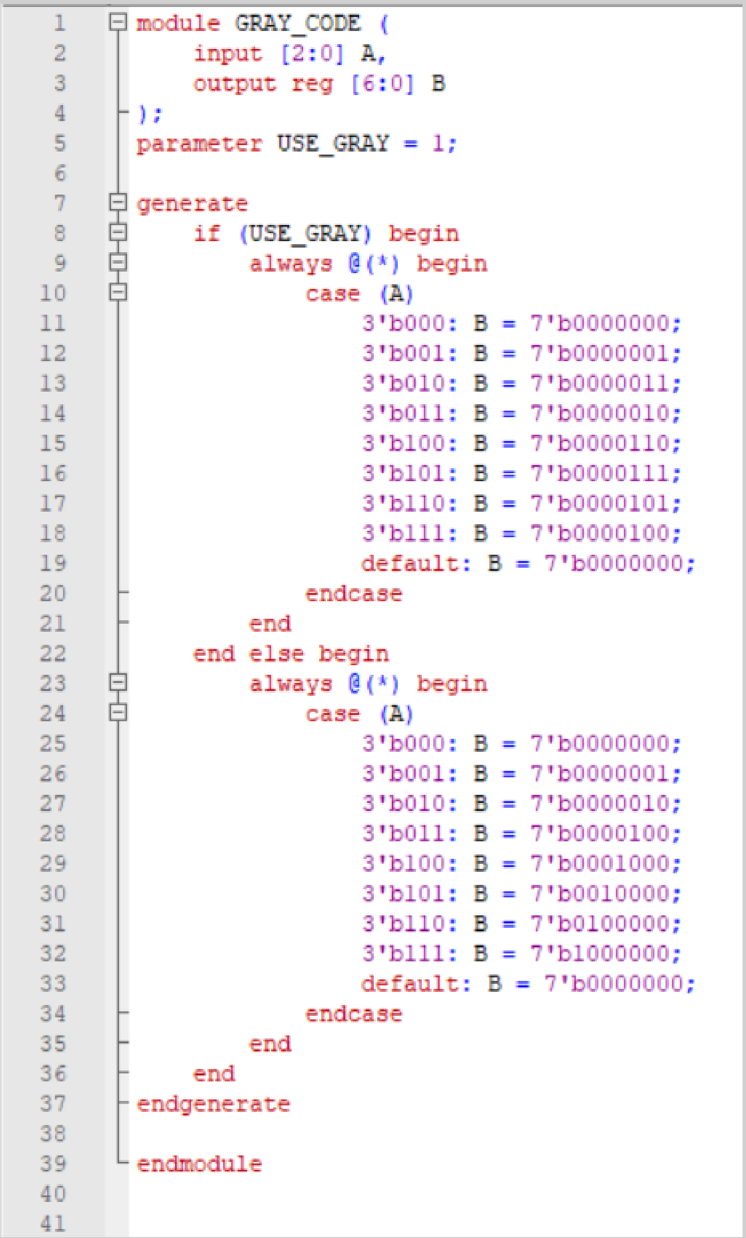
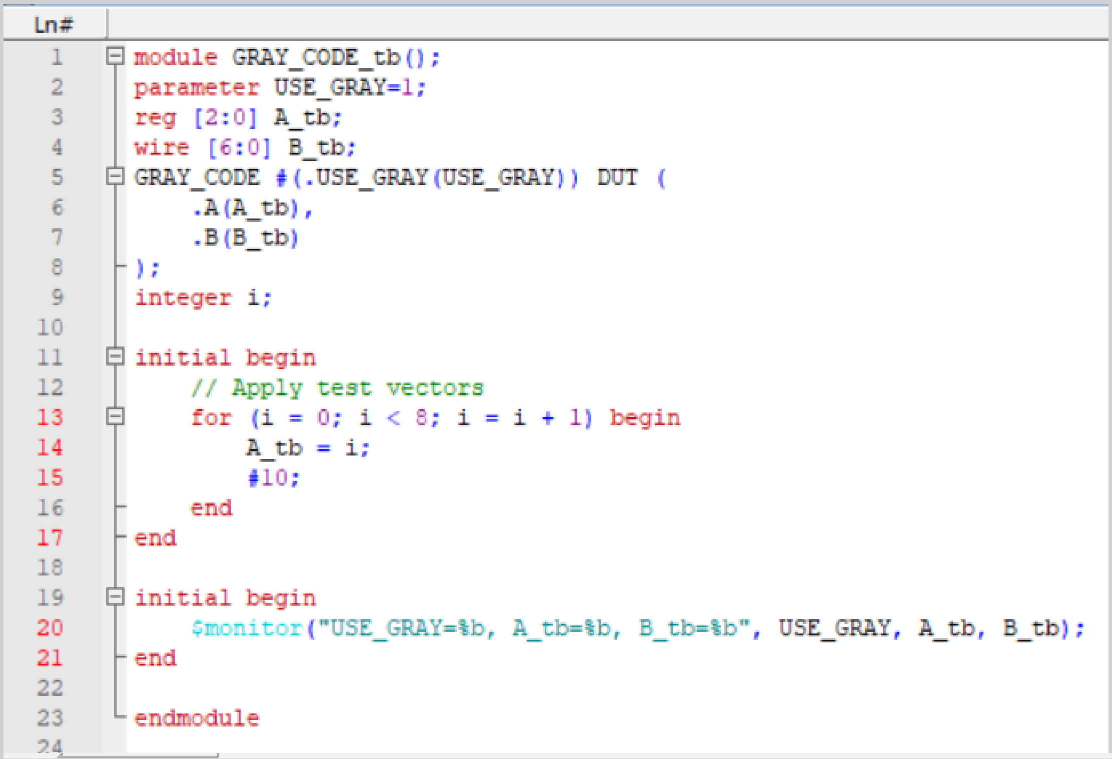
ASSINMENT 2_EXTENDED

GROUP	NAME
1	Ahmed Khalaf Mohamed Ali

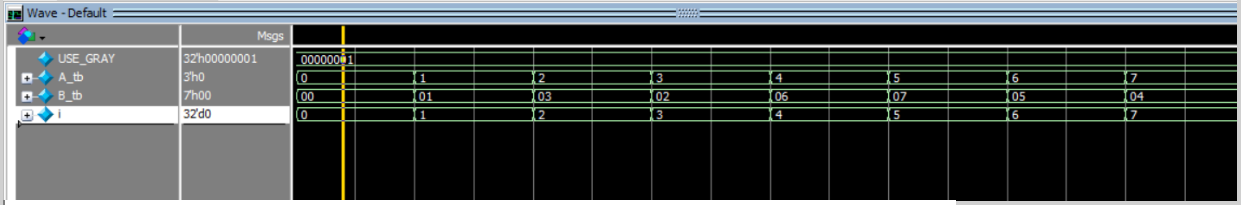
1)

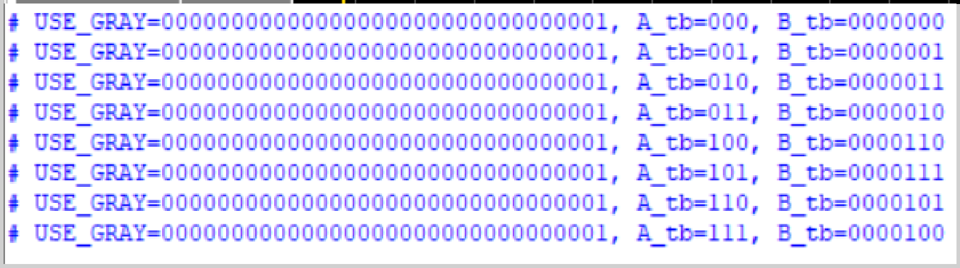


```
module GRAY_CODE (
    input [2:0] A,
    output reg [6:0] B
);
parameter USE_GRAY = 1;

generate
    if (USE_GRAY) begin
        always @(*) begin
            case (A)
                3'b000: B = 7'b0000000;
                3'b001: B = 7'b0000001;
                3'b010: B = 7'b0000011;
                3'b011: B = 7'b0000010;
                3'b100: B = 7'b0000110;
                3'b101: B = 7'b0000111;
                3'b110: B = 7'b0000101;
                3'b111: B = 7'b0000100;
                default: B = 7'b0000000;
            endcase
        end
    end else begin
        always @(*) begin
            case (A)
                3'b000: B = 7'b0000000;
                3'b001: B = 7'b0000001;
                3'b010: B = 7'b0000010;
                3'b011: B = 7'b0000100;
                3'b100: B = 7'b0001000;
                3'b101: B = 7'b0010000;
                3'b110: B = 7'b0100000;
                3'b111: B = 7'b1000000;
                default: B = 7'b0000000;
            endcase
        end
    end
endgenerate

endmodule
```

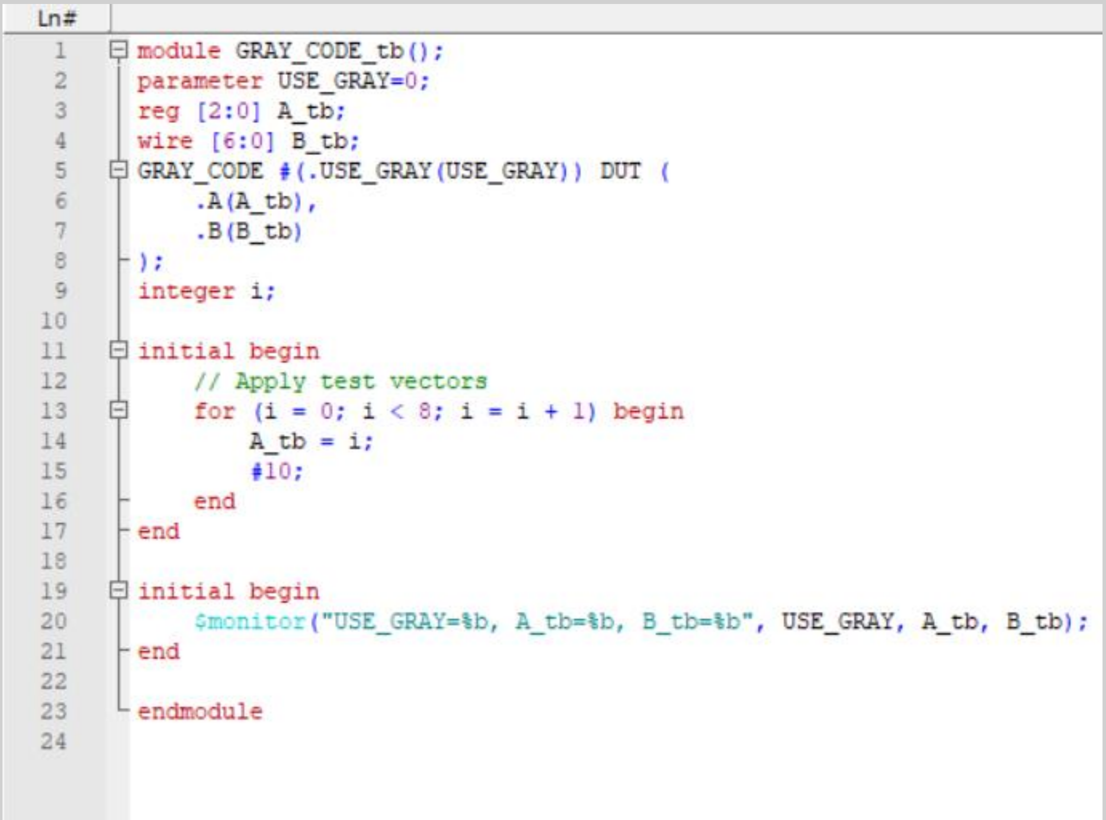


```
module GRAY_CODE_tb();
parameter USE_GRAY=1;
reg [2:0] A_tb;
wire [6:0] B_tb;
GRAY_CODE #(.USE_GRAY(USE_GRAY)) DUT (
    .A(A_tb),
    .B(B_tb)
);
integer i;

initial begin
    // Apply test vectors
    for (i = 0; i < 8; i = i + 1) begin
        A_tb = i;
        #10;
    end
end

initial begin
    $monitor("USE_GRAY=%b, A_tb=%b, B_tb=%b", USE_GRAY, A_tb, B_tb);
end

endmodule
```

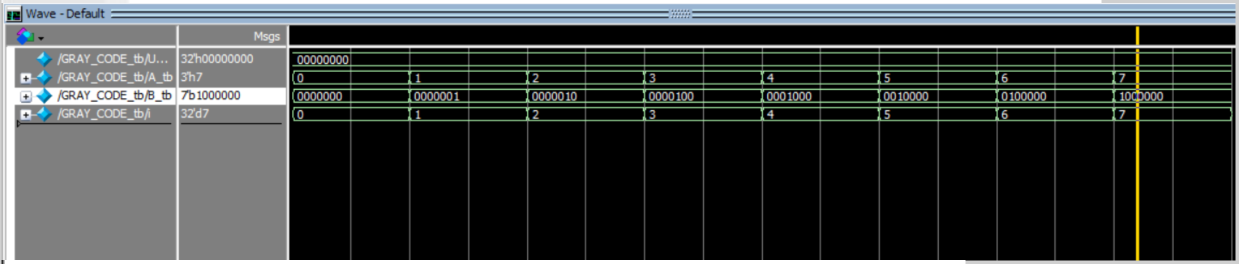


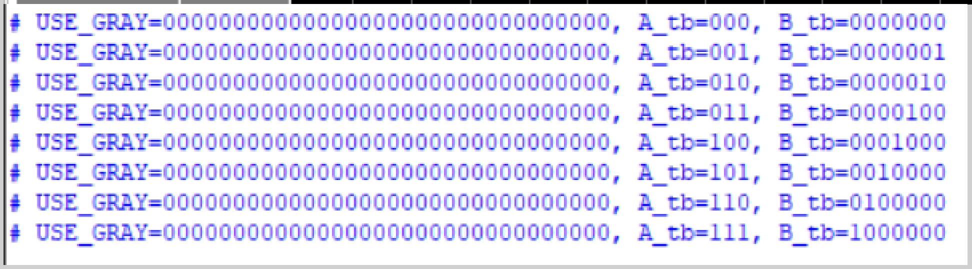


```
# USE_GRAY=00000000000000000000000000000001, A_tb=000, B_tb=0000000
# USE_GRAY=00000000000000000000000000000001, A_tb=001, B_tb=0000001
# USE_GRAY=00000000000000000000000000000001, A_tb=010, B_tb=0000011
# USE_GRAY=00000000000000000000000000000001, A_tb=011, B_tb=0000010
# USE_GRAY=00000000000000000000000000000001, A_tb=100, B_tb=0000110
# USE_GRAY=00000000000000000000000000000001, A_tb=101, B_tb=0000111
# USE_GRAY=00000000000000000000000000000001, A_tb=110, B_tb=0000101
# USE_GRAY=00000000000000000000000000000001, A_tb=111, B_tb=0000100
```



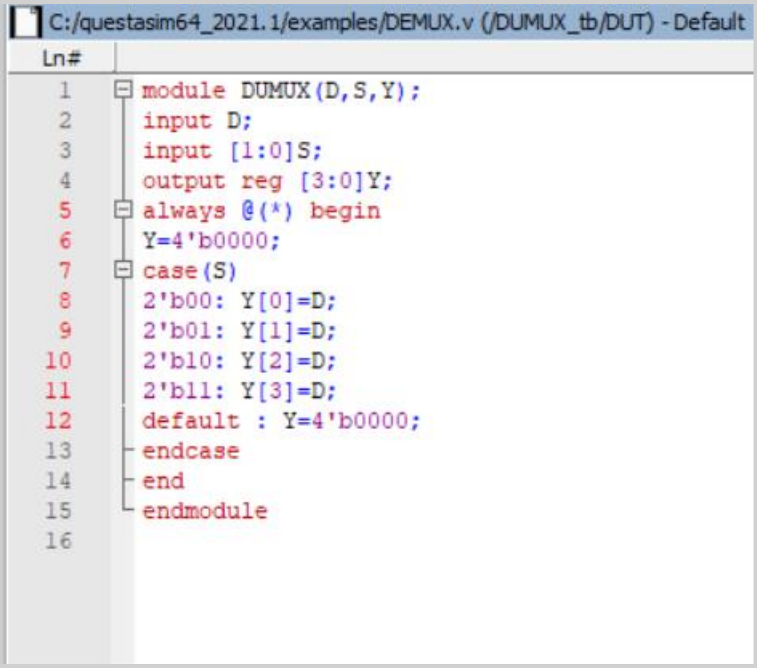
```
module GRAY_CODE_tb();
parameter USE_GRAY=0;
reg [2:0] A_tb;
wire [6:0] B_tb;
GRAY_CODE #(.USE_GRAY(USE_GRAY)) DUT (
    .A(A_tb),
    .B(B_tb)
);
integer i;

initial begin
    // Apply test vectors
    for (i = 0; i < 8; i = i + 1) begin
        A_tb = i;
        #10;
    end
end

initial begin
    $monitor("USE_GRAY=%b, A_tb=%b, B_tb=%b", USE_GRAY, A_tb, B_tb);
end

endmodule
```



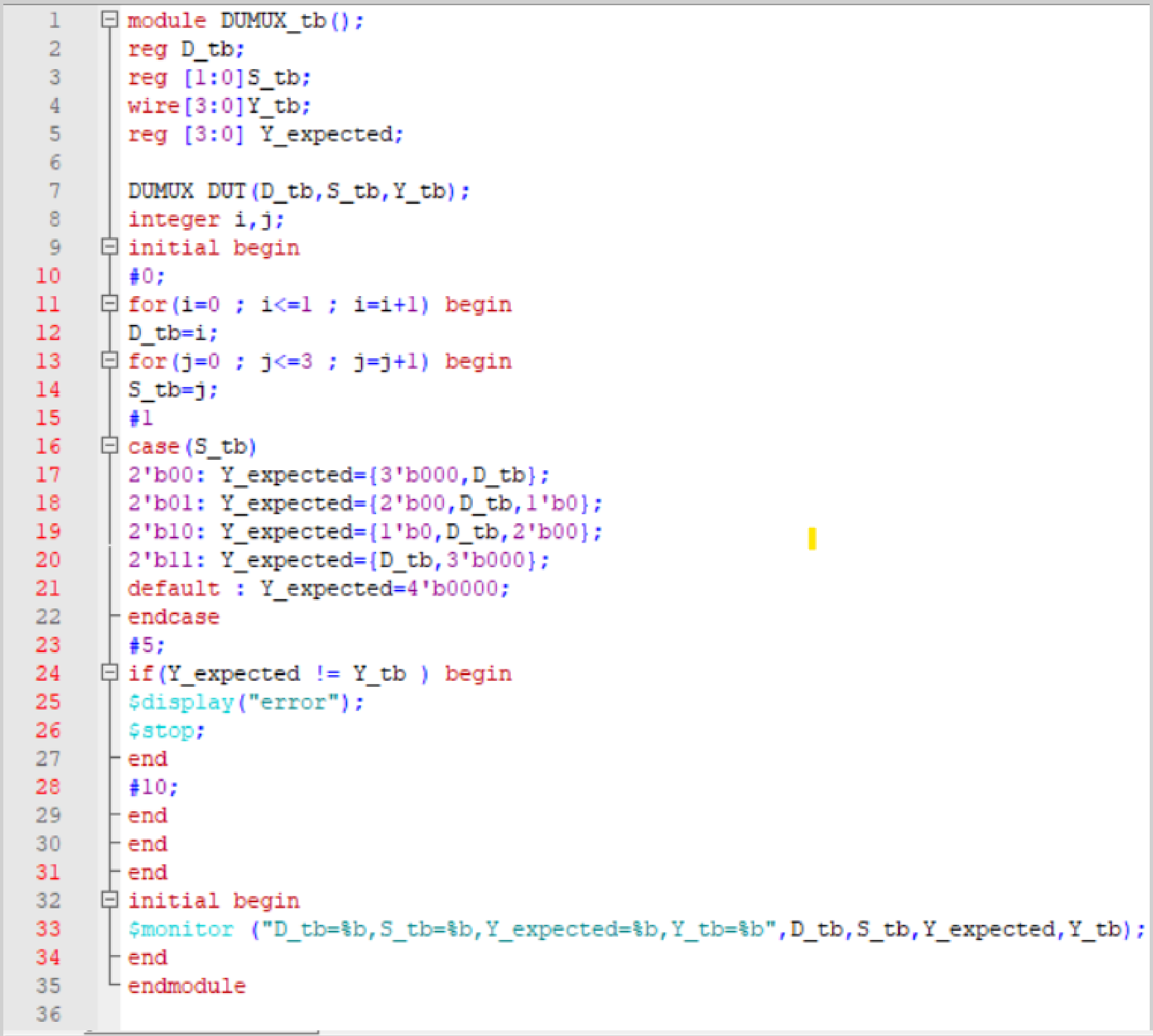


```
# USE_GRAY=00000000000000000000000000000000, A_tb=000, B_tb=0000000
# USE_GRAY=00000000000000000000000000000000, A_tb=001, B_tb=0000001
# USE_GRAY=00000000000000000000000000000000, A_tb=010, B_tb=0000010
# USE_GRAY=00000000000000000000000000000000, A_tb=011, B_tb=0000100
# USE_GRAY=00000000000000000000000000000000, A_tb=100, B_tb=0001000
# USE_GRAY=00000000000000000000000000000000, A_tb=101, B_tb=0010000
# USE_GRAY=00000000000000000000000000000000, A_tb=110, B_tb=0100000
# USE_GRAY=00000000000000000000000000000000, A_tb=111, B_tb=1000000
```

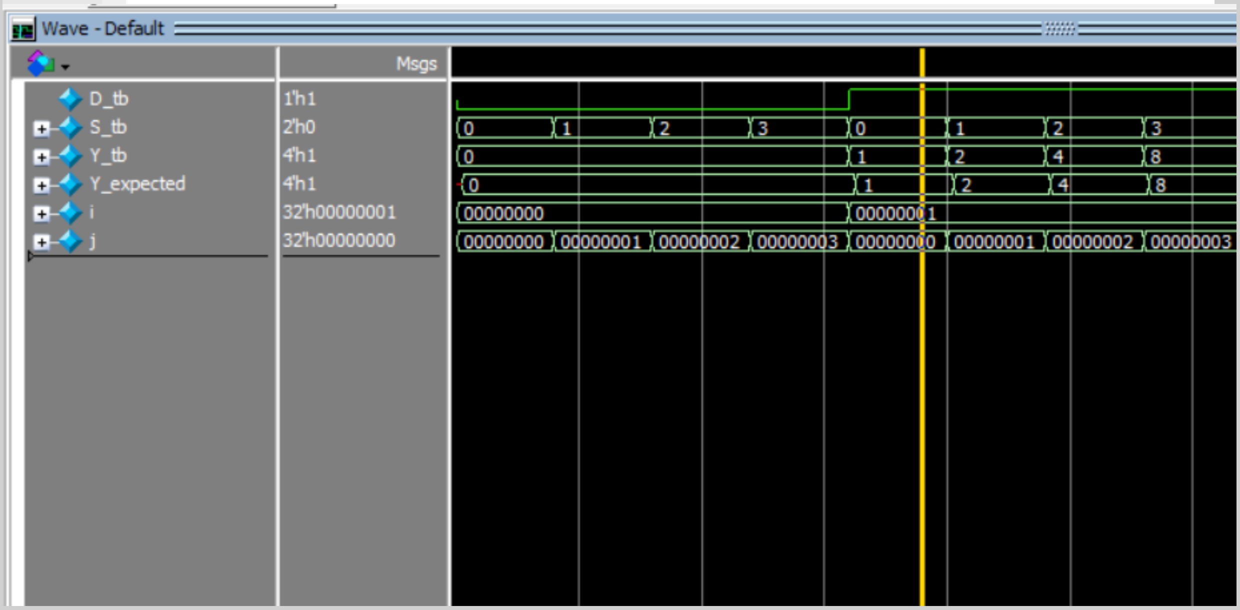
2)

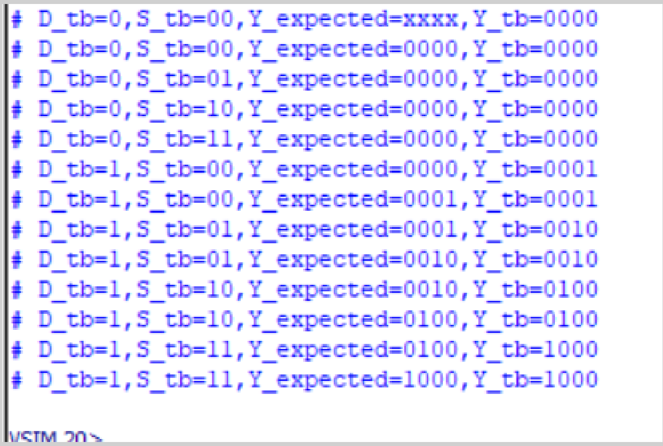

C:/questasim64_2021.1/examples/DEMUX.v (/DUMUX_tb/DUT) - Default
Ln#

```
module DUMUX(D,S,Y);
input D;
input [1:0]S;
output reg [3:0]Y;
always @(*) begin
Y=4'b0000;
case(S)
2'b00: Y[0]=D;
2'b01: Y[1]=D;
2'b10: Y[2]=D;
2'b11: Y[3]=D;
default : Y=4'b0000;
endcase
end
endmodule
```



```
module DUMUX_tb();
reg D_tb;
reg [1:0]S_tb;
wire[3:0]Y_tb;
reg [3:0] Y_expected;

DUMUX DUT(D_tb,S_tb,Y_tb);
integer i,j;
initial begin
#0;
for(i=0 ; i<=1 ; i=i+1) begin
D_tb=i;
for(j=0 ; j<=3 ; j=j+1) begin
S_tb=j;
#1
case(S_tb)
2'b00: Y_expected={3'b000,D_tb};
2'b01: Y_expected={2'b00,D_tb,1'b0};
2'b10: Y_expected={1'b0,D_tb,2'b00};
2'b11: Y_expected={D_tb,3'b000};
default : Y_expected=4'b0000;
endcase
#5;
if(Y_expected != Y_tb ) begin
$display("error");
$stop;
end
#10;
end
end
end
initial begin
$monitor ("D_tb=%b,S_tb=%b,Y_expected=%b,Y_tb=%b",D_tb,S_tb,Y_expected,Y_tb);
end
endmodule
```





```
# D_tb=0,S_tb=00,Y_expected=xxxx,Y_tb=0000
# D_tb=0,S_tb=00,Y_expected=0000,Y_tb=0000
# D_tb=0,S_tb=01,Y_expected=0000,Y_tb=0000
# D_tb=0,S_tb=10,Y_expected=0000,Y_tb=0000
# D_tb=0,S_tb=11,Y_expected=0000,Y_tb=0000
# D_tb=1,S_tb=00,Y_expected=0000,Y_tb=0001
# D_tb=1,S_tb=00,Y_expected=0001,Y_tb=0001
# D_tb=1,S_tb=01,Y_expected=0001,Y_tb=0010
# D_tb=1,S_tb=01,Y_expected=0010,Y_tb=0010
# D_tb=1,S_tb=10,Y_expected=0010,Y_tb=0100
# D_tb=1,S_tb=10,Y_expected=0100,Y_tb=0100
# D_tb=1,S_tb=11,Y_expected=0100,Y_tb=1000
# D_tb=1,S_tb=11,Y_expected=1000,Y_tb=1000

VSIM 20>
```