01144 FREE A03
1004 ABL45 GMSK /N(ABL46),SF1,IMA,LB3,RF4,FFA,MK7FEF 01154 EXC 0020 FREE A04
1005 ***** D E C O D E R ***** A05
1006 ORG X'1CF DECODER REFERENCES 'DP' * A06
1007 DP GMSK /N(DPRM),GF2,LB2,FFA,MKFFF8 I&7, SAMPLE *DECO A07
1008 ***** A08
1009 ABL46 GMSK /N(ABL47),SF1,IMA,LB3,RF4,FFA,MK7FF1 01155 EXC 0016 FREE A09
1010 ABL47 GMSK /N(ABL48),SF1,IMA,LB3,RF4,FFA,MK7CF1 01156 SEN 0416 FREE A10
1011 ABL48 GMSK /N(ABL49),SF1,IMA,LB3,RF4,FFA,MKFD91 01157 01156 FREE A11
1012 ABL49 GMSK /N(ABL4A),SF1,IMA,LB3,RF4,FFA,MK7AB1 01160 CIA 0016 FREE A12
1013 ABL4A GMSK /N(ABL4B),SF1,IMA,LB3,RF4,FFA,MK2D89 01162 ANA 01166 FREE A13
1014 ***** D E C O D E R ***** A14
1015 ORG X'1D5 DECODER REFERENCES NEXT 2 INSTRUCTIONS * A15
1016 SWA45 GMSK /F(STA2-2),2(6),FSF,SF1,IM6,LB2, FSEL(13-12),OS(ALU), * A16
1017 CFF9,MKFEE0,AK2 I&X'1FF + X -> ALU * A17
1018 SWA15 GMSK /F(LDA1),2(C),MT1,FSF,SF1,IM5, FSEL(15-13),OF(ALU), * A18
1019 CLB2,RF3,FF9,MKFEE0,AK2 I&X'1FF + X -> DOR * A19
1020 ***** A20
1021 ABL4B GMSK /N(ABL4C),SF1,IMA,LB3,RF4,FFA,MKDFD1 01162 JANZ FREE A21
1022 ***** D E C O D E R ***** A22
1023 ORG X'1D8 DECODER REFERENCES NEXT 8 INSTRUCTIONS * A23
1024 REG10 GEN /F(REG20),2(7),FS4,SF1,IM8,RF7,SH1 * A24
1025 REG12 GEN /F(REG20),2(7),FS4,SF1,IM8,RF7,AA1 * A25

```

```

1026 REG14 GEN /F(REG20),2(7),FS4,SF1,IM8,RF7,AA2 * A26
1027 REG16 GEN /F(REG20),2(7),FS4,SF1,IM8,RF7,FFE,BB2,AA1,MF1 * A27
1028 REG11 GEN /F(REG20),2(7),FS4,SF1,IM8,RF7 * A28
1029 REG13 GEN /F(REG20),2(7),FS4,SF1,IM8,RF7,FFE,BB1,MF1 * A29
1030 REG15 GEN /F(REG20),2(7),FS4,SF1,IM8,RF7,FFE,BB2,MF1 * A30
1031 REG17 GEN /N(REG30),RF7,FFE,BB2,AA1,MF1 * A31
1032 ***** A32
1033 EJEC A33
1034 * INSTRUCTIONS FORCED TOGETHER BY TEST ADDRESSING. A34
1035 * JDPZ1, SS1M, JMP1, BT50, SENS MUST HAVE THE SAME 5 HIGH-ORDER A35
1036 * BITS IN THEIR ADDRESSES. A36
1037 * A37
1038 MORG 16 REF BY CND FSEL(2) FROM JDPZ, JDPNZ A38
1039 JDPZ1 GEN /F(JRZ),2(1),FS7,GF2,FF0,AA1 FSEL(3),R1->ALU,SAMPLE A39
1040 JDPZ2 GEN /F(JRZ),2(1),FS7,GF2,FF0,AA5 FSEL(3),R5->ALU,SAMPLE A40
1041 MORG 2 T(JMP1,SS1M),T(BT50,SS1M),T(SENS,SS1M),S(JDPZ1,SS1M) A41
1042 SS1M GEN /N(SS2M),SF1,IM8 IF(P) A42
1043 ***** D E C O D E R ***** A43
1044 ORG X'1E3 DECODER REFERENCES 'BT10' * A44
1045 BT10 GEN /F(BT20),2(F),FS4,RF4,XF1 FSEL(3-0),INCP, * A45
1046 * RESET CINTF * A46
1047 ***** A47
1048 MORG 2 S(JDPZ1,JMP1) A48
1049 JMP1 GEN /T(JMP2,SS2M),TF2,GF8,IM5,LB1, MILS=>JMP2,OF(ALU), A49
1050 CRF1,FFA,MF1,XF2,BB1 MIL->P,JUMP SIGNAL A50
1051 ***** D E C O D E R ***** A51
1052 ORG X'1E5 DECODER REFERENCES NEXT 3 INSTRUCTIONS * A52
1053 SWA46 GMSK /F(STA2-2),2(6),FSF,SF1,IM6,LB2, FSEL(13-12),OS(ALU), * A53
1054 CFF9,MKFEO0,AK1 I&X'1FF + B -> ALU * A54
1055 SWA16 GMSK /F(LDA1),2(C),MT1,FSF,SF1,IM5, FSEL(15-13),OF(ALU), * A55
1056 CLB2,RF3,FF9,MKFEO0,AK1 I&X'1FF + B -> DOR * A56
1057 JI GEN /N(JI1),TS0,AB1,RF4,XF1 I(2-0)->BB,INCP, *DECO A57
1058 * RESET CINTF * A58
1059 ***** A59
1060 MORG 2 /T(BT51,BT52) A60
1061 BT52 GEN /N(SS3M),SF1,IM8 IF(P) A61
1062 BT51 GEN /N(BT50),SF1,IMC,RF5 IF(MIL),INCSC A62
1063 * BT50 MUST BE NEAR SS1M A63
1064 BT50 GEN /T(BT51,BT52),TF2,GF8,LB1,RF1, MILS=>BT51,MIL+1->P, A64
1065 CFF9,CF3,XF2,SH1,BB1 JUMP SIGNAL A65
1066 ***** D E C O D E R ***** A66
1067 ORG X'1EB DECODER REFERENCES 'SRE10' * A67
1068 SRE10 GEN /F(SRE20),2(3),FS4,IM1,LA1,RF3, F.SEL(1-0),WAIT(M), * A68
1069 CCF3,XF1 P+1->DOR,0->CINTF * A69
1070 ***** A70
1071 TSA1 GEN /N(TSA2),IM2 WAIT(IO) FREE A71
1072 TSA2 GEN /N(SS3M1),SF1,IM8,LB1,RF4,FFA, IF(P),INCP,IOR->A FREE A72
1073 CMF1,WR1,BB2,AA0 A73
1074 * SENS MUST BE NEAR SS1M. A74
1075 SENS GEN /T(SEN6,SEN7),TF3,GF8,LB1,RF1, MILS=>SEN7,MIL+1->P A75
1076 CFF9,CF3,SH1,BB1 A76
1077 ***** D E C O D E R ***** A77
1078 ORG X'1EF DECODER REFERENCES 'IJMP' * A78
1079 IJMP GEN /F(IJPRE),2(1),FS6,IM1,FF0,WF1, FSEL(2),0->QS, * A79
1080 CXF1,SH1 0->CINTF * A80
1081 ***** A81
1082 ABL4C GMSK /N(ABL4D),SF1,IMA,LB3,RF4,FFA,MKFDA7 01163 01130 FREE A82

```

```

1083 ABL4D GMSK /N(ABL4E),SF1,IMA,LB3,RF4,FFA,MKFDF 01164 JMP FREE A83
1084 ABL4E GMSK /N(ABL4F),SF1,IMA,LB3,RF4,FFA,MKFE7F 01165 00600 FREE A84
1085 ***** D E C O D E R ***** A85
1086 ORG X'1F3 DECODER REFERENCES NEXT 2 INSTRUCTIONS * A86
1087 ILL0P * A87
1088 EXC1 GMSK /N(EXC2),LB3,RF3,FFE,MKF7FF,16(DR) X'800 V DR -> OPR * A88
1089 ***** A89
1090 ABL4F GMSK /N(ABL50),IM1,LB3,FFA,MKFDF 01166 07760 FREE A90
1091 ABL50 GMSK /N(SS2),SF1,IM4,LB3,RF1,FFA,MKFDA7 01130->P FREE A91
1092 ***** D E C O D E R ***** A92
1093 ORG X'1F7 DECODER REFERENCES NEXT 7 INSTRUCTIONS * A93
1094 EXC21 GEN /N(EXC22),1(1),IME START I/O AT 4 * A94
1095 * NOTE - BCS WILL BE DETECTED BY WCS ON MACHINES SO EQUIPPED, * A95
1096 * RATHER THAN CAUSING ILLEGAL INSTRUCTION PROCESSING. * A96
1097 BCS ILL0P * A97
1098 SEN1 GMSK /N(SEN2),LB3,RF3,FFE,MKEFFF,16(DR) X'1000 V DR -> DOR * A98
1099 IABM1 GMSK /N(IABM2),LB3,RF3,FFE,MKDFFF,16(DR) X'2000 V DR -> DOR * A99
1100 OABM1 GMSK /F(OME1),2(3),FSA,LB3,RF3,FFE, FSEL(7-6), * B00
1101 CMKBFFF,16(DR) X'2000 V DR -> DOR * B01
1102 SETDR GEN /F(SET1),2(7),FSA,FFA,MF1,WR1, FSEL(8-6), A->DR * B02
1103 CBBO,24(DR) * B03
1104 SETDRX GEN /F(SET1),2(7),FSA,FFA,MF1,WR1, FSEL(8-6), A->DR * B04
1105 CBBO,24(DR) * B05
1106 ***** B06
1107 ***** R E S E T ***** B07
1108 ORG X'1FE REF BY FSEL(CINTF) * B08
1109 HLT8 GEN /N(RESET),RF4,XF1 RESET CINTF, INCP * B09
1110 ORG X'1FF HARDWARE INTERRUPTS HERE FOR STEP AND RESET * B10
1111 RESET GEN /N(HLT7),1(4),GF4,IMC,LB2,RF3,FFA INTRPT,1->DOR * B11
1112 ***** B12
1113 END B13
    
```

CPU:0280 CTP:024 SUPS:5.406

@BRKPT PRINT\$

102712355