8 bit machine

X nop → add reg 0,0

5 bits inst

8 regs (0-7)

256 bytes RAM

255 = screen
254 = key

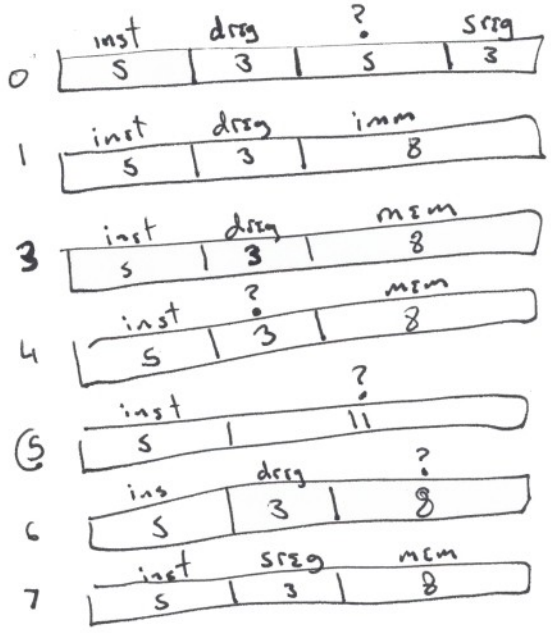
0 1		mov	0,1
2 3		add	0,1
4 5		sub	0,1
6 7		mul	0,1
8 9		div	0,1
10 11		cmp	0,1
12		jmp	4
13		je	4
14		jne	4
15		jgr	4
16		jge	4
17		jls	4
18		jle	4
19		call	4
20		ret	5
21 22		and	0,1
23 24		or	0,1
25 26		xor	0,1
	X	not	6 → xor reg, FF
27		neg	6
28		load	3
29		stor	7
30		push	6
31		pop	6

Modes reg←reg 0
reg←imm 1 = [illegible]

reg ← mem 3
mem ← imm 4
none ← return 5
reg 6
mem ← reg 7

ROM image
= 254 bytes
reset at 0

16 bits



```
 1  jmp 2
 2  var cnt 26
 3  lod r0 13
 4  mov r1 'a'
 5  str FF r1
 6  add r1 1
 7  sub r0 1
 8  cmp r0 0
 9  jgr 48
10  mov r3 0D
11  mov r4 0A
12  psh r3
13  psh r4
14  pop r6
15  pop r5
16  str r5 FF
17  str r6 FF
18  lod r7 FE
19  add r7 3
20  str r7 FF
21  ret
```

```
0100 0000 0000 0100   60 04
                      00 1A
1110 0000 0000 0011   E0 03
0000 1001 0110 0001   09 61
1110 1001 1111 1111   E9 FF
0001 1001 0000 0001   19 01
0010 1000 0000 0001   28 01
0101 1000 0000 0000   58 00
0111 1000 0000 0000   78 00

0000 1011 0000 1101   0B 0D
0000 1100 0000 1010   0C 0A
1111 0011 0000 0000   F3 00
1111 0100 0000 0000   F4 00
1111 1110 0000 0000   FE 00
1111 1101 0000 0000   FD 00
1110 1101 1111 1111   ED FF
1110 1110 1111 1111   EE FF
1110 0111 1111 1110   E7 FE
0001 1111 0000 0011   1F 03
1110 1111 1111 1111   EF FF
1010 0000 0000 0000   A0 00
```