HERA: ***H**opefully **E**lectronics **R**unning **A**utomatically*

A slightly less simple CPU than the SPEAR. It is way more powerful due to:

- Significantly more control lines, thus more instructions
- An extra `C` register
- Direct read and write access to all registers
- Jump and branches as individual instructions (SPEAR solution was creative, but tedious)
- A 16-bit bus (i.e. jump in single cycle)
- 16-bit registers and ALU
- RAM size increased from 256 bytes to 128 KiB
- Additional NOR operation
- Multiple flags (carry out, zero, negative) in a proper status register
- Far better input/output capabilities
- Instructions without literal value fit into one byte

Features

- Full 16-bit architecture
  - 16-bit addressing for ROM and RAM (ROM: 64 KiB, RAM: 128 KiB)
    - Two RAM chips for direct 16-bit access
  - 16-bit bus, registers and ALU
- Usable digital I/O
  - 256 input ports (16-bit)
  - 256 output ports (16-bit), can be extended at expense of RAM size

Technical details

Instructions

The upper 4 bits of any instruction signify control lines for writing to the internal bus (everything but `STAT` uses 16-bit values):

Nibble	Name	Description
0	`LIT`	Literal value
1	`A`	A register
2	`B`	B register
3	`C`	C register
4	`RAM`	RAM
5	`RAM_P`	RAM pointer register
6	`PC`	Program counter register
7	`STAT`	Status register (8-bit)

Nibble	Name	Description
8	-	-
9	-	-
A	`ADD`	Adder (`A` + `B`)
B	`COM`	Two's complement (-`A`)
C	`NOR`	NOR gate ~(`A` | `B`)
D	-	-
E	-	-
F	-	-

If a literal is selected to be written to the bus (upper nibble is 0x0), it is read from the two bytes after the instruction in big-endian format.

The lower 4 bits signify control lines for reading from the internal bus:

Nibble	Name	Description
0	-	Ignore value
1	`A`	A register
2	`B`	B register
3	`C`	C register
4	`RAM`	RAM
5	`RAM_P`	RAM pointer register
6	`PC`	Program counter register
7	`STAT`	Status register (8-bit)
8	-	-
9	-	-
A	`A` `B`	A & B registers
B	`B` `RAM_P`	B & RAM_P registers
C	`C` `PC`	C & PC registers
D	-	-
E	-	-
F	-	-

Assembly code Examples

Machine code Examples

Copying the value of the `A` register into the `B` register is done by instruction `0x12` (upper nibble 1 -> write `A` to bus, lower nibble 2 -> read from bus into `B`).

Setting `B` to a literal value is done by instruction `0x02` followed by two bytes in big-endian order (i.e. `0x02 0x12 0x34` -> `B=0x1234`).