Problem 1 (Gate Level Bench Screenshot)

Problem 1 (Gate Decoder Code)

Project Summary | Decoder_Behavioral.v | tb_decoder.v | **Decoder_Gate.v** | tb_gate.v

C:/Users/oscao/Assignment 2_1/Assignment 2_1.srcs/sources_1/new/Decoder_Gate.v

```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////////////////////////////////////////////////
3  // Company: FPGA & HDL
4  // Engineer: Oscar Osornio
5  //
6  // Create Date: 07/02/2022 02:29:07 PM
7  // Design Name: Assignment 2
8  // Module Name: Decoder_Gate
9  // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////////////////////////////////////////////////
21
22
23 module Decoder_Gate(
24     input A, B, C, D,
25     output zero, one, two, three, four, five, six, seven, eight,
26                 nine, ten, eleven, twelve, thirteen
27     );
28
29     wire A_not, B_not, C_not, D_not;
30
```

Project Summary | Decoder_Behavioral.v | tb_decoder.v | **Decoder_Gate.v** | tb_gate.v

C:/Users/oscao/Assignment 2_1/Assignment 2_1.srcs/sources_1/new/Decoder_Gate.v

```
22
23  module Decoder_Gate(
24      input A, B, C, D,
25      output zero, one, two, three, four, five, six, seven, eight,
26                  nine, ten, eleven, twelve, thirteen
27      );
28
29      wire A_not, B_not, C_not, D_not;
30
31      not NOT1(A_not, A);
32      not NOT2(B_not, B);
33      not NOT3(C_not, C);
34      not NOT4(D_not, D);
35
36      and AND13(thirteen, A_not, B_not, C_not, D_not);
37      and AND12(twelve, A_not, B_not, C_not, D);
38      and AND11(eleven, A_not, B_not, C, D_not);
39      and AND10(ten, A_not, B_not, C, D);
40      and AND9(nine, A_not, B, C_not, D_not);
41      and AND8(eight, A_not, B, C_not, D);
42      and AND7(seven, A_not, B, C, D_not);
43      and AND6(six, A_not, B, C, D);
44      and AND5(five, A, B_not, C_not, D_not);
45      and AND4(four, A, B_not, C, D_not);
46      and AND3(three, A, B_not, C, D);
47      and AND2(two, A, B, C_not, D_not);
48      and AND1(one, A, B, C_not, D);
49      and AND0(zero, A, B, C, D);
50
51  endmodule
```

Problem 1 (Gate Test Bench Code)

Project Summary | Decoder_Behavioral.v | tb_decoder.v | Decoder_Gate.v | **tb_gate.v**

C:/Users/oscao/Assignment 2_1/Assignment 2_1.srcs/sim_1/new/tb_gate.v

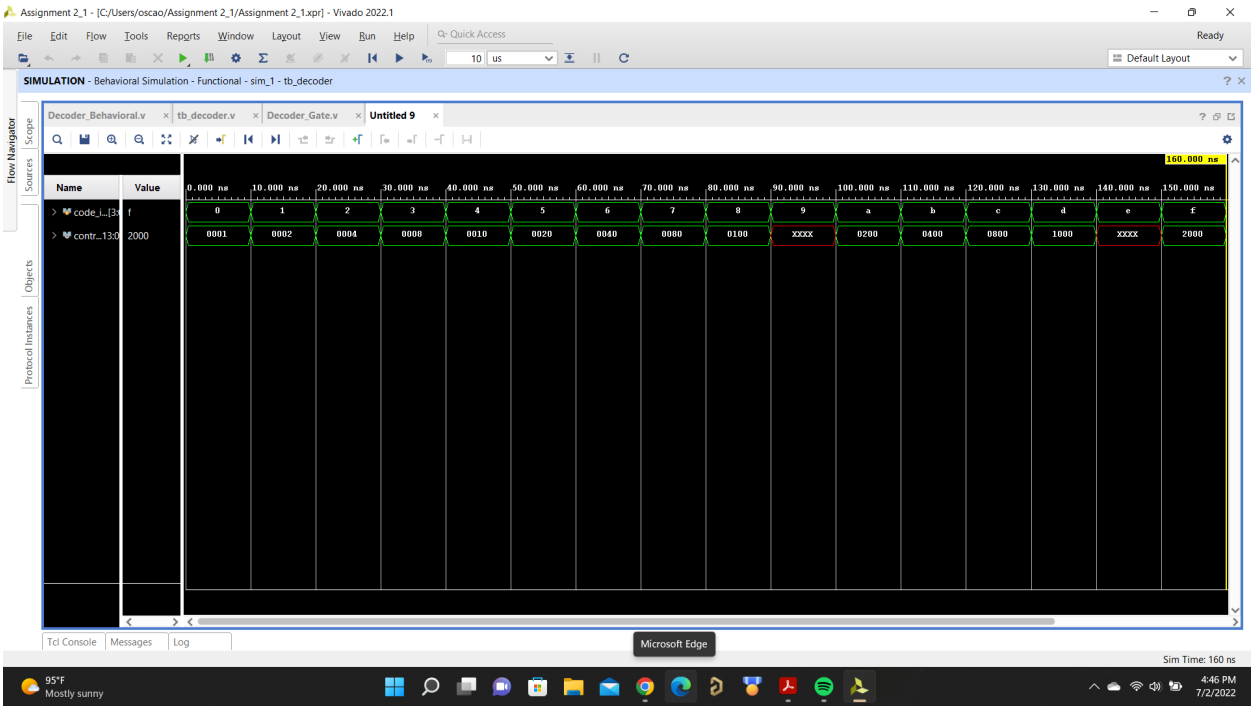
```
1   `timescale 1ns / 1ps
2   //////////////////////////////////////////////////////////////////////////////////
3   // Company: FPGA & HDL
4   // Engineer: Oscar Osornio
5   //
6   // Create Date: 07/02/2022 05:08:52 PM
7   // Design Name: Assignment 2 Problem 1
8   // Module Name: tb_gate
9   // Project Name:
10  // Target Devices:
11  // Tool Versions:
12  // Description:
13  //
14  // Dependencies:
15  //
16  // Revision:
17  // Revision 0.01 - File Created
18  // Additional Comments:
19  //
20  //////////////////////////////////////////////////////////////////////////////////
21
22
23  module tb_gate();
24
25      //declare test bench module variables
26      reg A_input, B_input, C_input, D_input;
27      wire zero, one, two, three, four, five, six, seven, eight,
28           nine, ten, eleven, twelve, thirteen;
29
30      //instantiate the design module and connect to the testbench variables
31      Decoder_Gate instantiation(.A(A_input), .B(B_input), .C(C_input), .D(D_input),
32                                 zero(zero),  one(one),   two(two),   three(three)
```

Project Summary | Decoder_Behavioral.v | tb_decoder.v | Decoder_Gate.v | tb_gate.v

C:/Users/oscao/Assignment 2_1/Assignment 2_1.srcs/sim_1/new/tb_gate.v

```
                            .zero(zero), .one(one), .two(two), .three(three),
                            .four(four), .five(five), .six(six), .seven(seven),
                            .eight(eight), .nine(nine), .ten(ten), .eleven(eleven),
                            .twelve(twelve), .thirteen(thirteen));

    initial
        begin
            $dumpfile("xyz.vcd");
            $dumpvars;

            //set stimulus to test the code
            A_input = 0;
            B_input = 0;
            C_input = 0;
            D_input = 0;
            #160 $finish;
        end

    //provide the toggling input (truth table)
    //this acts as the clock input
    always #80 A_input = ~A_input;
    always #40 B_input = ~B_input;
    always #20 C_input = ~C_input;
    always #10 D_input = ~D_input;

    //display output if there's a change in input event
    always @(A_input or B_input or C_input or D_input)
        $monitor("At TIME(in ns) = %t, input = %b%b%b%b, output = %b%b%b%b%b%b%b%b%b%b%b%b%b%b%b%b",
                 $time, A_input, B_input, C_input, D_input, zero, one, two, three,
                 four, five, six, seven, eight, nine, ten, eleven, twelve, thirteen);

endmodule
```

Problem 1 (Behavioral Test Bench Screenshot)



Problem 1 (Behavioral Decoder Code)

Project Summary | Decoder_Behavioral.v | tb_decoder.v | Decoder_Gate.v

C:/Users/oscao/Assignment 2_1/Assignment 2_1.srcs/sources_1/new/Decoder_Behavioral.v

```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////////////////////////////////////////////////
3  // Company: FPGA & HDL
4  // Engineer: Oscar Osornio
5  //
6  // Create Date: 07/02/2022 02:29:07 PM
7  // Design Name: Assignment 2 Problem 1
8  // Module Name: Decoder_Behavioral
9  // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////////////////////////////////////////////////
21
22
23 module Decoder_Behavioral(
24     input wire [3:0] op_code,
25     output reg [13:0] control
26     );
27
28     always @(op_code)
```

Project Summary | Decoder_Behavioral.v | tb_decoder.v | Decoder_Gate.v

C:/Users/oscao/Assignment 2_1/Assignment 2_1.srcs/sources_1/new/Decoder_Behavioral.v

```
18 // Additional Comments:
19 //
20 //////////////////////////////////////////////////////////////////////////////////
21
22
23 module Decoder_Behavioral(
24     input wire [3:0] op_code,
25     output reg [13:0] control
26     );
27
28     always @(op_code)
29         begin
30             if (op_code == 0)  control = 14'b00000000000001; else
31             if (op_code == 1)  control = 14'b00000000000010; else
32             if (op_code == 2)  control = 14'b00000000000100; else
33             if (op_code == 3)  control = 14'b00000000001000; else
34             if (op_code == 4)  control = 14'b00000000010000; else
35             if (op_code == 5)  control = 14'b00000000100000; else
36             if (op_code == 6)  control = 14'b00000001000000; else
37             if (op_code == 7)  control = 14'b00000010000000; else
38             if (op_code == 8)  control = 14'b00000100000000; else
39             if (op_code == 10) control = 14'b00001000000000; else
40             if (op_code == 11) control = 14'b00010000000000; else
41             if (op_code == 12) control = 14'b00100000000000; else
42             if (op_code == 13) control = 14'b01000000000000; else
43             if (op_code == 15) control = 14'b10000000000000; else
44                                control = 14'bx;
45         end
46
47 endmodule
48
```

Problem 1 (Behavioral Test Bench Code)

Project Summary | Decoder_Behavioral.v | tb_decoder.v | Decoder_Gate.v

C:/Users/oscao/Assignment 2_1/Assignment 2_1.srcs/sim_1/new/tb_decoder.v

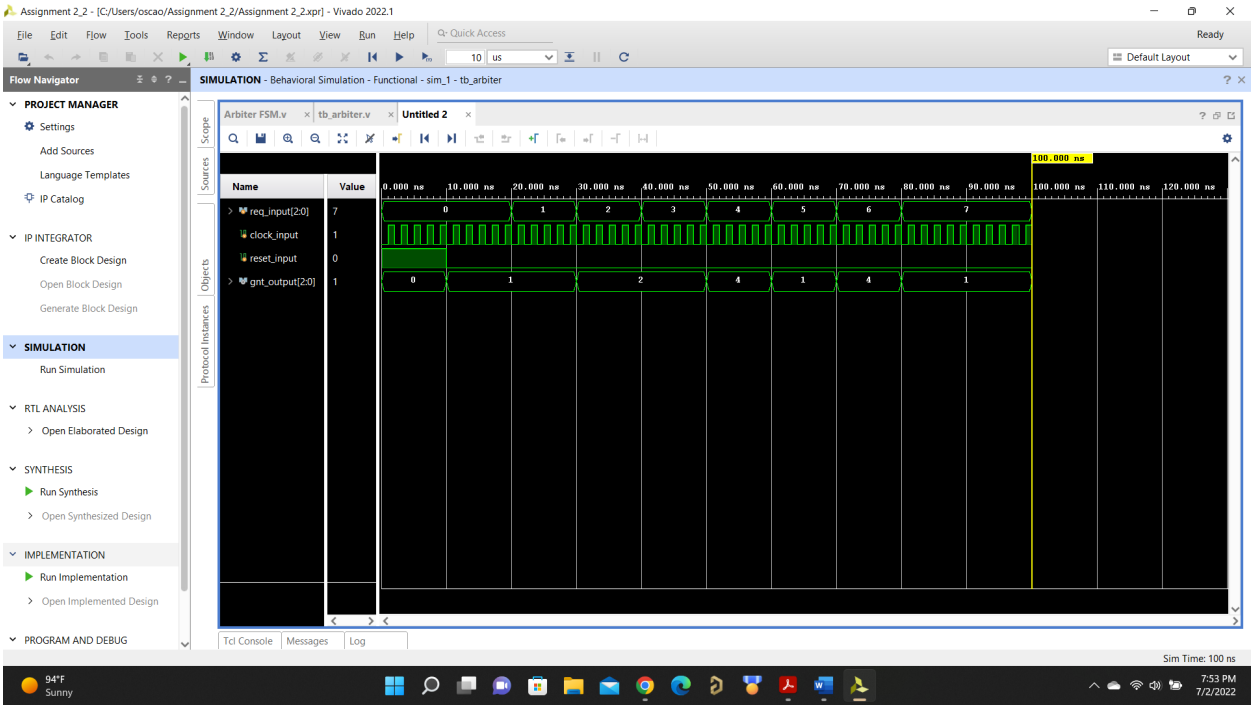
```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: FPGA & HDL
// Engineer: Oscar Osornio
//
// Create Date: 07/02/2022 02:55:54 PM
// Design Name: ASsignment 2 Problem 1
// Module Name: tb_decoder
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module tb_decoder();

    //declare testbench variables
    reg [3:0] code_input;
    wire [13:0] control_output;

    //instantiate the design module and connect to the testbench variables
    Decoder_Behavioral instantiation(.op_code(code_input), .control(control_output));

    initial
```

Project Summary | Decoder_Behavioral.v | tb_decoder.v | Decoder_Gate.v

C:/Users/oscao/Assignment 2_1/Assignment 2_1.srcs/sim_1/new/tb_decoder.v

```
    initial
        begin
            $dumpfile("xyz.vcd");
            $dumpvars;

            //set stimulus to test the code
            code_input = 4'b0000;
            #10 code_input = 4'b0001;
            #10 code_input = 4'b0010;
            #10 code_input = 4'b0011;
            #10 code_input = 4'b0100;
            #10 code_input = 4'b0101;
            #10 code_input = 4'b0110;
            #10 code_input = 4'b0111;
            #10 code_input = 4'b1000;
            #10 code_input = 4'b1001;
            #10 code_input = 4'b1010;
            #10 code_input = 4'b1011;
            #10 code_input = 4'b1100;
            #10 code_input = 4'b1101;
            #10 code_input = 4'b1110;
            #10 code_input = 4'b1111;
            #10 $finish;
        end

    //display output if there's a change in input event
    always @(code_input)
        $monitor("At TIME(in ns) = %t, op_code = %b, control = %b",
                 $time, code_input, control_output);

endmodule
```

Problem 2 (Test Bench Screenshot)



Problem 2 (Arbiter FSM Code)

Project Summary | Arbiter FSM.v | tb_arbiter.v

C:/Users/oscao/Assignment 2_2/Assignment 2_2.srcs/sources_1/new/Arbiter FSM.v

```
1  `timescale 1ns / 1ps
2  //////////////////////////////////////////////////////////////////////////////////
3  // Company: FPGA & Verilog HDL
4  // Engineer: Oscar Osornio
5  //
6  // Create Date: 07/01/2022 02:01:13 PM
7  // Design Name: Assignment 2 Problem 2
8  // Module Name: Arbiter FSM
9  // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Desoription:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////////////////////////////////////////////////
21
22
23 module Arbiter_FSM(
24     input wire [2:0] req,
25     input wire reset, clock,
26     output reg [2:0] gnt
27     );
28
29     always @(negedge clock, negedge reset)
30         begin
31             if (reset)
```

Project Summary | Arbiter FSM.v | tb_arbiter.v

C:/Users/oscao/Assignment 2_2/Assignment 2_2.srcs/sources_1/new/Arbiter FSM.v

```
18  // Additional Comments:
19  //
20  //////////////////////////////////////////////////////////////////////////////////
21
22
23  module Arbiter_FSM(
24      input wire [2:0] req,
25      input wire reset, clock,
26      output reg [2:0] gnt
27      );
28
29      always @(negedge clock, negedge reset)
30          begin
31              if (reset)
32                  gnt <= 3'b000;
33              else
34              begin
35                  case ({req})
36                      3'b000: begin gnt <= 3'b001; end
37                      3'b001: begin gnt <= 3'b001; end
38                      3'b010: begin gnt <= 3'b010; end
39                      3'b011: begin gnt <= 3'b010; end
40                      3'b100: begin gnt <= 3'b100; end
41                      3'b101: begin gnt <= 3'b001; end
42                      3'b110: begin gnt <= 3'b100; end
43                      3'b111: begin gnt <= 3'b001; end
44                  endcase
45              end
46          end
47  endmodule
```

Problem 2 (Arbiter FSM Test Bench Code)

Project Summary | Arbiter FSM.v | tb_arbiter.v

C:/Users/oscao/Assignment 2_2/Assignment 2_2.srcs/sim_1/new/tb_arbiter.v

```
1   `timescale 1ns / 1ps
2   //////////////////////////////////////////////////////////////////////////////////
3   // Company: FPGA & Verilog HDL
4   // Engineer: Oscar Osornio
5   //
6   // Create Date: 07/02/2022 06:54:53 PM
7   // Design Name: Assignment 2 Problem 2
8   // Module Name: tb_arbiter
9   // Project Name:
10  // Target Devices:
11  // Tool Versions:
12  // Description:
13  //
14  // Dependencies:
15  //
16  // Revision:
17  // Revision 0.01 - File Created
18  // Additional Comments:
19  //
20  //////////////////////////////////////////////////////////////////////////////////
21
22
23  module tb_arbiter();
24
25      //declare testbench variables
26      reg [2:0] req_input;
27      reg clock_input, reset_input;
28      wire [2:0] gnt_output;
29
```

Project Summary | Arbiter FSM.v | tb_arbiter.v

C:/Users/oscao/Assignment 2_2/Assignment 2_2.srcs/sim_1/new/tb_arbiter.v

```
20  //////////////////////////////////////////////////////////////////////////////////
21
22
23  module tb_arbiter();
24
25      //declare testbench variables
26      reg [2:0] req_input;
27      reg clock_input, reset_input;
28      wire [2:0] gnt_output;
29
30      //instantiate the design module and connect to the testbench module variables
31      Arbiter_FSM instantiation(.req(req_input), .clock(clock_input), .reset(reset_input),
32                                .gnt(gnt_output));
33
34      //generate the clock
35      initial
36          begin
37              clock_input = 1'b0;
38              forever #1 clock_input = ~clock_input;
39          end
40
41      //generate the reset
42      initial
43          begin
44              reset_input = 1'b1;
45              #10
46              reset_input = 1'b0;
47          end
48
49      initial
```

Project Summary | Arbiter FSM.v | tb_arbiter.v

C:/Users/oscao/Assignment 2_2/Assignment 2_2.srcs/sim_1/new/tb_arbiter.v

```
41      //generate the reset
42      initial
43          begin
44              reset_input = 1'b1;
45              #10
46              reset_input = 1'b0;
47          end
48
49      initial
50          begin
51              $dumpfile("xyz.vcd");
52              $dumpvars;
53
54              //set stimulus to test the code
55              req_input = 3'b000; #20
56              req_input = 3'b001; #10
57              req_input = 3'b010; #10
58              req_input = 3'b011; #10
59              req_input = 3'b100; #10
60              req_input = 3'b101; #10
61              req_input = 3'b110; #10
62              req_input = 3'b111; #20
63              $finish;
64          end
65
66      //display output if there's a change in input event
67      always @(req_input)
68          $monitor("At TIME(in ns) = %t, req = %b, gnt = %b",
69                   $time, req_input, gnt_output);
70
71  endmodule
```