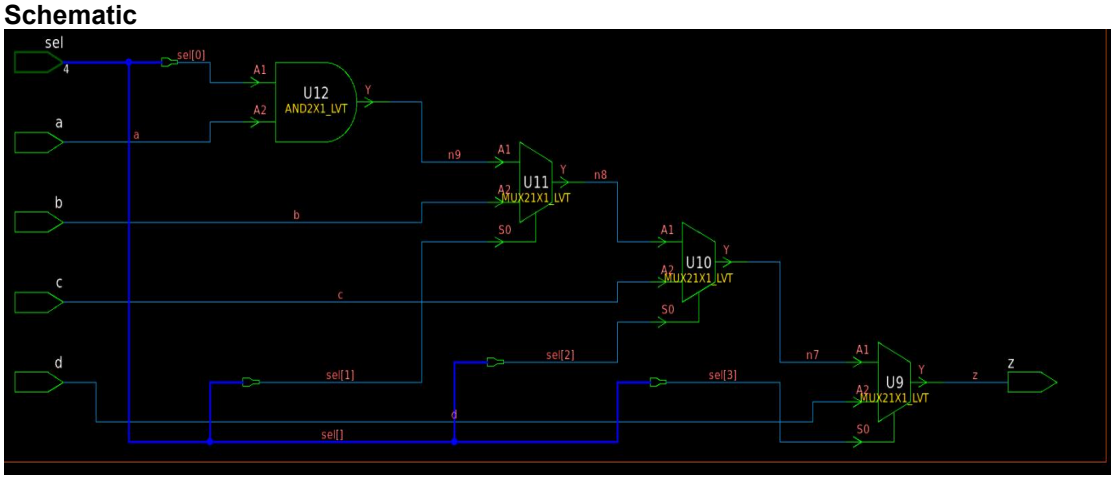
Lab2)

Design:

```
module mult_if(a, b, c, d, sel, z);
input a, b, c, d;
input [3:0] sel;
output z;
reg z;
always @(a or b or c or d or sel)
begin
z = 0;
if (sel[0]) z = a;
if (sel[1]) z = b;
if (sel[2]) z = c;
if (sel[3]) z = d;
end
endmodule
```

Schematic



In the synthesized design of lab2, the inputs to the first SELECT_OP in the chain
(sel[0] and a or 0) have the longest delay to the output z. The inputs to the last SELECT_OP in the chain (sel[3] and d) have the shortest delay to the output.

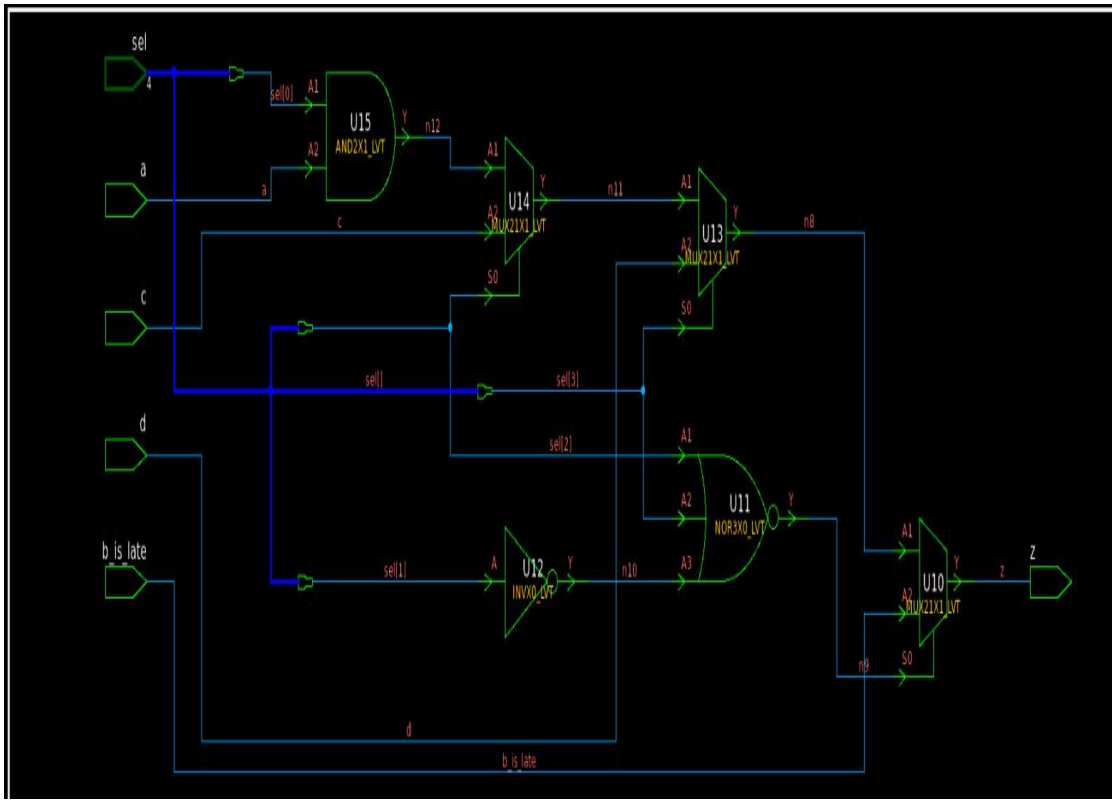
Lab3)

Design

```
module mult_if_improved(a, b_is_late, c, d, sel, z);
input a, b_is_late, c, d;
input [3:0] sel;
output z;
reg z, z1;
always @(a or b_is_late or c or d or sel)
begin
z1 = 0;
if (sel[0])
z1 = a;
if (sel[2])
z1 = c;
if (sel[3])
z1 = d;
```

```
if (sel[1] & ~(sel[2]|sel[3]))
z = b_is_late;
else
z = z1;
end
Endmodule
```

Schematic

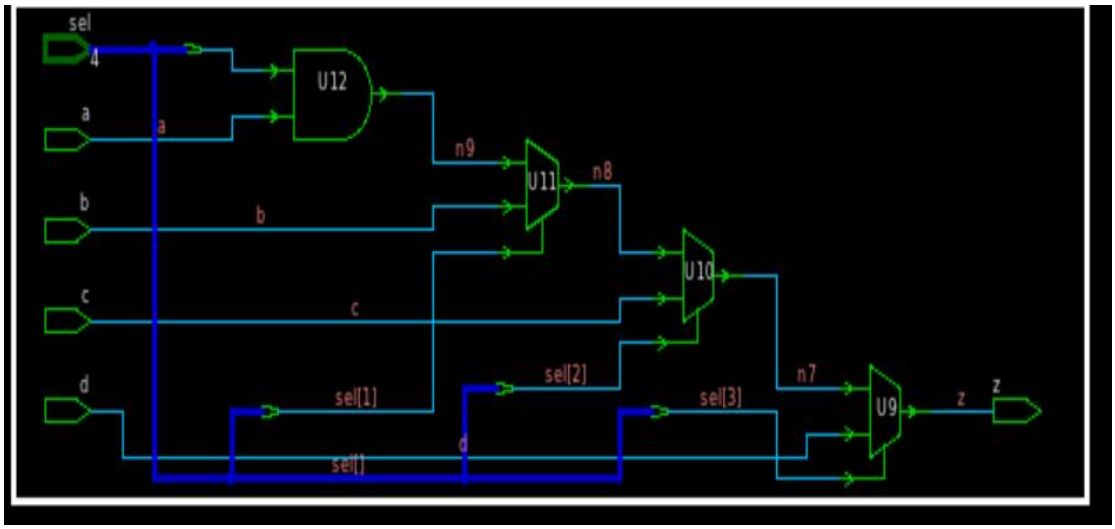


Lab4)
Design:

```
module single_if(a, b, c, d, sel, z);
input a, b, c, d;
input [3:0] sel;
output z;
reg z;
always @(a or b or c or d or sel)
begin
z = 0;
if (sel[3])
z = d;
else if (sel[2])
z = c;
else if (sel[1])
z = b;
else if(sel[0])
z = a;
```

end
endmodule

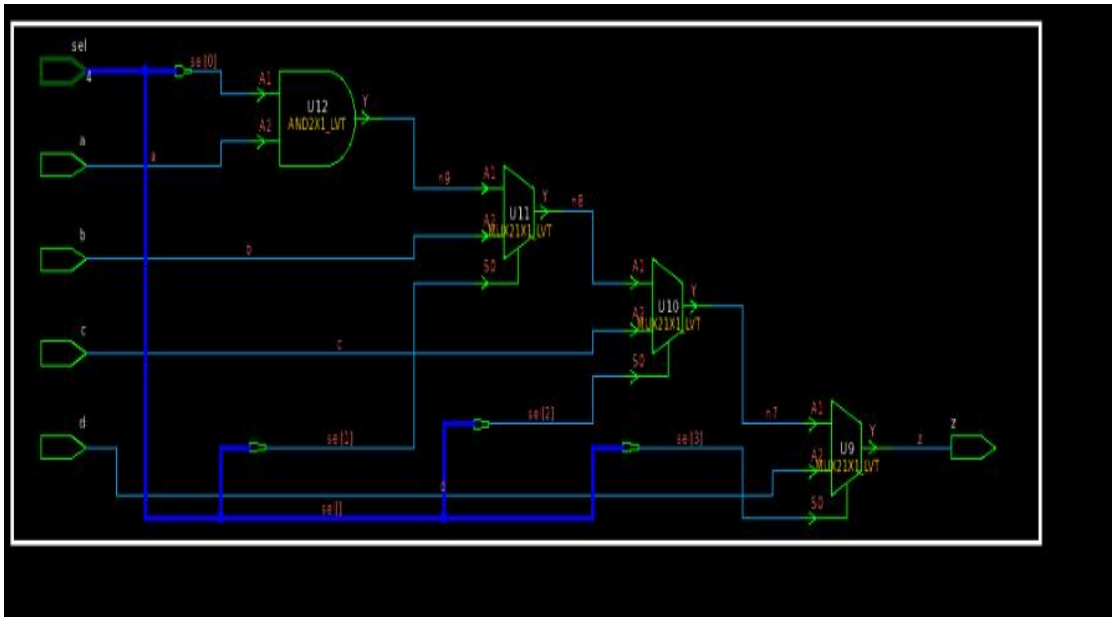
schematic:



Lab 5)
Design:

```
module case1(a, b, c, d, sel, z);
input a, b, c, d;
input [3:0] sel;
output z;
reg z;
always @(a or b or c or d or sel)
begin
casex (sel)
4'b1xxx: z = d;
4'bx1xx: z = c;
4'bxx1x: z = b;
4'bxxx1: z = a;
default: z = 1'b0;
endcase
end
endmodule
```

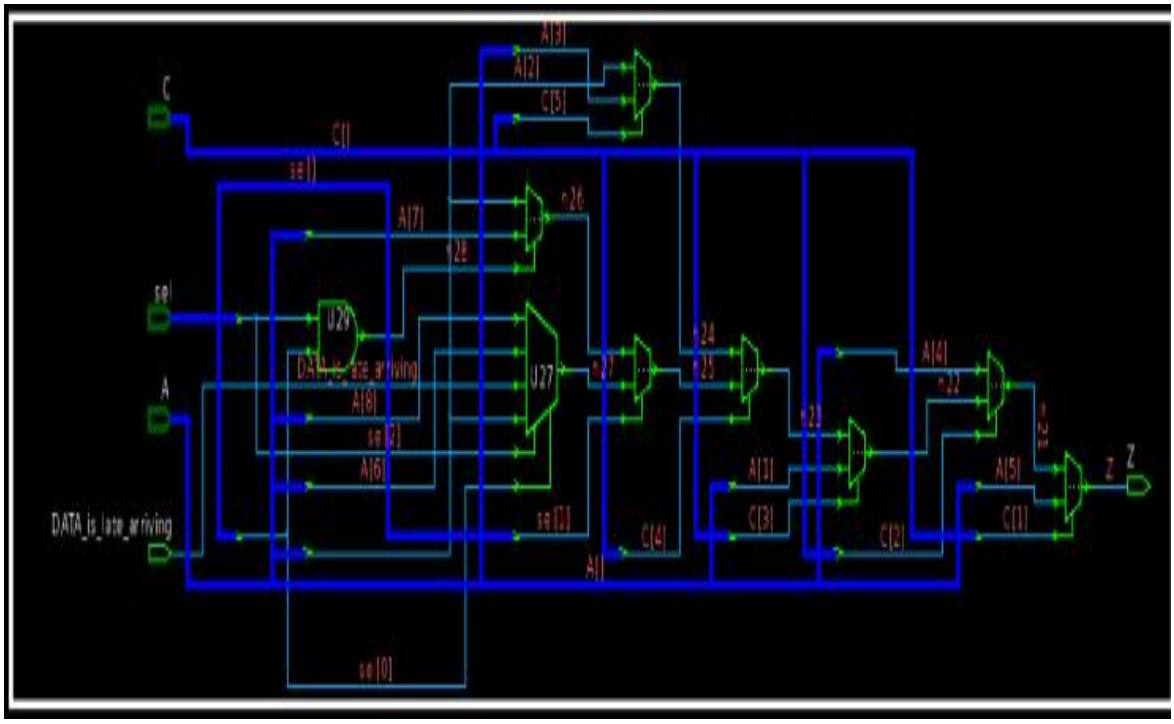
Schematic:



Lab 6)
Design:

```
module case_in_if_01(A, DATA_is_late_arriving, C, sel, Z);
input [8:1] A;
input DATA_is_late_arriving;
input [2:0] sel;
input [5:1] C;
output Z;
reg Z;
always @ (sel or C or A or DATA_is_late_arriving)
begin
if (C[1])
Z = A[5];
else if (C[2] == 1’b0)
Z = A[4];
else if (C[3])
Z = A[1];
else if (C[4])
case (sel)
3’b010: Z = A[8];
3’b011: Z = DATA_is_late_arriving;
3’b101: Z = A[7];
3’b110: Z = A[6];
default: Z = A[2];
endcase
else if (C[5] == 1’b0)
Z = A[2];
else
Z = A[3];
end
endmodule
```

Schematic


C
A
sel
DATA_is_late_arriving
A[3]
A[2]
C[3]
C[1]
sel[1]
A[7]
n26
n28
U29
U27
A[8]
sel[2]
A[6]
sel[3]
n24
n25
n27
A[4]
n22
n21
A[3]
C[3]
C[4]
A[1]
C[2]
A[5]
C[1]
n23
Z
sel[0]