1 PSM Example

```
   columns
 1 CONSTANT A_port, 00
 2 CONSTANT B_port, 01
 3 CONSTANT C_port, 02
 4 CONSTANT D_port, 03
 5 CONSTANT W_port, 01
 6 CONSTANT X_port, 02
 7 CONSTANT Y_port, 04
 8 CONSTANT Z_port, 08
 9
10 start:
11     ENABLE INTERRUPT
12     HWBUILD sF
13     JUMP test_star      ; change to test_pc to test under/overflow
14
15 test_star:
16     LOAD s0, 01
17     STAR s1, s0
18     REGBANK B
19     COMPARE s1, 01
20     JUMP NZ, error
21     REGBANK A
22     JUMP test_add
23
24 test_add:
25     LOAD s0, 01
26     ADD s0, 04
27     COMPARE s0, 05      ; check simple add, 1 + 4 = 5
28     JUMP NZ, error      ; 0 means it's equal
29     LOAD s0, 10
30     LOAD s1, 0D
31     ADD s0, s1
32     COMPARE s0, 1D      ; check 2 register add, 0x10 + 0x0D = 0x1D
33     JUMP NZ, error
34     LOAD s0, 05
35     ADD s0, FB
36     JUMP NC, error      ; check overflow = carry, 5 + 251 = 0 + carry
37     ADD s0, 01
38     COMPARE s0, 01
39     JUMP NZ, error
40     LOAD s0, 00
41     ADD s0, 00
42     JUMP NZ, error      ; check for zero flag
43     JUMP test_add_carry
44
45 test_add_carry:
46     LOAD s0, FF
47     ADD s0, 01          ; s0 = 0, carry set
48     ADDCY s0, 01            ; s0 = s0 + 1 + carry(1)
49     COMPARE s0, 02
50     JUMP NZ, error      ; s0 is not 2
```

```
51      LOAD s0, FF
52      LOAD s1, 01
53      ADD s0, s1          ; s0 = 0, carry set
54      ADDCY s0, s1        ; s0 = s0 + s1(1) + carry(1)
55      COMPARE s0, 02
56      JUMP NZ, error
57      JUMP test_sub
58
59  test_sub:
60      LOAD s0, 0A
61      SUB s0, 0A
62      JUMP NZ, error      ; s0 is supposed to be 0
63      LOAD s0, AB
64      LOAD s1, 0B
65      SUB s0, s1          ; s0 = s0(AB) - s1(0B) = A0
66      COMPARE s0, A0
67      JUMP NZ, error
68      JUMP test_sub_carry
69
70  test_sub_carry:
71      LOAD s0, 00
72      SUB s0, 01          ; s0 = 255, carry set
73      JUMP NC, error
74      SUBCY s0, FE        ; s0 = s0(255) - FE - carry(1) = 0
75      JUMP NZ, error
76      LOAD s0, 00
77      SUB s0, 01
78      LOAD s1, 0A
79      SUBCY s0, s1        ; s0 = s0(255) - s1(10) - carry(1) = 244
80      COMPARE s0, F4
81      JUMP NZ, error
82      JUMP test_logic
83
84  test_logic:
85      LOAD s0, CA
86      AND s0, 53
87      COMPARE s0, 42      ; CA and 53 = 42!! (but it's just a hex 42)
88      JUMP NZ, error
89      LOAD s0, CA
90      LOAD s1, 14
91      AND s0, s1          ; CA and 14 = 0
92      JUMP NZ, error
93      LOAD s0, FF
94      ADD s1, 01          ; carry set
95      AND s0, 01
96      JUMP C, error       ; carry was not cleared
97      LOAD s0, CA         ; -- testing or --
98      OR s0, 53
99      COMPARE s0, DB      ; CA or 53 = DB
100     JUMP NZ, error
101     LOAD s0, F0
102     LoAD s1, 0F
103     OR s0, s1           ; F0 or 0F = 0
104     JUMP Z, error
```

```
105     LOAD s0, FF
106     ADD s1, 01          ; carry set
107     OR s0, 01
108     JUMP C, error       ; carry was not cleared
109     LOAD s0, CA         ; -- testing xor --
110     XOR s0, 53
111     COMPARE s0, 99      ; CA xor 53 = 99
112     JUMP NZ, error
113     LOAD s0, F0
114     LOAD s1, F0
115     XOR s0, s1          ; F0 or F0 = 0
116     JUMP NZ, error
117     LOAD s0, FF
118     ADD s1, 01          ; carry set
119     XOR s0, 01
120     JUMP C, error       ; carry was not cleared
121     JUMP test_shift
122
123 test_shift:
124     LOAD s0, 7F
125     SL1 s0
126     JUMP C, error
127     COMPARE s0, FF
128     JUMP NZ, error
129     LOAD s0, 80
130     SL0 s0
131     JUMP NZ, error
132     SLA s0
133     COMPARE s0, 01
134     JUMP NZ, error
135     LOAD s0, 11
136     RL s0
137     COMPARE s0, 22
138     JUMP NZ, error
139     LOAD s0, 81
140     SLX s0
141     COMPARE s0, 03
142     JUMP NZ, error
143     LOAD s0, FE
144     SR1 s0
145     JUMP C, error
146     COMPARE s0, FF
147     JUMP NZ, error
148     LOAD s0, 01
149     SR0 s0
150     JUMP NZ, error
151     SRA s0
152     COMPARE s0, 80
153     JUMP NZ, error
154     LOAD s0, 22
155     RR s0
156     COMPARE s0, 11
157     JUMP NZ, error
158     LOAD s0, 81
```

```
159     SRX s0
160     COMPARE s0, C0
161     JUMP NZ, error
162     JUMP test_io
163
164 test_io:
165     LOAD s0, 01
166     LOAD s1, 02
167     LOAD s2, s1
168     LOAD s4, 1E
169     OUTPUT s0, (s2)      ; output value 01 on port 02
170     OUTPUT s1, 03        ; output value 02 on port 03
171     OUTPUTK 03, 4        ; output value 03 on port 04
172     INPUT s3, 05         ; read value on port id 05 into s3
173     INPUT s1, (s4)       ; read value on port id 1E into s1
174     OUTPUT s1, 10        ; output read value on port id 10
175     JUMP test_spm
176
177 test_spm:
178     LOAD s0, 12
179     LOAD s1, 0A
180     LOAD s2, FF
181     STORE s0, (s1)       ; write 12 into addr 0A
182     STORE s1, C3         ; should be addr 03 in a 64 byte spm
183     FETCH s3, (s1)       ; read data 12 back from addr 0A
184     COMPARE s0, s3
185     JUMP NZ, error
186     FETCH s4, 03         ; read data from previously masked addr C3 = 03
187     COMPARE s4, s1
188     JUMP NZ, error
189     JUMP test_call
190
191 inc_s00:
192     ADD s0, 01
193     LOAD s2, 05
194     LOAD&RETURN s2, 07
195
196 test_call:
197     LOAD s0, 01
198     LOAD s5, inc_s00'upper
199     LOAD s4, inc_s00'lower
200     CALL@ (s5, s4)
201     COMPARE s0, 02
202     JUMP NZ, error
203     CALL Z, inc_s00      ; zero flag still set
204     COMPARE s0, 03
205     JUMP NZ, error
206     HWBUILD s6           ; generate a carry
207     CALL NC, inc_s00
208     COMPARE s0, 03       ; carry still set, s0 should be 03
209     JUMP NZ, error       ; if inc was called (s0 = 04) ... it was wrong
210     CALL C, inc_s00      ; carry set to 0 by compare
211     COMPARE s0, 03       ; call was done ? s0 = 4 -> error
212     JUMP NZ, error
```

```
213     LOAD s1, passed'upper
214     LOAD s0, passed'lower
215     JUMP@ (s1, s0)
216
217 test_pc:
218     COMPARE sD, C9      ; random value to switch between over and underflow
            test
219     JUMP NZ, test_underflow
220     CALL test_overflow
221
222 test_overflow:
223     CALL test_overflow
224
225 test_underflow:
226     LOAD sD, C9
227     RETURN
228
229 error:
230     JUMP error
231
232 passed:
233     JUMP passed
234
235     ADDRESS 300
236 ISR:
237     REGBANK B
238     LOAD s0, FF
239     CALL inc_s00
240     RETURNI ENABLE
```