Appendix: the PC-231 computer

Basic operation

- When turned on, all registers and RAM are set to full 12-bit zero values.
- A program (possibly including data values as well) is loaded into RAM (unloaded locations remain as zeroes).
- The PC register is used as an index into RAM to fetch an instruction and bring it into the CPU.
- The PC register is incremented by one, so as to address the next instruction in RAM.
- The instruction fetched from RAM is decoded and performed: this may move data into registers, between registers or from registers into or out of RAM. It may also compute results from two registers into one result, based on some operation (the result is left in the second register specified in the instruction).
- If a HALT instruction was executed in the last step, the machine stops; otherwise, the process continues from the 'fetch' step above (third on this list).

Instructions: their opcodes, formats and meanings

Opcode	Name	Argument format	Description of the operation
00 = x0	HALT	---- ----	Halts the machine
01 = x1	ZERO	RRRR ----	Zeroes (or "clears") out reg RRRR
02 = x2	SET	RRRR BBBB	Sets 4 low-order bits of reg RRRR to BBBB
03 = x3	DATA	BBBB BBBB	Clears DR, then sets its 8 lowest-order bits
04 = x4	INC	RRRR sNNN	Adds or subtracts between 1 and 8 from reg RRRR
05 = x5	SHIFT	RRRR sNNN	Shifts RRRR left (-) or right (+) by 1 to 8 bits
06 = x6	ADD	RRR1 RRR2	Adds the contents of reg RRR1 *into* RRRR2
07 = x7	SUB	RRR1 RRR2	Subtracts the contents of reg RRR1 *from* RRRR2
08 = x8	AND	RRR1 RRR2	Logically ANDs contents of reg RRR1 *into* RRRR2
09 = x9	COPY	RRR1 RRR2	Copies the contents of reg RRR1 to RRRR2
10 = xA	LOAD	RRR1 RRR2	Loads contents from RRR2 *(indirect)* into RRR1
11 = xB	STORE	RRR1 RRR2	Stores contents of RRR1 into RRR2 *(indirect)*
12 = xC	READ	RRRR DDDD	Reads a value from device DDDD into reg RRRR
13 = xD	WRITE	RRRR DDDD	Writes a value from reg RRRR to device DDDD
14 = xE	JPIF	RRRR CCJJ	If RRRR meets condition CC, jump to address in JJ
15 = xF	JUMP	AAAA AAAA	Jump directly to the address AAAA AAAA

Notes:

- results of ADD, SUB and AND always replace the contents of second argument (RRR2)
- the indirect role of RRR2 in LOAD and STORE means that contents of RRR2 is used an address into RAM
- for INC and SHIFT, the format sNNN means s is a sign bit and NNN a number from 1 to 8 (000 = 1, …, 111 = 8)