```
0:mar := pc; rd;                                { main loop  }
1:pc := 1 + pc; rd;                             { increment pc }
2:ir := mbr; if n then goto 28;                 { save, decode mbr }
3:tir := lshift(ir + ir); if n then goto 19;
4:tir := lshift(tir); if n then goto 11;        { 000x or 001x? }
5:alu := tir; if n then goto 9;                 { 0000 or 0001? }
6:mar := ir; rd;                                { 0000 = LODD }
7:rd;
8:ac := mbr; goto 0;
9:mar := ir; mbr := ac; wr;                     { 0001 = STOD }
10:wr; goto 0;
11:alu := tir; if n then goto 15;               { 0010 or 0011? }
12:mar := ir; rd;                               { 0010 = ADDD }
13:rd;
14:ac := ac + mbr; goto 0;
15:mar := ir; rd;                               { 0011 = SUBD }
16:ac := 1 + ac; rd;                { Note: x - y = x + 1 + not y }
17:a := inv(mbr);
18:ac := a + ac; goto 0;
19:tir := lshift(tir); if n then goto 25;       { 010x or 011x? }
20:alu := tir; if n then goto 23;               { 0100 or 0101? }
21:alu := ac; if n then goto 0;                 { 0100 = JPOS }
22:pc := band(ir, amask); goto 0;               { perform the jump }
23:alu := ac; if z then goto 22;                { 0101 = JZER }
24:goto 0;                                      { jump failed }
25:alu := tir; if n then goto 27;               { 0110 or 0111? }
26:pc := band(ir, amask); goto 0;               { 0110 = JUMP }
27:ac := band(ir, amask); goto 0;               { 0111 = LOCO }
28:tir := lshift(ir + ir); if n then goto 40;   { 10xx or 11xx? }
29:tir := lshift(tir); if n then goto 35;       { 100x or 101x? }
30:alu := tir; if n then goto 33;               { 1000 or 1001? }
31:a := sp + ir;                                { 1000 = LODL }
32:mar := a; rd; goto 7;
33:a := sp + ir;                                { 1001 = STOL }
34:mar := a; mbr := ac; wr; goto 10;
35:alu := tir; if n then goto 38;               { 1010 or 1011? }
36:a := sp + ir;                                { 1010 = ADDL }
37:mar := a; rd; goto 13;
38:a := sp + ir;                          { 1011 = SUBL }
39:mar := a; rd; goto 16;
40:tir := lshift(tir); if n then goto 46; { 110x or 111x? }
41:alu := tir; if n then goto 44;         { 1100 or 1101? }
42:alu := ac; if n then goto 22;          { 1100 = JNEG }
43:goto 0;
44:alu := ac; if z then goto 0;           { 1101 = JNZE }
45:pc := band(ir, amask); goto 0;
46:tir := lshift(tir); if n then goto 50;
47:sp := sp + (-1);                       { 1110 = CALL }
48:mar := sp; mbr := pc; wr;
49:pc := band(ir, amask); wr; goto 0;
50:tir := lshift(tir); if n then goto 65; { 1111, 7 bit opcode }
51:tir := lshift(tir); if n then goto 59;
52:alu := tir; if n then goto 56;
53:mar := ac; rd;                         { 1111 000 0 = PSHI }
54:sp := sp + (-1); rd;
55:mar := sp; wr; goto 10;
56:mar := sp; sp := sp + 1; rd;           { 1111 001 0 = POPI }
57:rd;
58:mar := ac; wr; goto 10;
59:alu := tir; if n then goto 62;
60:sp := sp + (-1);                       { 1111 010 0 = PUSH }
61:mar := sp; mbr := ac; wr; goto 10;
62:mar := sp; sp := sp + 1; rd;           { 1111 011 0 = POP }
```

```
63:rd;
64:ac := mbr; goto 0;
65:tir := lshift(tir); if n then goto 73;
66:alu := tir; if n then goto 70;
67:mar := sp; sp := sp + 1; rd;           { 1111 100 0 = RETN }
68:rd;
69:pc := mbr; goto 0;
70:a := ac;                               { 1111 101 0 = SWAP }
71:ac := sp;
72:sp := a; goto 0;
73:alu := tir; if n then goto 76;
74:a := band(ir, smask);                  { 1111 110 0 = INSP }
75:sp := sp + a; goto 0;
76:tir := tir + tir; if n then goto 80;
77:a := band(ir, smask);                  { 1111 111 0 = DESP }
78:a := inv(a);
79:a := a + 1; goto 75;
80:tir := tir + tir; if n then goto 97;   { 1111 111 1 1x = HALT }
81:alu := tir + tir; if n then goto 89;   { 1111 111 1 01 = RSHIFT }
82:mar := sp; a := sp + 1; rd;            { 1111 111 1 00 = NAND }
83:rd;
84:mar := a; b := mbr; rd;
85:rd;
86:c := mbr;
87:a := band(b, c);
88:ac := inv(a); goto 0;
89:a := lshift(1);                        { 1111 1111 01 = RSHIFT }
90:a := lshift(a + 1);
91:a := lshift(a + 1);
92:a := a + 1;
93:b := band(ir, a);
94:b := b + (-1); if n then goto 96;
95:ac := rshift(ac); goto 94;
96:goto 0;
97:rd; wr;                                { 1111 1111 1x = HALT }
```