**CA Lab 08 Report**

**Task01**

module Instruction\_Memory

(

input[63:0] Inst\_Address,

output reg[31:0] Instruction

);

reg[7:0] Inst\_Mem[15:0];

//initialize acc to fig:8.3

initial

begin

Inst\_Mem[0] = 8'b10000011;

Inst\_Mem[1] = 8'b00110100;

Inst\_Mem[2] = 8'b00000101;

Inst\_Mem[3] = 8'b00001111;

Inst\_Mem[4] = 8'b10110011;

Inst\_Mem[5] = 8'b10000100;

Inst\_Mem[6] = 8'b10011010;

Inst\_Mem[7] = 8'b00000000;

Inst\_Mem[8] = 8'b10010011;

Inst\_Mem[9] = 8'b10000100;

Inst\_Mem[10] = 8'b00010100;

Inst\_Mem[11] = 8'b00000000;

Inst\_Mem[12] = 8'b00100011;

Inst\_Mem[13] = 8'b00111000;

Inst\_Mem[14] = 8'b10010101;

Inst\_Mem[15] = 8'b00001110;

end

always@(\*)

begin

//concatenate Inst\_Mem[Inst\_Address+3:Inst\_Address] as output Instruction

assign Instruction = { Inst\_Mem[Inst\_Address+3], Inst\_Mem[Inst\_Address+2] ,Inst\_Mem[Inst\_Address+1], Inst\_Mem[Inst\_Address]};

end

endmodule

module tb

(

);

reg[63:0] Inst\_Address;

wire[31:0] Instruction;

Instruction\_Memory instruction\_memory

(

.Inst\_Address(Inst\_Address),

.Instruction(Instruction)

);

initial

begin

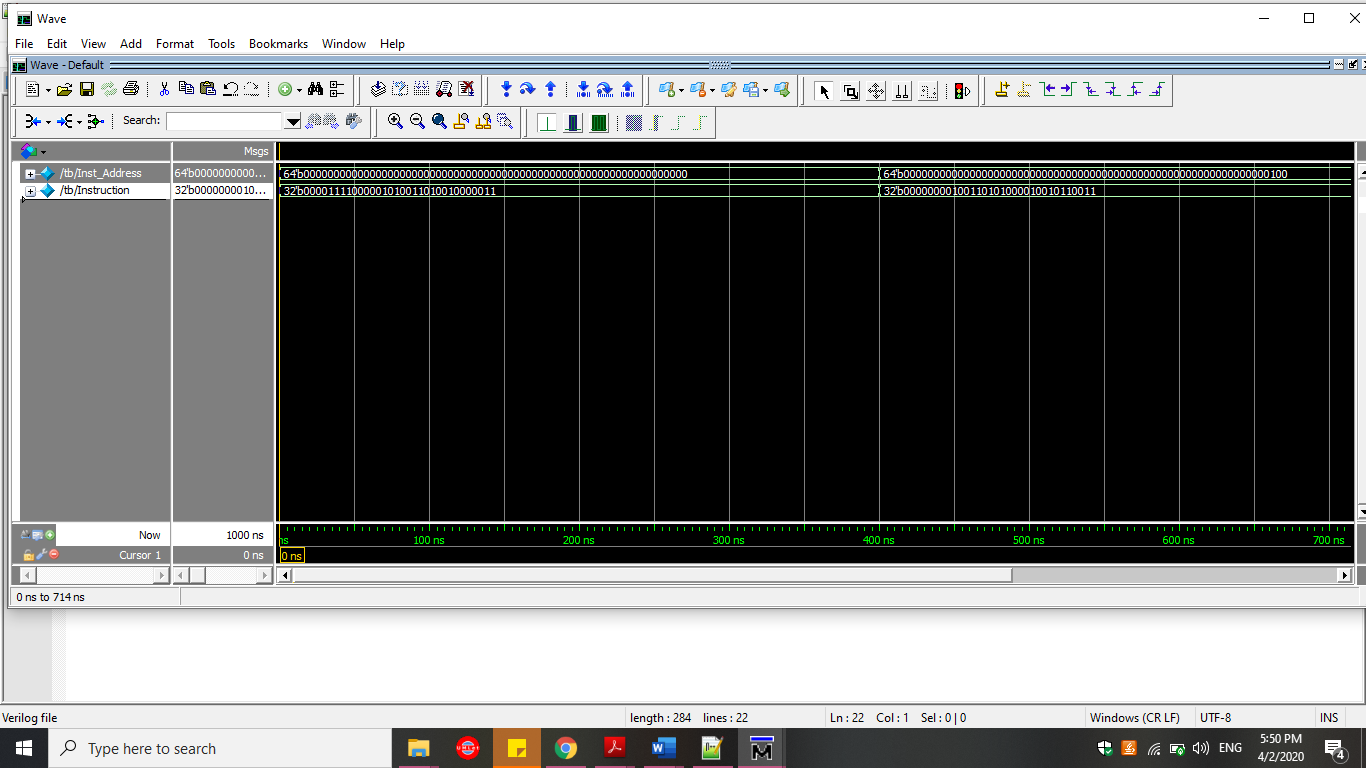
Inst\_Address = 64'd0;

#400

Inst\_Address = 64'd4;

end

endmodule



**Exercise**

module Data\_Memory

(

input[63:0] Mem\_Addr, Write\_Data,

input clk, Mem\_Write, Mem\_Read,

output reg[63:0] Read\_Data

);

reg [7:0] DM [63:0];

integer i;

initial

begin

for (i=0; i<64; i=i+1)

DM[i] = i;

end

always@(posedge clk)

begin

if (Mem\_Write == 1)

begin

DM[Mem\_Addr] = Write\_Data[7:0];

DM[Mem\_Addr+1] = Write\_Data[15:8];

DM[Mem\_Addr+2] = Write\_Data[23:16];

DM[Mem\_Addr+3] = Write\_Data[31:17];

DM[Mem\_Addr+4] = Write\_Data[39:18];

DM[Mem\_Addr+5] = Write\_Data[47:40];

DM[Mem\_Addr+6] = Write\_Data[55:48];

DM[Mem\_Addr+7] = Write\_Data[63:56];

end

end

always@(Mem\_Addr or Mem\_Read)

begin

if (Mem\_Read == 1)

begin

Read\_Data = { DM[Mem\_Addr+7], DM[Mem\_Addr+6], DM[Mem\_Addr+5], DM[Mem\_Addr+4], DM[Mem\_Addr+3], DM[Mem\_Addr+2], DM[Mem\_Addr+1], DM[Mem\_Addr] };

end

end

endmodule

module tb

(

);

reg[63:0] Mem\_Addr, Write\_Data;

reg clk, Mem\_Write, Mem\_Read;

wire[63:0] Read\_Data;

Data\_Memory data\_memory

(

.Mem\_Addr(Mem\_Addr),

.Write\_Data(Write\_Data),

.clk(clk),

.Mem\_Write(Mem\_Write),

.Mem\_Read(Mem\_Read),

.Read\_Data(Read\_Data)

);

initial

begin

clk = 0;

Write\_Data = 64'd10;

Mem\_Addr = 64'd0;

Mem\_Write = 1;

Mem\_Read = 1;

#100

Mem\_Addr = 64'd8;

Mem\_Read = 0;

#100

Write\_Data = 64'd4;

#50

Mem\_Read = 1;

end

always

#50 clk = ~clk;

endmodule

