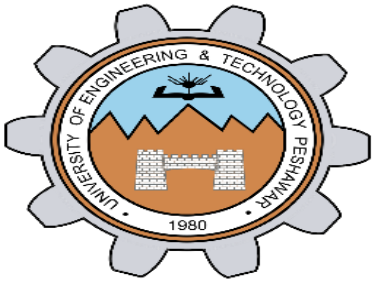
Lab report no 12



Fall 2021

Computer Architecture and organization Lab

Submitted By

Names	Registration No
Muhammad Ali	**19pwcse1801**

Section: **A**

Date:24,2,21

Submitted to: Dr Amaad khalil

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Task no 1 (2x1mux): -

Code: -

Mux2x1assign module: -

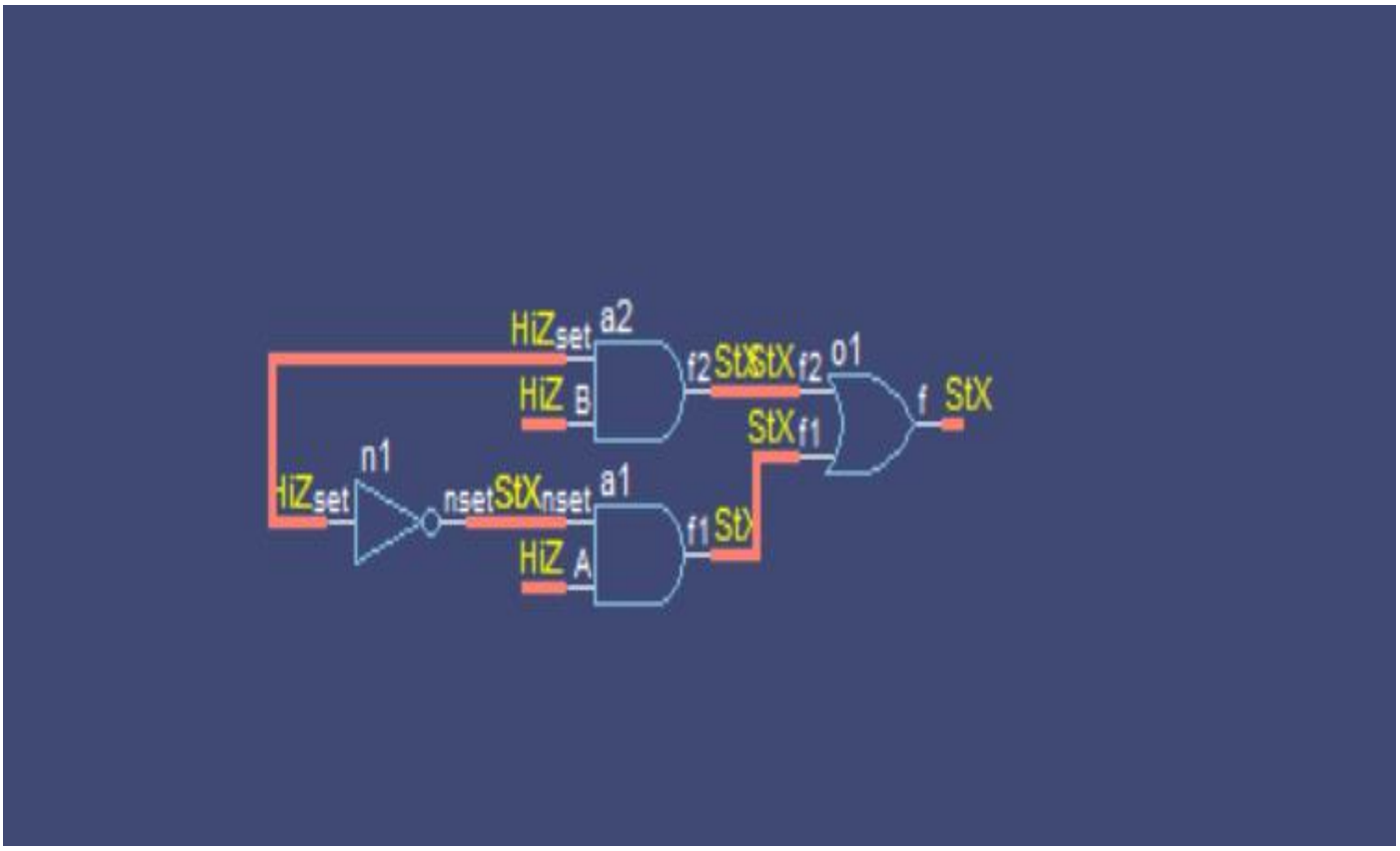
```
module mux2x1assign(A,B,set,f);
input A,B,set;
output f;
assign f = set? B:A;
endmodule
```

Stim module: -

```
module stim6();
reg A,B,set;
wire f;
mux2x1assign trr(A,B,set,f);
initial
begin
$display ("A B set f");

A=0;B=0; set=0;
#10 $display("%b %b %b %b",A,B,set,f);

A=0;B=0; set=1;
#10 $display("%b %b %b %b",A,B,set,f);
A=0;B=1; set=0;
#10 $display("%b %b %b %b",A,B,set,f);
A=0;B=1; set=1;
#10 $display("%b %b %b %b",A,B,set,f);
```

```
A=1;B=0; set=0;
#10 $display("%b %b %b %b",A,B,set,f);
A=0;B=1; set=1;
#10 $display("%b %b %b %b",A,B,set,f);
A=0;B=0; set=1;
#10 $display("%b %b %b %b",A,B,set,f);
A=1;B=1; set=1;
#10 $display("%b %b %b %b",A,B,set,f);
end
endmodule
```

Design Circuit: -



Task no 2 (4x1mux): -

Code: -

Mux4x1assign module: -

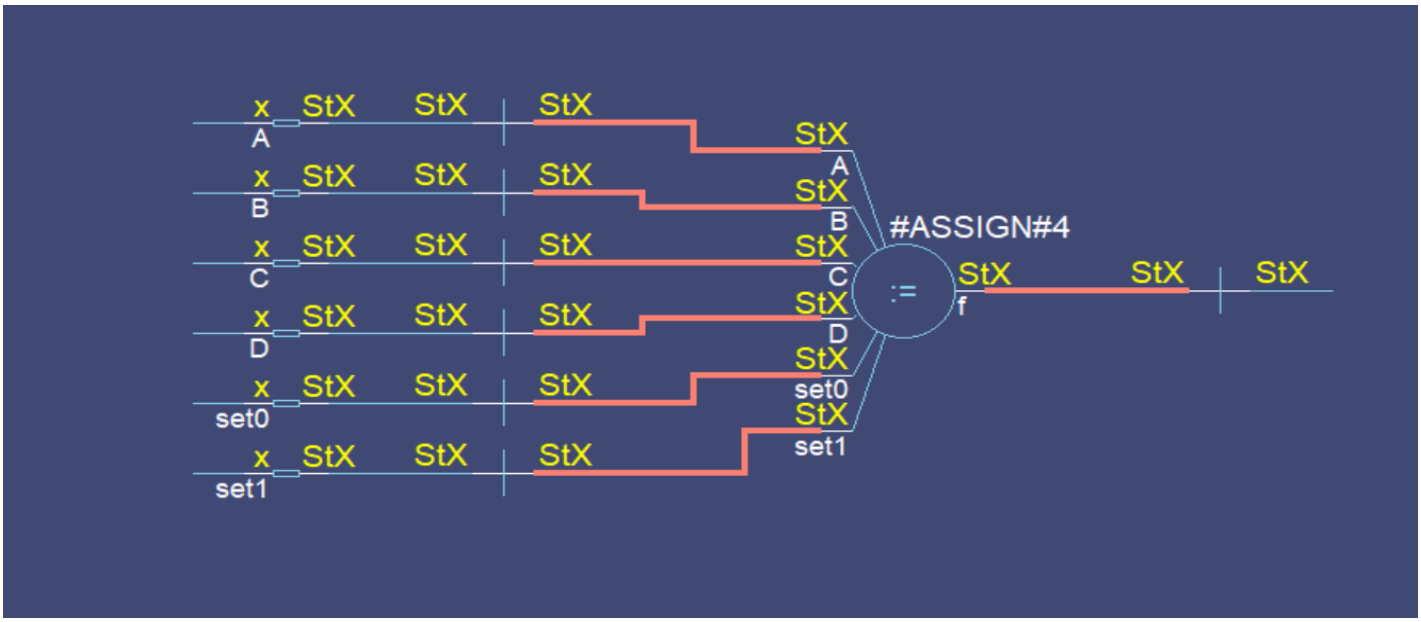
```
module mux4x1assign(A,B,C,D,set0,set1,f);
input A,B,C,D,set0,set1;
output f;
assign f = set1?(set0?D:C):(set0? B:A);
endmodule
```

Stim module (using monitor): -

```
module stim7();
reg A,B,C,D,set0,set1;
wire f;
mux4x1assign tr(A,B,C,D,set0,set1,f);

initial
begin

$display ("A B C D Set0 set1");

#10 A=0;B=0; C=0; D=0; set0=0; set1=0;
#10 A=0;B=0; C=0; D=0; set0=0; set1=1;
#10 A=0;B=0; C=0; D=0; set0=1; set1=0;
#10 A=0;B=0; C=0; D=1; set0=1; set1=0;
#10 A=0;B=0; C=1; D=0; set0=1; set1=0;
#10 A=0;B=1; C=0; D=0; set0=1; set1=0;
#10 A=1;B=0; C=0; D=0; set0=1; set1=0;
```

```
#10 A=0;B=0; C=0; D=0; set0=0; set1=0;
end
initial
begin
#2 $monitor("A=%b B=%b C=%b D=%b set0=%b set1=%b f=%b",A,B,C,D,set0,set1,f);
end
endmodule
```

Design Circuit: -



Task no 3 (Half adder with dataflow method): -

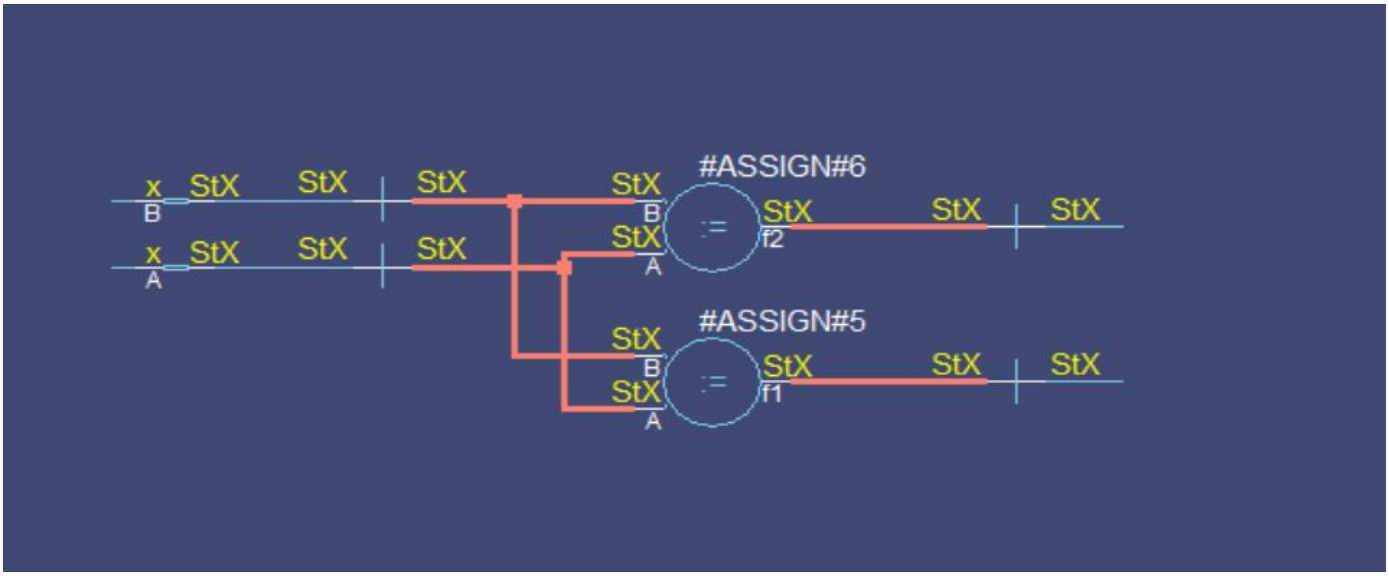
Code: -

```
module halfadderassign(A,B,f1,f2);
input A,B;
output f1,f2;

assign f1= A & B;
assign f2= A & ~B | ~A & B;
endmodule
```

Stim Half adder module: -

```
module stim9addasign();
reg A,B;
wire f1,f2;
halfadderassign t(A,B,f1,f2);
initial
begin

$display ("A B f1 f2");

A=0;B=0;
#10 $display("%b %b %b %b",A,B,f1,f2);

A=0;B=1;
#10 $display("%b %b %b %b",A,B,f1,f2);

A=1;B=0;
#10 $display("%b %b %b %b",A,B,f1,f2);

A=1;B=1;
#10 $display("%b %b %b %b",A,B,f1,f2);
end
endmodule
```

Design Circuit: -



Task no 4 (2x4 decoder with dataflow method): -

Code: -

```
module decoder2x4(A,B,F1,F2,F3,F4);
input A,B;
output F1,F2,F3,F4;

assign F1 = ~A & ~B;
assign F2 = ~A &  B;
assign F3 =  A & ~B;
assign F4 =  A &  B;
endmodule
```

Stim 2x4 decoder module: -

```
module stimdecoderwithovrflow();
reg A,B;
wire F1,F2,F3,F4;
```

```
decoder2x4 tr(A,B,F1,F2,F3,F4);

initial

begin


$display ("A B F1 F2 F3 F4");

A=0; B=0;

#10 $display("%b %b %b %b %b %b",A,B,F1,F2,F3,F4);

A=0; B=1;

#10 $display("%b %b %b %b %b %b", A,B,F1,F2,F3,F4);

A=1; B=0;

#10 $display("%b %b %b %b %b %b", A,B,F1,F2,F3,F4);

A=1; B=1;

#10 $display("%b %b %b %b %b %b", A,B,F1,F2,F3,F4);

end

endmodule
```

Design Circuit: -