LAB REPORT
CSE4010 Computer Architecture
Instructor: Lawrence Orijuela

Name: Daniel Delgado Acosta SCORE: _ __/30 Student ID: 006896598 DUE: 5/11/23 LAB: Lab4

Report

o ***What is a multiplexor? What is its purpose on the MIPS Datapath?***

A multiplexor is a logical circuit design that allows a single output selection from many input signals. The purpose of the MIPS datapath is to have a reasonable way to execute each class of MIPS instructions.

o ***Without coding one, describe in a few sentences how you would create a 16x1 multiplexor.***

A 16x1 multiplexer will comprise 4 selectors, 16 logical inputs(4 bits), and a single output. I would have 4 input variables for the selectors and another 16 for the signal inputs. One output variable would be needed and I would create the multiplexor by nesting ternary operands when creating the datapath.

Source Code for Parts A and B

Part A

mux4x1.v

```
//initialize wires

module mux_4x1(a, b, c, d, sel0, sel1, out);



   // input wires

   input a, b, c, d, sel0, sel1;

   // output wire

   output out;



   /* wire out will be equal to the turnary operand of

   sel1 to two nested turnary operand sel0, d c and sel0, b a */

   assign out = sel1 ? (sel0 ? d : c) : (sel0 ? b : a);



endmodule
```

mux4x1_tb.v

```
//creates time limit and include mux4x1.v file

`timescale 1 ns / 1 ns

`include "mux4x1.v"



//creates test bench

module mux4x1_tb;



//denoted wires

reg A, B, C, D, Sel0, Sel1;

wire O;



//creates instance of mux4x1

mux_4x1 uut(A, B, C, D, Sel0, Sel1, O);



initial begin



    //creates a separate .vvp file and .vcd file for outputs

    $dumpfile("mux4x1_tb");

    $dumpvars(0, mux4x1_tb);



    //mux 4x1
```

```
    A = 0; B = 1; C = 0; D = 1;

    {Sel1, Sel0} = 2'd0; #20;

    {Sel1, Sel0} = 2'd1; #20;

    {Sel1, Sel0} = 2'd2; #20;

    {Sel1, Sel0} = 2'd3; #20;


    //displays message to terminal

    $display("Complete!");


end


endmodule
```

Part B

mux8x1.v

```
/ //initialize wires

 module mux_8x1(a, b, c, d, e, f, g, h, sel0, sel1, sel2, out);



    // input wires

    input a, b, c, d, e, f, g, h, sel0, sel1, sel2;

    // output wire

    output out;



    /* wire out will be equal to the turnary operand of sel2 to two nested turnary
operand of sel1

    with two other nested turnary operand of sel0 and corresponding letters*/

    assign out = sel2 ? (sel1 ? (sel0 ? h : g) : (sel0 ? f : e)) : (sel1 ? (sel0 ? d : c)
: (sel0 ? b : a));



 endmodule
```

mux8x1_tb.v

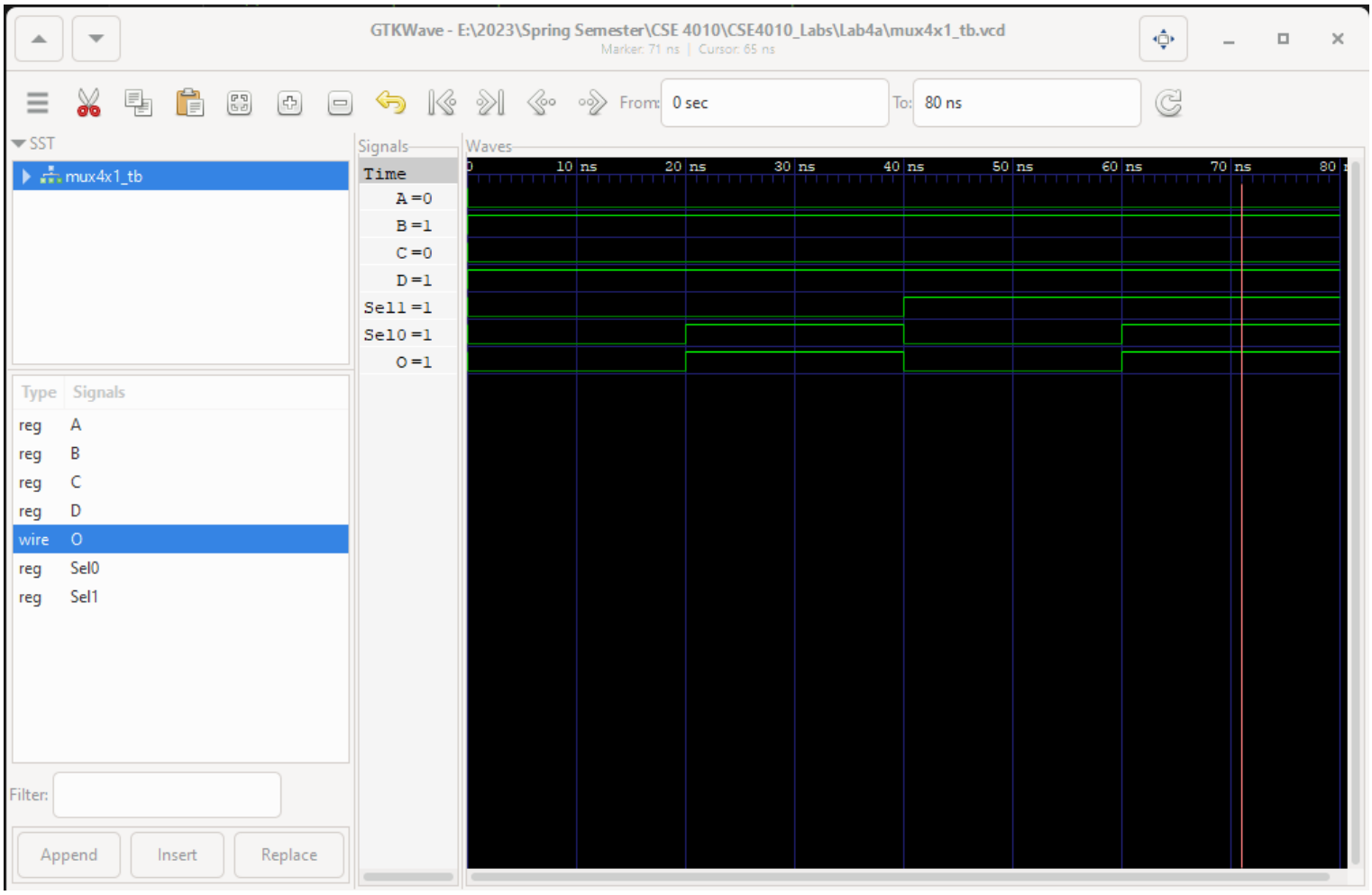
```
//creates time limit and include mux8x1.v file

`timescale 1 ns / 1 ns

`include "mux8x1.v"



//creates test bench

module mux8x1_tb;



//denoted wires

reg A, B, C, D, E, F, G, H, Sel0, Sel1, Sel2;

wire O;



//creates instance of mux8x1

mux_8x1 uut(A, B, C, D, E, F, G, H, Sel0, Sel1, Sel2, O);



initial begin



    //creates a separate .vvp file and .vcd file for outputs

    $dumpfile("mux8x1_tb");

    $dumpvars(0, mux8x1_tb);



    //mux 8x1
```

```
    A = 0; B = 1; C = 0; D = 1; E = 0; F = 1; G = 0; H = 1;

    {Sel2, Sel1, Sel0} = 3'd0; #20;

    {Sel2, Sel1, Sel0} = 3'd1; #20;

    {Sel2, Sel1, Sel0} = 3'd2; #20;

    {Sel2, Sel1, Sel0} = 3'd3; #20;

    {Sel2, Sel1, Sel0} = 3'd4; #20;

    {Sel2, Sel1, Sel0} = 3'd5; #20;

    {Sel2, Sel1, Sel0} = 3'd6; #20;

    {Sel2, Sel1, Sel0} = 3'd7; #20;


    //displays message to terminal

    $display("Complete!");


end


endmodule
```

Screenshots for Parts A and B

Part A

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL                                  Verilog

PS E:\2023\Spring Semester\CSE 4010\CSE4010_Labs\Lab4a> iverilog -o mux4x1_tb.vvp mux4x1_tb.v
No top level modules, and no -s option.
PS E:\2023\Spring Semester\CSE 4010\CSE4010_Labs\Lab4a> vvp mux4x1_tb.vvp
mux4x1_tb.vvp: Unable to open input file.
Complete!
PS E:\2023\Spring Semester\CSE 4010\CSE4010_Labs\Lab4a> gtkwave

GTKWave Analyzer v3.3.108 (w)1999-2020 BSI

GTKWAVE | Use the -h, --help command line flags to display help.

GTKWave Analyzer v3.3.108 (w)1999-2020 BSI

[0] start time.
[80] end time.
```



Part B

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL                                  Verilog

PS E:\2023\Spring Semester\CSE 4010\CSE4010_Labs\Lab4b> iverilog -o mux8x1_tb.vvp mux8x1_tb.v
mux8x1_tb.v:22: syntax error
PS E:\2023\Spring Semester\CSE 4010\CSE4010_Labs\Lab4b> vvp mux8x1_tb.vvp
VCD info: dumpfile mux8x1_tb.vcd opened for output.
Complete!
PS E:\2023\Spring Semester\CSE 4010\CSE4010_Labs\Lab4b> gtkwave

GTKWave Analyzer v3.3.108 (w)1999-2020 BSI

GTKWAVE | Use the -h, --help command line flags to display help.
```