Netlist Simulator

Netlist simulator written in OCaml for Digital Systems Course (Ecole Normale Superieure, Paris).

Compilation

```
ocamlbuild ./netlist_sim.byte
```

Execution example

```
./netlist_sim.byte -net test/fulladder.net
```

Options

```
./netlist_sim.byte [-steps {n}] [-mem {*}] [-net {*}] [-input {*}] [-output {*}]
```

- `-steps {n}`: Sets the number of steps to n. Defaults to 100.
- `-mem {*}`: Sets the filepath to the RAM/ROM initialisation file. Defaults to "" (No RAM/ROM initialisation).
- `-net {*}`: Sets the filepath to the netlist. Defaults to "./net.net".
- `-input {*}`: Sets the filepath to the input file. Defaults to "" (stdin).
- `-output {*}`: Sets the filepath to the output file. Defaults to "" (stdout).
- `-nowr` : Benchmarking option (no printing).
- `-help`: Shows information on available options.

Format

Memory

The memory initialisation file should contain, on the first line, the number of variables to be initialised.

The description of each of those variables has 3 lines:

- The first line contains the name of the variable
- The second line contains the parameters of the memory block to be allocated (address_size and word_size) separated by one space
- The third line contains 2^address_size * word_size bits (without spaces) - the initial values of the allocated memory block

Example:

We want to initialize 2 ROM memories, named u and v.

u has the following parameters: (address_size = 1, word_size = 2); the first address of u contains 10 and the second address of u contains 11

v has the following parameters: (address_size = 2, word_size = 4); the first address of v contains 0000, the second address of v contains 1100, the third adress of v contains 0011, and the fourth address of v contains 1111

Then, then .mem file should look like:

```
2
u
1 2
1011
v
2 4
0000110000111111
```

Input

The input file should contain (number of steps) * (number of entries) lines. There should be (number of steps) blocks of (number of entries) lines, line i of block j representing the value of the i-th input for the j-th step. The order must be the same as in the netlist declaration of inputs.

Example:

We want to execute 5 steps of a netlist that looks like:

```
INPUT a, b
...
VAR
    ..., a, b : 2, ...
...
```

Then, the .in file should look like:

```
0
00
0
10
1
11
0
01
1
00
```

(0, 00) - value of (a, b) for the first step, (0, 10) - value of (a, b) for the second step, etc.

Notice

Memory adresses should be given in little-endian format (lowest bit should be the leftmost one; e.g. 10(2) = 1(10) and 01(2) = 2(10)).