

Macros

ALU[f] A,B,Y = Either ALU0 or ALU1

IF[AcB] H,L

TC A,B;

JC[c] H,L;

SUB PCH,PCL,H,L

LPC;

IN[h] PCH;

IN[l] PCL;

JC[1] H,L;

RET PCH,PCL

IF[PCL<=OVR] ADL

ALU[++] PCH,PCH;

ADL ALU[+] FORTN,PCL,PCL;

JC[1] PCH,PCL;

OVR ST H-F1;

FORTN ST D-014;

ST

This instruction places numbers in memory and does not produce code.

END

This instruction does not produce code.

Subroutines

8BC2 8-bit multiplication subroutine

A ST H-08;

```

B    ST H-AF;
C    ST H-00;
MULT ALU[=] ZERO,C;
END? IF[B>ZERO] ADDH,ADDL;
      RETS MULTH,MULTL;
ADD  ALU[+] A,C,C;
      JC[1] END?H,END?L;
ZERO ST 0;
ADDH ST _; High address of ADD
ADDL ST _; Low address of ADD
END?H ST _; High address of END?
END?L ST _; Low address of END?

```

Programs

Boot loader

```

0000 OUT[REG] H00;      Replace REG with port number of register
      ALU[=] S1H,STRDH;  where RE is connected to bit 6.
      ALU[=] S1L,STRDL;
      SUB DISSH,DISSL,DISH,DISL
      SUB KBDSH,KBDSL,KBDFH,KBDFL; Filter extra bits out of keyboard and display value.
      ALU[=] HKBD,MULTA;
      ALU[=] H10,MULTB;
      SUB PCH,PCL,MULTH,MULTL;  Multiply key by 16.
      ALU[=] MULTY,FILE;

```

Note 2/13/2018: The boot loader became a lot uglier and inelegant when I realized that the program in ROM had to write a program in RAM so that it could modify the address that pointed to where the new program was to be loaded. 8BP3 was a very large improvement over 8BC2.