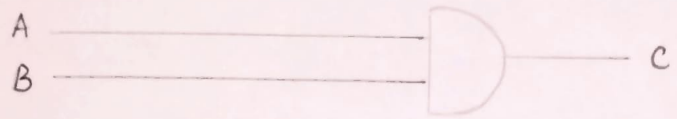
Name - MD Quzal Jawed
Department - CSE (Second Year)
Semester - 4th
Roll No. - 37

Sub - Computer Architecture Lab
Date - 03.03.23

AND GATE :-



$C = A \cdot B$

Truth Table :

Input		Output
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

- When alder Riviera is selected time slow not required in Edu.
- Write in default test bench & design; Can't create.

Code i) Design Module :-

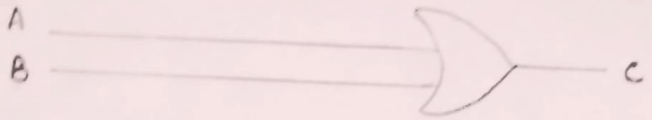
```
module and 2
(input A;
 input B;
 Output C);
assign C = A & B;
end module
```

ii) Test Bench :-

```
timescale 1ns/10ps
module and 2_tb;
reg A, B;
wire C;
and2 wt(.A(A), .B(B), .c(c));
initial
  begin
  A = 0;
  B = 0
  #10;
  A = 1;
  B = 0;
  # 10;
  A = 1;
  B = 1;
  #10;
end
end module
```

Name	Value	0ns	500
A	1		
B	1		
C	1		

OR GATE :-



$C = A + B$

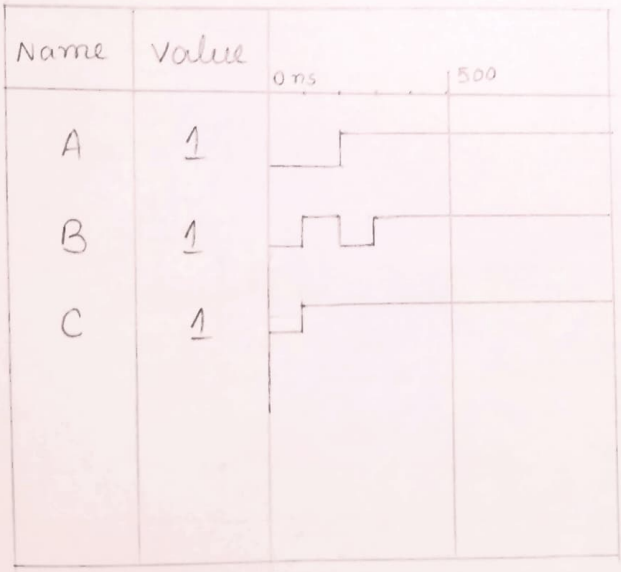
Truth Table :-

Input		Output
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

Code

i) Design Module :

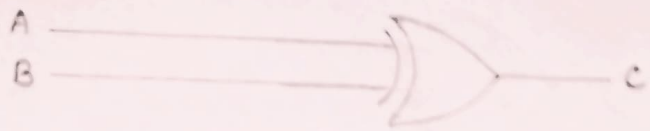
```
module Or2
(input A,
 input B,
 Output C
);
assign C = A|B;
end module
```



ii) Test Bench :

```
timescale 0r2-tb;
reg A, B;
wire C;
Or2 out(.A(A), .B(B), .C(C));
initial
  begin
  A=0;
  B=0;
  # 10;
  A=0;
  B=1;
  #10;
  A=1;
  B=0;
  # 10;
  A=1;
  B=1;
  # 10;
  end
end module
```

XOR GATE :-



$C = A \oplus B$

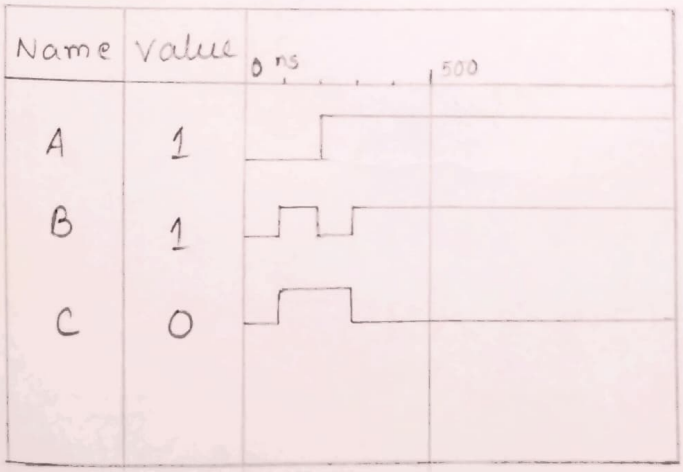
Truth Table :-

Input		Output
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

Code :-

i) Design Module :-

```
module xor2
(input A,
input B,
Output C
);
assign C = A^B;
end module
```

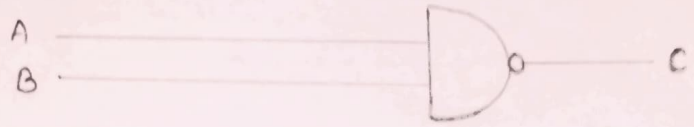


ii) Test Bench :-

```
timescale 1ns/10ps
module xor2_tb;
reg A,B;
wire C;
xor2 uut (.A(A),.B(B),.C(C));
initial
  begin
    A=0;
    B=0;
    #10;
    A=0;
    B=1;
    #10;
    A=1;
    B=0;
    #10;
    A=1;
    B=1;
    #10;
  end
endmodule
```

NAND GATE :-

$C = \overline{A \cdot B}$



Truth Table :-

Input		Output
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

Code :-

i) Design module :-

```
module nand2
  (input A,
   input B,
   Output C
   );
assign C = ! (A & B);
 end module
```

Name	Value	0 ns	500
A	1		
B	1		
C	0		

ii) Test Bench :-

```
timescale 1ns/10ps
module nand2_tb;
reg A, B;
wire C;
nand2 uut (.A(A), .B(B), .C(C));
initial
  begin
    A = 0;
    B = 0;
    #10;
    A = 1;
    B = 0;
    #10;
    A = 0;
    B = 1;
    #10;
    A = 1;
    B = 1;
    #10;
  end
end module
```

NOR GATE :-

$C = \overline{A+B}$

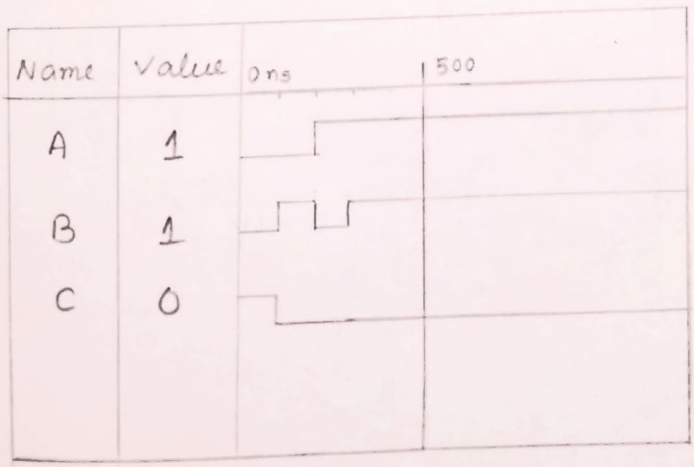
Truth Table :-

Input		Output
A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

Code :-

i) Design module :-

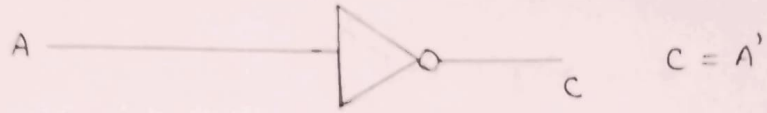
```
module nor2
(input A,
 input B,
 Output C
 );
assign C = !(A|B);
end module
```



ii) Test bench :

```
timescale 1ns/10ps
module nor2_tb;
reg A, B;
wire C;
nor2 uut(.A(A),.B(B),.C(C));
initial
  begin
   A=0;
   B=0;
   #10;
   A=0;
   B=1;
   #10;
   A=1;
   B=0;
   #10;
   A=1;
   B=1;
   #10;
   end
end module
```

NOT GATE :-



Truth Table :-

Input	Output
A	C
0	1
1	0

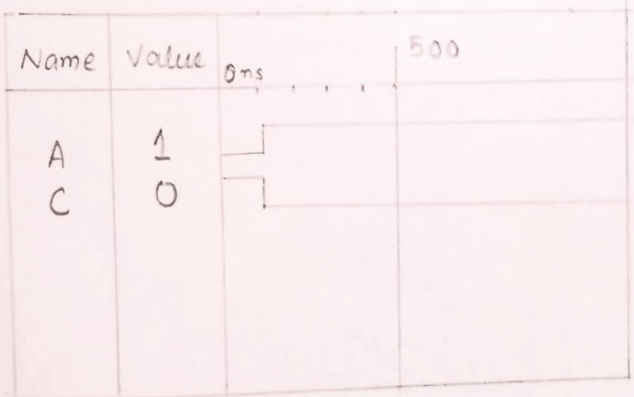
Code :-

i) Design module :-

```
module not2
  (input A,
   Output C
  );
assign C =! A;
 end module
```

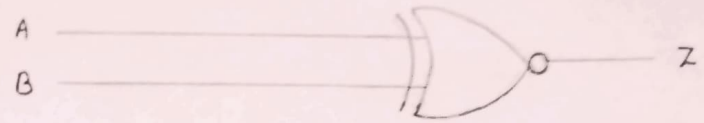
ii) Test Bench :-

```
timescale 1ns/10ps
module not2_tb;
 reg A;
 wire C;
not2 uut (.A(A), .C(C));
initial
   begin
    A = 0;
    #10;
    A = 1;
    #10;
    end
end module
```



XNOR GATE :-

$Z = \overline{A \oplus B}$



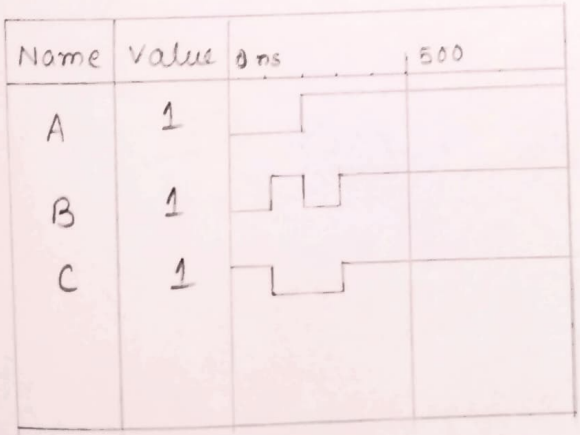
Truth Table :-

Input		Output
A	B	Z
0	0	1
0	1	0
1	0	0
1	1	1

Code :-

i) Design module :-

```
module xnor2 (
  input a,
  input b;
  Output z
  );
  assign z = ~(a^b);
   end module
```



ii) Test Bench :-

```
module xnor2_testbench;
reg a,b;
wire z;
xnor2 out (.a(a), .b(b), z(z));
initial
   begin
    a=0;
    b=0;
    #100;
    a=0;
    b=1;
    #100;
    a=1;
    b=0;
    #100;
    a=1;
    b=1;
    #100;
  end
initial
   begin
    $dumpfile("dump.vcd");
    $dumpvars();
     end
end module
```