

Coursework 1

Computer Processors (COMP1212)

The files required to complete this coursework are available on GitLab. You should complete the coursework in the files provided, **do not** rename any of the files or change the directory structure.

Submission You **must** submit your work via GitLab and also on the Minerva.

1. **GitLab:** Ensure that all changes have been committed and pushed before the deadline. You should commit frequently.
2. **Minerva:** Ensure that the coursework directory is compressed into a **.tar.gz** archive and submitted to the Minerva.

Late submissions are accepted up to 7 days late. Each day, or part of a day, will incur a 5% penalty.

Deadline TBC.

Weighting This piece of summative coursework is worth 15% of your grade.

This coursework involves implementing components of a computer processor covered in the lectures. You should provide an implementation of each of the following components in hdl. You are provided with a description of the behaviour of each of the components as well as test files to ensure it behaves according to specification.

Implement the following .hdl files

- And.hdl
- Or.hdl
- Not.hdl
- Xor.hdl
- Mux.hdl
- DMux.hdl
- And16.hdl

- Or16.hdl
- Not16.hdl
- Mux16.hdl
- Or8Way.hdl
- Mux4Way16.hdl
- Mux8Way16.hdl
- DMux4Way.hdl
- DMux8Way.hdl

The reflective account is submitted via a Minerva test.

Marking

The marking of the .hdl files is automated so it is important that you do not rename or alter the directory structure of your submission from that provided on GitLab and Minerva. Marks will be allocated in the following way:

- **10 marks** for correct implementation of the components according to their specification
- **5 marks** for minimising complexity of the implementations

Chip descriptions

Chip name: And

Inputs: x, y

Outputs: out

Function: out = 1 if $(x == 1 \text{ and } y == 1)$ 0 otherwise

Chip name: Or

Inputs: x, y

Outputs: out

Function: out = 1 if $(x == 1 \text{ or } y == 1)$ 0 otherwise

Chip name: Not

Inputs: x

Outputs: out

Function: out = not x

Chip name: Xor

Inputs: x, y

Outputs: out

Function: out = not $(x == y)$

Chip name: Mux

Inputs: x, y, sel

Outputs: out

Function: If $\text{sel} == 1$ then $\text{out} = y$ else $\text{out} = x$

Chip name: DMux

Inputs: in, sel

Outputs: x, y

Function: $\{x, y\} = \{in, 0\}$ if $\text{sel} == 0$ $\{0, in\}$ if $\text{sel} == 1$

Chip name: And16

Inputs: $x[16], y[16]$

Outputs: out[16]

Function: For $i = 0 \dots 15$ $\text{out}[i] = x[i]$ and $y[i]$

Chip name: Or16

Inputs: $x[16], y[16]$

Outputs: out[16]

Function: For $i = 0 \dots 15$ $\text{out}[i] = x[i]$ or $y[i]$

Chip name: Not16

Inputs: $x[16]$

Outputs: $\text{out}[16]$

Function: For $i = 0 \dots 15$ $\text{out}[i] = \text{not } x[i]$

Chip name: Mux16

Inputs: $x[16], y[16], \text{sel}$

Outputs: $\text{out}[16]$

Function: If $\text{sel} == 1$ then $\text{out} = y$ else $\text{out} = x$

Chip name: Or8Way

Inputs: $x[8]$

Outputs: out

Function: $\text{out} = x[0] \text{ or } x[1] \text{ or } \dots \text{ or } x[7]$

Chip name: Mux4Way16

Inputs: $w[16], x[16], y[16], z[16], sel[2]$

Outputs: $out[16]$

Function: $out = w$ if $sel == 00$

x if $sel == 01$

y if $sel == 10$

z if $sel == 11$

Chip name: Mux8Way16

Inputs: $s[16], t[16], u[16], v[16], w[16], x[16], y[16], z[16], sel[3]$

Outputs: $out[16]$

Function: $out = s$ if $sel == 000$

t if $sel == 001$

u if $sel == 010$

v if $sel == 011$

w if $sel == 100$

x if $sel == 101$

y if $sel == 110$

z if $sel == 111$

Chip name: DMux4Way

Inputs: $in, sel[2]$

Outputs: w, x, y, z

Function: $(w, x, y, z) = (in, 0, 0, 0)$ if $sel == 00$
 $(0, in, 0, 0)$ if $sel == 01$
 $(0, 0, in, 0)$ if $sel == 10$
 $(0, 0, 0, in)$ if $sel == 11$

Chip name: DMux8Way

Inputs: $in, sel[2]$

Outputs: s, t, u, v, w, x, y, z

Function: $(s, t, u, v, w, x, y, z) =$
 $(in, 0, 0, 0, 0, 0, 0, 0)$ if $sel == 000$
 $(0, in, 0, 0, 0, 0, 0, 0)$ if $sel == 001$
 etc.
 $(0, 0, 0, 0, 0, 0, 0, in)$ if $sel == 111$