**Course Code: EE461 Lab**

**LAB REPORT -3**

**PREPARED BY**

**Khandoker Samiul Hoque**

**Student Id: 19837**

**(a)**

**Question No Answer: 01**

wire C, D, E;

always@(A or B)

begin

if (A)

C = B ^ A;

else C = D & E;

F = C | A;

end

Here is a syntax error in this program. , begin should be in the next line and have to declare the connection C, D and E.

**Question No Answer: 02**

reg C;

always@(B)

begin

C | = B;

end

always@(E)

begin

C \*= E;end

Syntax should be like this C | = B; and also C \*= E; in this way cause in this order assignment operator was not valid.

**Question No Answer: 03**

always@(posedge clock)

begin

if(A) Q <= D;

end

always@(Q or E)

begin

case (Q)

0: F = E;

default: F = 1;

endcase

end

clock : 00001111

D : xxxxxxxx

E : 11110000

Q : xxxx0000

F : 11110000

According to statement when Q = 0 then F = E otherwise F=1. So F is totally rely on the Q and its assigned according to case assignment. F changes whenever Q or E changes.

**Question No Answer: 04**

module top;

wire A, B, C;

bar u1 (A,B);

bar u2 (C,B);

endmodule

module bar (input D; output wire E);

assign E= ~D;

endmodule

Only B was defined. It should be A B and C as defined.

**Question No Answer: 05**

module foo(input wire A, input wire B; output reg E);

wire C, D;

always@(posedge clock)

begin

E = B & D;

end

initial

begin

assign C = A ^ D;

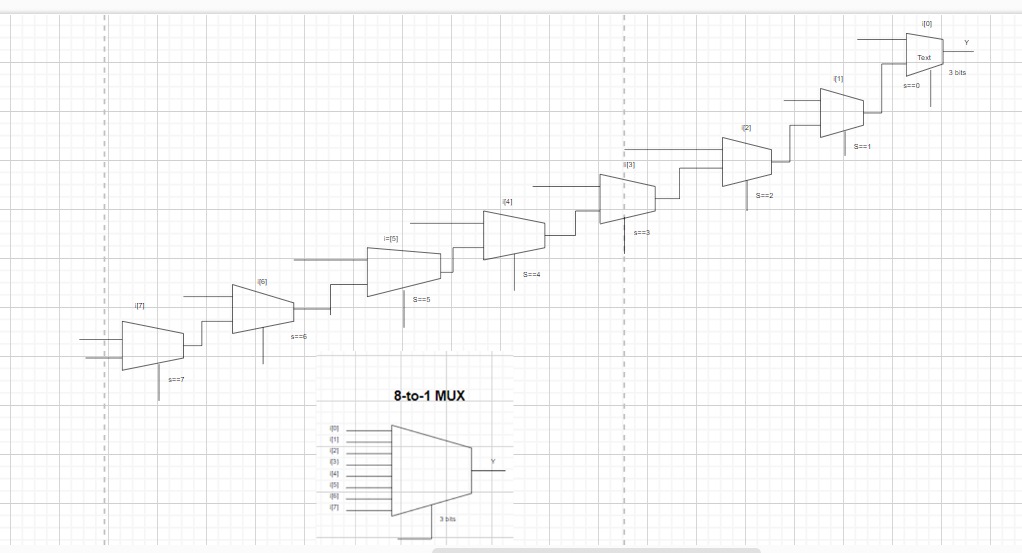
assign D = C | B;

end

endmodule

assign condition will be under the initial block. begin and end should be write on each part . Input A and B must be declared as input wire “x” like that.

(b)



The CaseMux8 has more input options than the IfMux8, but requires a larger control signal, whereas the IfMux8 has fewer input options but a smaller control signal.

The appropriate multiplexer to use is determined by the application's specific requirements.