TAO_LAB#5

LAB 5: Write back stage

1. implementation

A. 32-bit mux

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer: hau tao
//
// Create Date:15:05:07 12/06/2015
// Design Name:
// Module Name:	mux_32bit
//////////////////////////////////////////////////////////////////////////////////
module mux_32bit(a,b,sel,y
	);
input[31:0] a, b;
input sel;
output[31:0] y;
assign y = sel ? a: b;

endmodule
```

B. Write back stage connection

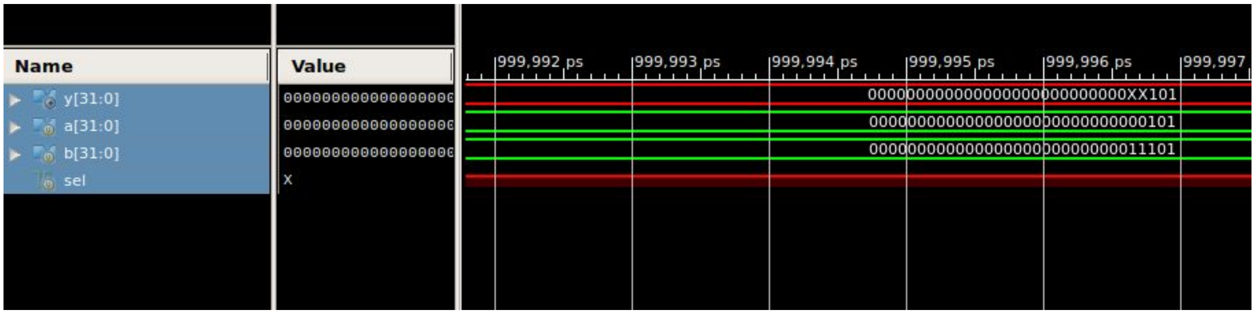
```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer: Hau Tao
//
// Create Date:15:13:33 12/06/2015
// Design Name:
// Module Name:        WBstage
//////////////////////////////////////////////////////////////////////////////////
module WBstage(input[1:0] MEM_WB,input[31:0] mem_Read_data, input[31:0]
mem_ALU_result, output wire[31:0] wb_data
           );
    wire regWrite ;
    assign regWrite = MEM_WB[1];
    mux_32bit mux_1 (
           .a(mem_Read_data),
           .b(mem_ALU_result),
           .sel(MEM_WB[0]),
           .y(wb_data)
   );


endmodule
```

2. test bench

```
`timescale 1ns / 1ps

////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer: hau tao
//
// Create Date:   15:07:43 12/06/2015
// Design Name:   mux_32bit
// Module Name:   /home/hau/Desktop/lab5/test_mux_32bit.v
////////////////////////////////////////////////////////////////////////////////

module test_mux_32bit;

    // Inputs
    reg [31:0] a;
    reg [31:0] b;
    reg sel;

    // Outputs
    wire [31:0] y;

    // Instantiate the Unit Under Test (UUT)
    mux_32bit uut (
            .a(a),
            .b(b),
            .sel(sel),
            .y(y)
    );

    initial begin
          a = 32'b01010;
          b = 32'b10101;
          sel = 1'b1;
          #10;
          a = 32'b00000;
          #10
          sel = 1'b1;
          #10;
          b = 32'b11111;
          #5;
```

```
        a= 32'b00101;
        #5;
        sel = 1'b0;
        b = 32'b11101;
   #5;
   sel = 1'bx;
        end
    always @(a or b or sel)

        #1 $display("At t = %0d sel =%b a =%b b=%b y=%b",$time, sel, a, b, y);


endmodule
```

3. simulation



```
y[31:0]  000000000000000000   00000000000000000000000000000XX101
a[31:0]  000000000000000000   00000000000000000000000000000000101
b[31:0]  000000000000000000   00000000000000000000000000000011101
sel      x
```

At t = 11 sel =1 a =00000000000000000000000000000000
b=00000000000000000000000000010101 y=00000000000000000000000000000000
At t = 31 sel =1 a =00000000000000000000000000000000
b=00000000000000000000000000011111 y=00000000000000000000000000000000
At t = 36 sel =1 a =00000000000000000000000000000101
b=00000000000000000000000000011111 y=00000000000000000000000000000101
At t = 41 sel =0 a =00000000000000000000000000000101
b=00000000000000000000000000011101 y=00000000000000000000000000011101
At t = 46 sel =x a =00000000000000000000000000000101
b=00000000000000000000000000011101 y=000000000000000000000000000xx101