SYNOPSYS®
Silicon to Software™

Verilog LAB II

Lab 3 - Simulation

- This lab will be focused on simulating each module created in Lab 2 (a, b, c)
  - Simulations will be done using VCS tool.

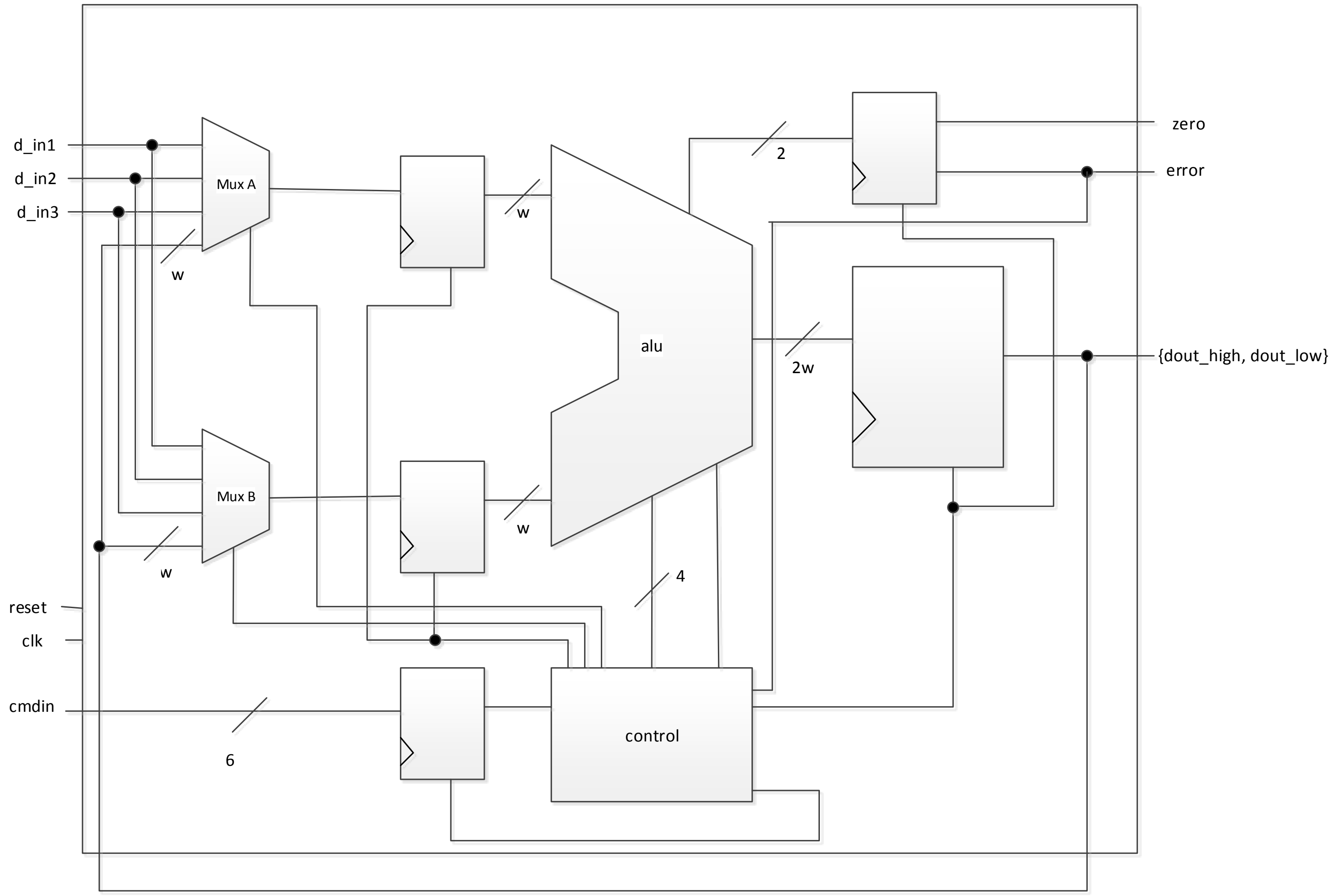
- For lab 2a (Multiplexer) there is a test bench available, so the task is to simulate the module using that test bench.

```
vcs -full_debug -sverilog mux4_tb.v <verilog_file.v>
```

- For labs 2b and 2c:
  - Write a test bench for each module.
  - Simulate them in VCS.

Lab 4 – Full design

- The objective is to create a basic multicycle processor as shown in the diagram below.
  - Modules previously designed should be used. (`Mux, ALU, Regbank`)

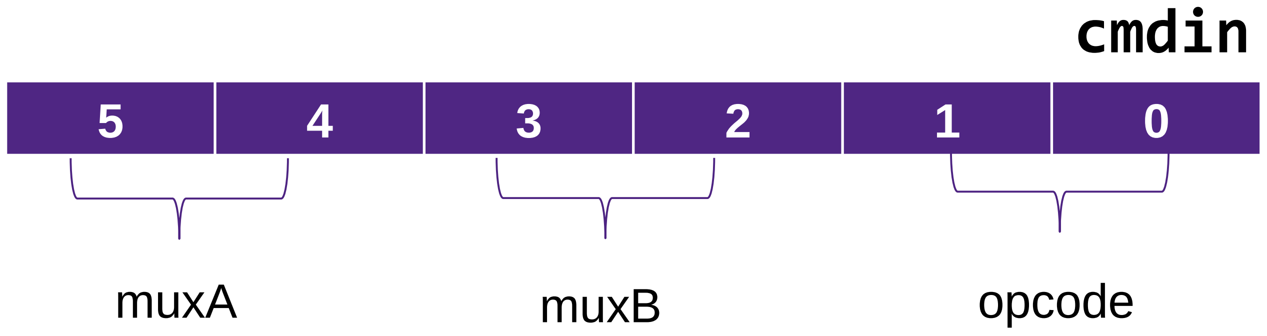


Lab 4 – Full design (cont.)

- Specifications of the design:
  - This is a multicycle processor (3 stages) with no pipelining.
  - It has 3 variable width input buses to be used as operands.
  - The **`ALU`** has 2 multiplexed inputs:
    - Each **`MUX`** is feed by the 3 input buses and the **`ALU`** output.
    - Details on how to control the **`ALU`** are available in Lab 2d.
  - Both reset (**`rst`**) and clock (**`clk`**) signals should be connected to every module that uses them.
  - It has a control unit feed by a 6 bit word (**`cmdin`**). This control unit has to:
    - Enable the register stages only when applicable
    - Select the outputs of each **`MUX`**
    - Provide the control signals for the **`ALU`**.
      - Inverted valid data signal must be asserted when data is feed from the feedback loop and the previous result was not valid (**`Error`** condition)
      - OPCODE to select the operation performed by the **`ALU`**.

Lab 4 – Full design (cont.)

- The **`control`** block shall be designed to implement the following Instruction Set Architecture (**`ISA`**):



- The **`opcode`** bus should be decoded according to the following table:

opcode[1]	opcode[0]	Operation
0	0	**`Add`**
0	1	**`Sub`**
1	0	**`Mul`**
1	1	**`Div`**

Lab 4 – Full design (cont.)

- The control module
  - Write the Verilog code for implementing this module based on the details and ISA provided previously.
  - A test bench to validate this module. It should cover all possible operations, corner cases and exercise all outputs.
  - Its interface should be as follows:

```
module control (
  input wire   [0:0]  clk,
  input wire   [0:0]  rst,
  input wire   [5:0]  cmd_in,
  input wire   [5:0]  p_error,
  output reg   [0:0]  aluin_reg_en,
  output reg   [0:0]  datain_reg_en,
  output reg   [0:0]  aluout_reg_en,
  output reg   [0:0]  nvalid_data,
  output wire  [1:0]  in_select_a,
  output wire  [1:0]  in_select_b,
  output reg   [4:0]  opcode
);
```

Lab 4 – Full design (cont.)

- Full design
  - Write the Verilog code that implements this simple microprocessor. At this point it is mostly about connecting together all the modules created so far.
  - A test bench will be provided.
  - Its interface should be as follows

```verilog
module top #(
  parameter WIDTH= 8
  ) (
  input wire  [0:0] clk,
  input wire  [0:0] rst,
  input wire  [5:0] cmdin,
  input wire  [WIDTH-1:0] din_1,
  input wire  [WIDTH-1:0] din_2,
  input wire  [WIDTH-1:0] din_3,
  output wire [WIDTH-1:0] dout_low,
  output wire [WIDTH-1:0] dout_high,
  output wire [0:0] zero,
  output wire [0:0] error
);
```

SYNOPSYS®
Silicon to Software™

Thank You