Assignment 2 Extended

By: Abdelrahman Maher Hassan

Question 1

Codes:

generated_conditional_construction.v

```
module generated_conditional_construction(A, B);

    parameter Use_Gray; // if USE_Gray == 1 preform gray encoding else preform one-hot encoding
    input  [2:0]A;
    output reg [6:0]B;

    always@(A) begin
        if(Use_Gray)begin
            case(A)
                0: B = 7'b000;
                1: B = 7'b001;
                2: B = 7'b011;
                3: B = 7'b010;
                4: B = 7'b110;
                5: B = 7'b111;
                6: B = 7'b101;
                7: B = 7'b100;
                default:B = 0;
            endcase
        end

        else begin
            case(A)
                0: B = 7'b0000000;
                1: B = 7'b0000001;
                2: B = 7'b0000010;
                3: B = 7'b0000100;
                4: B = 7'b0001000;
                5: B = 7'b0010000;
                6: B = 7'b0100000;
                7: B = 7'b1000000;
                default:B = 0;
            endcase
        end
    end
endmodule
```

Test Bench:

We can make it in the same file by using generate block but in the instructions, it is stated to use 2 test bench files.

One Hot

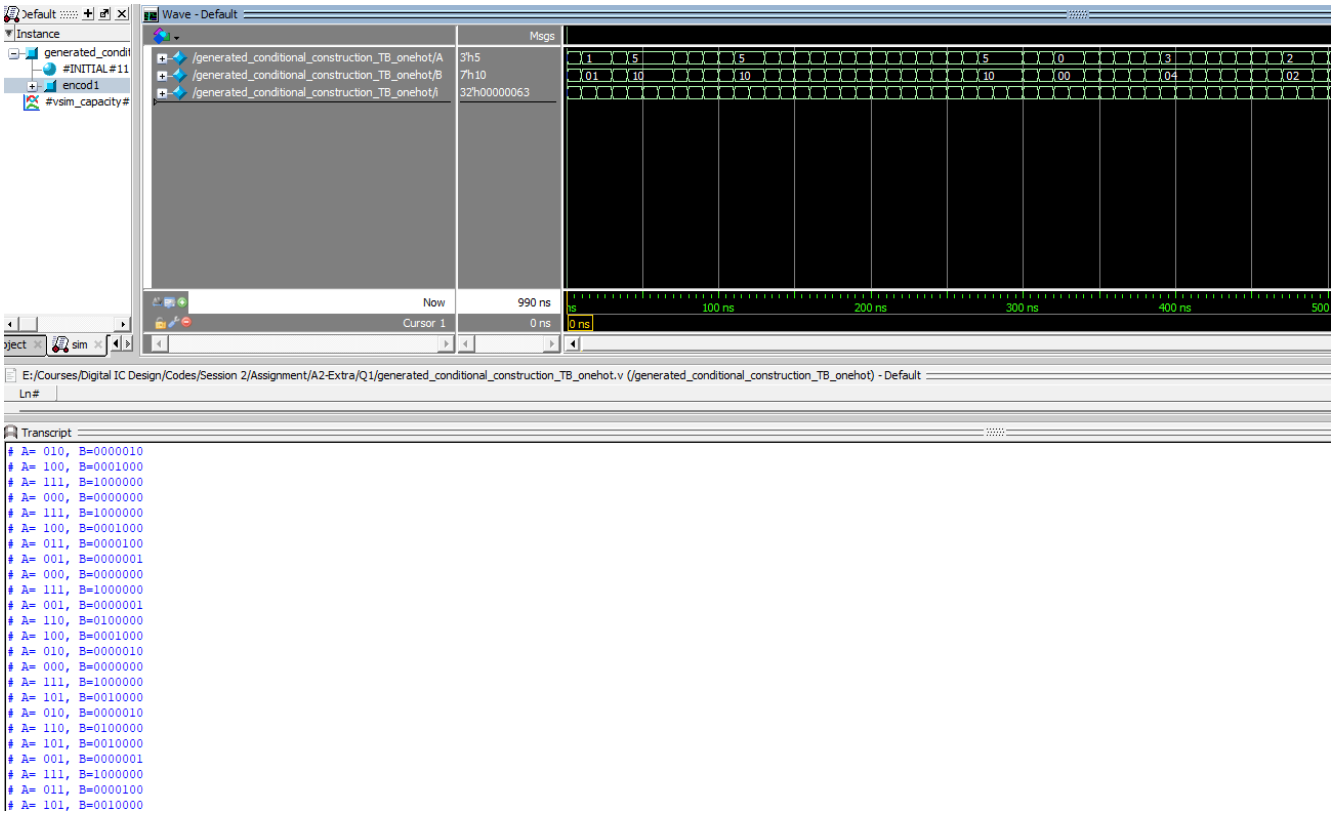
```
module generated_conditional_construction_TB_onehot();

    reg  [2:0]A;
    wire [6:0]B;
    parameter encodeing_type = 0;

    generated_conditional_construction #(encodeing_type) encod1 (A, B);

    integer i;

    initial begin
        for(i=0;i<99;i=i+1) begin
            A = $random;
            #10;
        end
        $stop;
    end

    initial begin
        $monitor("A= %b, B=%b",A,B);
    end

endmodule
```

Gray coding

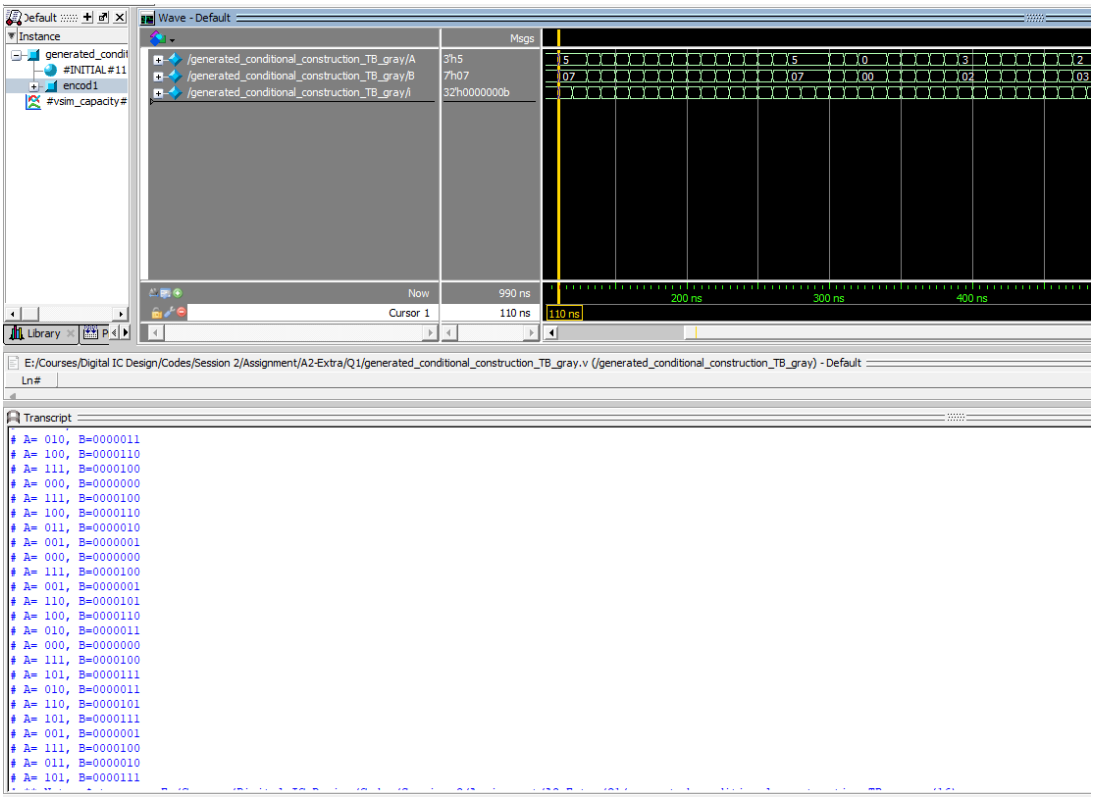
```
module generated_conditional_construction_TB_gray();

    reg  [2:0]A;
    wire [6:0]B;
    parameter encodeing_type = 1;

    generated_conditional_construction #(encodeing_type) encod1 (A, B);

    integer i;

    initial begin
        for(i=0;i<99;i=i+1) begin
            A = $random;
            #10;
        end
        $stop;
    end

    initial begin
        $monitor("A= %b, B=%b",A,B);
    end

endmodule
```

Simulation:

One hot



Gray coding:


Default
Instance
generated_condit
#INITIAL#11
encod1
#vsim_capacity#
Library
Wave - Default
Msgs
/generated_conditional_construction_TB_gray/A
/generated_conditional_construction_TB_gray/B
/generated_conditional_construction_TB_gray/i
3'h5
7'h07
32'h0000000b
Now
990 ns
Cursor 1
110 ns
200 ns
300 ns
400 ns
110 ns
E:/Courses/Digital IC Design/Codes/Session 2/Assignment/A2-Extra/Q1/generated_conditional_construction_TB_gray.v (/generated_conditional_construction_TB_gray) - Default
Ln#
Transcript
A= 010, B=0000011
A= 100, B=0000110
A= 111, B=0000100
A= 000, B=0000000
A= 111, B=0000100
A= 100, B=0000110
A= 011, B=0000010
A= 001, B=0000001
A= 000, B=0000000
A= 111, B=0000100
A= 001, B=0000001
A= 110, B=0000101
A= 100, B=0000110
A= 010, B=0000011
A= 000, B=0000000
A= 111, B=0000100
A= 101, B=0000111
A= 010, B=0000011
A= 110, B=0000101
A= 101, B=0000111
A= 001, B=0000001
A= 111, B=0000100
A= 011, B=0000010
A= 101, B=0000111

Question 2

Codes:

one_to_four_demux.v

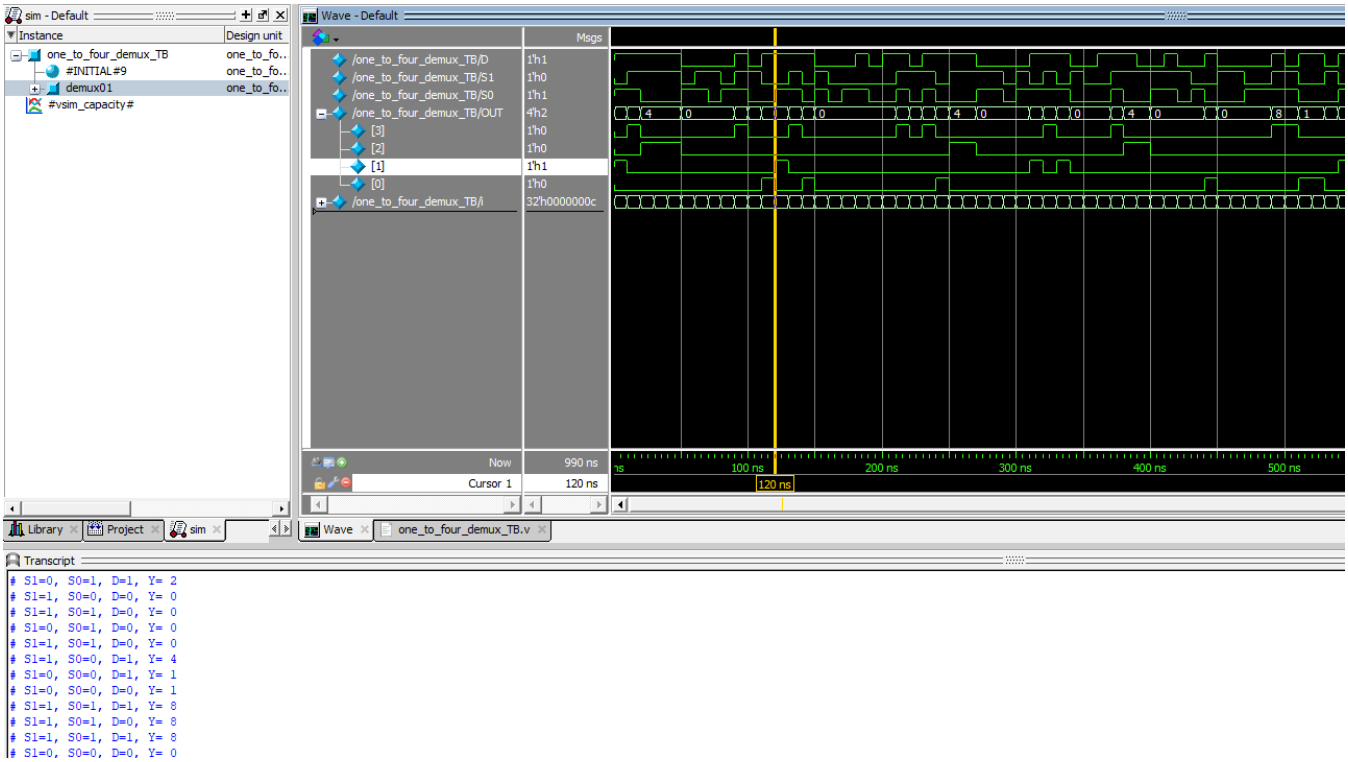
```
module one_to_four_demux (D, S0, S1, Y);

    input D, S1, S0;
    output reg [3:0]Y;

    always@(S1,S0) begin
        case ({S1,S0})
        2'b00: begin
               Y[3]=0;
               Y[2]=0;
               Y[1]=0;
               Y[0]=D;
               end
        2'b01: begin
               Y[3]=0;
               Y[2]=0;
               Y[1]=D;
               Y[0]=0;
               end
        2'b10: begin
               Y[3]=0;
               Y[2]=D;
               Y[1]=0;
               Y[0]=0;
               end
        2'b11: begin
               Y[3]=D;
               Y[2]=0;
               Y[1]=0;
               Y[0]=0;
               end
        default: Y = 0;
        endcase
    end

endmodule
```

Test Bench:

one_to_four_demux_TB.v

```
module one_to_four_demux_TB ();

    reg D, S1, S0;
    wire [3:0] OUT;
    one_to_four_demux demux01 (D, S0, S1, OUT);

    integer i;

    initial begin
        for(i=0;i<99;i=i+1) begin
            S1 = $random ;
            S0 = $random ;
            D  = $random ;
            #10;
        end
        $stop;
    end

    initial begin
        $monitor ("S1=%d, S0=%d, D=%d, Y=%d",S1,S0,D,OUT);
    end
endmodule
```

Simulation:


sim - Default
Instance
Design unit
one_to_four_demux_TB
one_to_fo..
#INITIAL#9
demux01
#vsim_capacity#
Wave - Default
Msgs
/one_to_four_demux_TB/D
/one_to_four_demux_TB/S1
/one_to_four_demux_TB/S0
/one_to_four_demux_TB/OUT
[3]
[2]
[1]
[0]
/one_to_four_demux_TB/i
1'h1
1'h0
1'h1
4'h2
1'h0
1'h0
1'h1
1'h0
32'h0000000c
Now
990 ns
Cursor 1
120 ns
100 ns
200 ns
300 ns
400 ns
500 ns
Library
Project
sim
Wave
one_to_four_demux_TB.v
Transcript
S1=0, S0=1, D=1, Y= 2
S1=1, S0=0, D=0, Y= 0
S1=1, S0=1, D=0, Y= 0
S1=0, S0=1, D=0, Y= 0
S1=1, S0=1, D=0, Y= 0
S1=1, S0=0, D=1, Y= 4
S1=0, S0=0, D=1, Y= 1
S1=0, S0=0, D=0, Y= 1
S1=1, S0=1, D=1, Y= 8
S1=1, S0=1, D=0, Y= 8
S1=1, S0=1, D=1, Y= 8
S1=0, S0=0, D=0, Y= 0