

HDL implementation :-

HDL - Hardware descriptive language.

- 1) Design
- 2) Simulation - Will take input by test bench wave form
- 3) Synthesis

It is software used to simulate the logic design
Three main characteristics

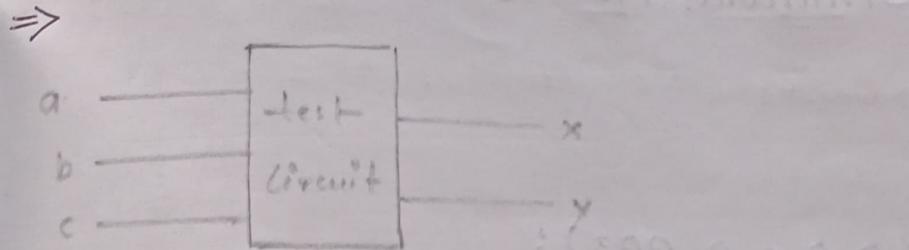
- 1) Design
- 2) Simulation
- 3) Synthesis

HDL is divided into 2 types

- 1) Verilog

2) VHDL - Very high speed integrated circuit of
Hardware descriptional language.

→ Verilog is more simple compared to VHDL.
So we mostly use Verilog. And the Software
used XILINX Software (7.1 Version)



\Rightarrow module mod-test(a,b,c,x,y);
 input a,b,c;
 output x,y;
 : module body
 endmodule

AND gate :-

```

graph LR
    a --- g1((g1))
    b --- g1
    g1 -- c --> null
  
```

module gate-test1(a,b,c);
 input a,b;
 output c;
 AND g1(c,a,b);
 endmodule.

OR gate :-

```

graph LR
    a --- g1((g1))
    b --- g1
    g1 -- y --> null
  
```

module or-gate(a,b,y);
 input a,b;
 output y;
 OR g1(y,a,b);
 endmodule

NOT gate :-

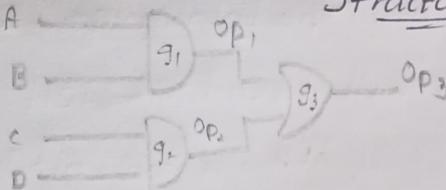
```

graph LR
    a --- g1((g1))
    g1 -- y --> null
  
```

module not-gate(a,y);
 input a;
 output y;
 NOT g1(y,a);
 endmodule.

Structural Model :-

1)



module gate-test2 (A,B,C,D,OP3);

input A,B,C,D ;

Output OP3 ;

wire OP1, OP2 ; internal Connection

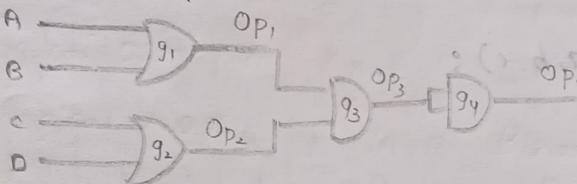
AND g1 (OP1, A,B) ;

AND g2 (OP2, C,D) ;

OR g3 (OP3, OP1, OP2) ;

endmodule

2)



module gate-test3 (A,B,C,D,OP4);

input A,B,C,D ;

Output OP4 ;

wire OP1, OP2, OP3 ;

OR g1 (OP1, A,B) ;

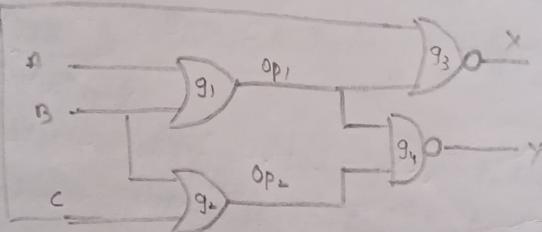
OR g2 (OP2, C,D) ;

AND g3 (OP3, OP1, OP2) ;

AND g4 (OP4, OP3) ;

endmodule.

3)



module gate-test3 (A,B,C,X,Y);

input A,B,C ;

Output X,Y ;

wire OP1, OP2 ;

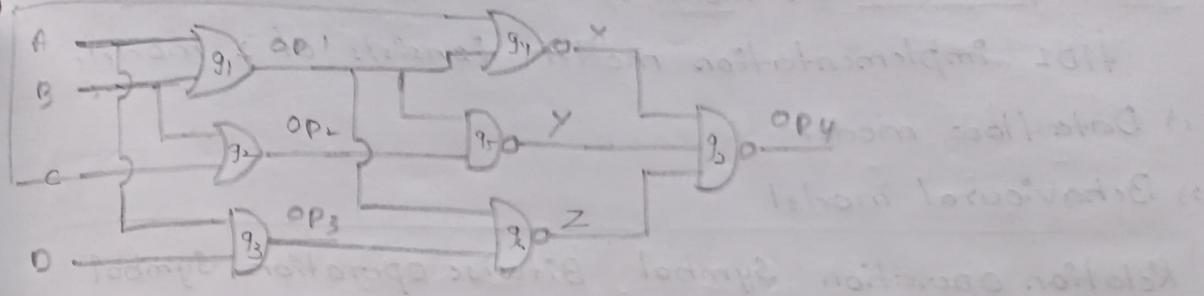
OR g1 (OP1, A,B) ;

OR g2 (OP2, B,C) ;

NOR g3 (X, OP1, C) ;

NAND g4 (Y, OP1, OP2) ;

4)



module gak-test (A,B,C,D,OP4);

input A,B,C,D;

output OP4;

wire OP1,OP2,OP3,X,Y,Z;

OR g1 (OP1, A, B);

OR g2 (OP2, B, C);

AND g3 (OP3, A, D);

NOR g4 (X, C, OP1);

NAND g5 (Y, OP1, OP2);

NAND g6 (Z, OP1, OP3);

NAND g7 (OP4, X, Y, Z);

endmodule

5) A

module gate-test (A,B,C,D,X);

input A,B,C,D;

output X;

wire OP1,OP2,OP3,OP4,OP5;

NOR g1 (OP1, A, B);

- NOR g2 (OP2, C, B);

AND g3 (OP3, A, D);

NOR g4 (OP4, OP1, OP2);

NAND g5 (OP5, OP1, OP3);

AND g6 (X, OP4, OP5);

endmodule

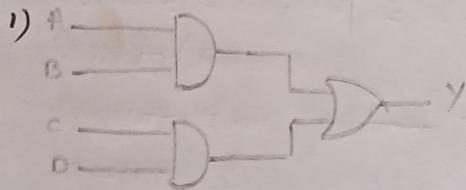
HDL Implementation model :-

HDL implementation model consists of 2 models,

- 1) Data-flow model
- 2) Behavioural model

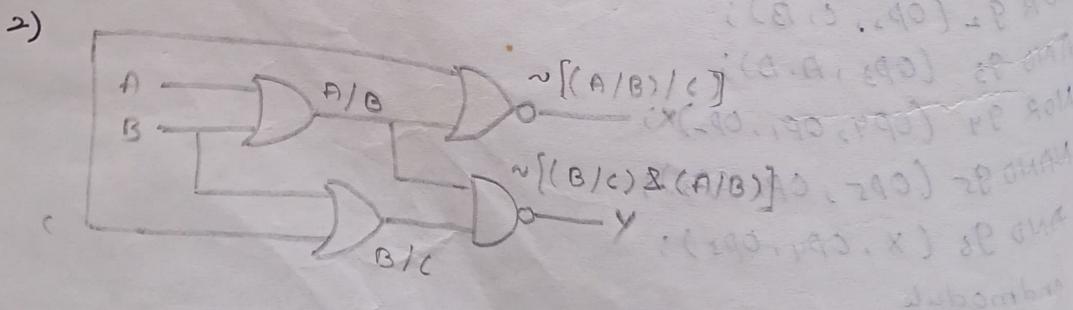
Relation operation	Symbol	Bitwise operation	Symbol
less-than	<	Bitwise NOT	~
less than or equal to	<=	Bitwise AND	&
Greater than	>	Bitwise OR	
equal to	= =	Bitwise EX-OR	^
NOT equal to	!=		

Logical operation	Symbol	Bitwise operation	Symbol
logical NOT	!	Binary addition	+
logical AND	&&	binary subtraction	-
logical OR		binary multiplication	*
		binary division	/



$$AB + CD.$$

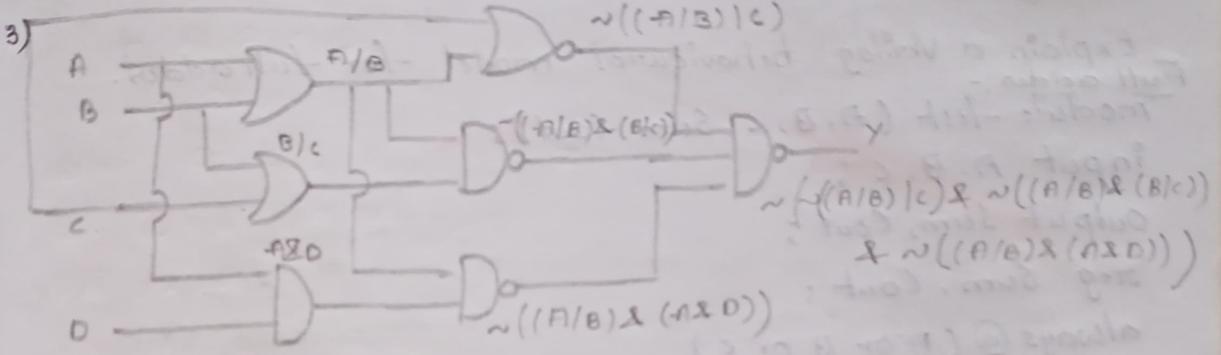
```
module test1(A,B,C,D,Y);
input A,B,C,D;
output Y;
assign Y = ((A&B)|(C&D));
endmodule
```



```

module test2 (A,B,C,X,Y);
  input A,B,C;
  output X,Y;
  assign X = ~((A|B)|C);
  assign Y = ~((A|B)&(B|C));
endmodule

```



```
module test3 (A,B,C,D,Y);
```

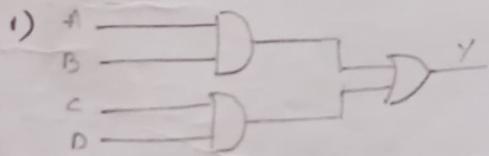
```
input A,B,C,D;
```

```
output Y;
```

```
assign Y = ~((~(C|(A|B)))&(~((A|B)&(B|C))&(~((A|B)&(A&D))))
```

```
endmodule
```

Behavioural Model :-



```
module test (A,B,C,D,Y);
```

```
input A,B,C,D;
```

```
Output Y;
```

```
neg Y;
```

```
always @ (A or B or C or D)
```

```
if ((A==1)&&(B==1))
```

Y=1;

```
else if ((C==1)&&(D==1))
```

Y=1;

```
else
```

Y=0;

```
endmodule
```

Realize the expression $y = C' + A'D' + B'D'$ using data flow. $y = \sim c | (\sim a \& \sim d) | (\sim b \& \sim d)$

```
module test (A, B, C, D, Y);  
    input A, B, C, D;  
    output Y;  
    assign Y = (\sim c | (\sim a & \sim d) | (\sim b & \sim d));  
endmodule
```

Explain a Verilog behavioural model for full adder.

Full adder:-

```
module test (A, B, C, Sum, Cout);  
    input A, B, C;  
    output Sum, Cout;  
    reg Sum, Cout;  
    always @ (A or B or C)  
        begin
```

$$\text{Sum} = A^{\wedge}B^{\wedge}C;$$

$$\text{Cout} = (A \& B) | (A \& C) | (B \& C);$$

end

endmodule

full adder :-

$$\text{Sum} = A \oplus B \oplus C$$

$$\text{Cout} = \sim A B + B C + A C$$

data flow :-

```
module test (A, B, C, Sum, Cout);  
    input A, B, C;  
    output Sum, Cout;  
    assign Sum = A^{\wedge}B^{\wedge}C;  
    assign Cout = (A \& B) | (A \& C) | (B \& C);  
endmodule
```

\Rightarrow Half adder :-

data flow :-

```
module test (A, B, Sum, Cout);  
    input A, B;  
    output Sum, Cout;  
    assign Sum = A^{\wedge}B;  
    assign Cout = A \& B;  
endmodule
```

Half adder :-

$$\text{Sum} = A \oplus B$$

$$Cout = \text{Carry}(AB)$$

behavioural model :-

```

module test (A, B, C, Diff, Bout)
input A, B, C;
output Diff, Bout;
Reg Diff, Bout;
always @ (A or B or C)
begin
    Diff = A ^ B ^ C;
    Bout = (~A & B) | (~A & C) | (B & C)
end
endmodule

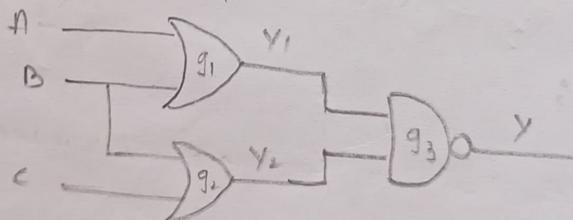
```

Draw a timing diagram and write a Verilog HDL code using structural model for the boolean function.

$$Y = \text{NAND}(Y_1, Y_2)$$

$$Y_1 = A + B$$

$$Y_2 = B + C$$



```

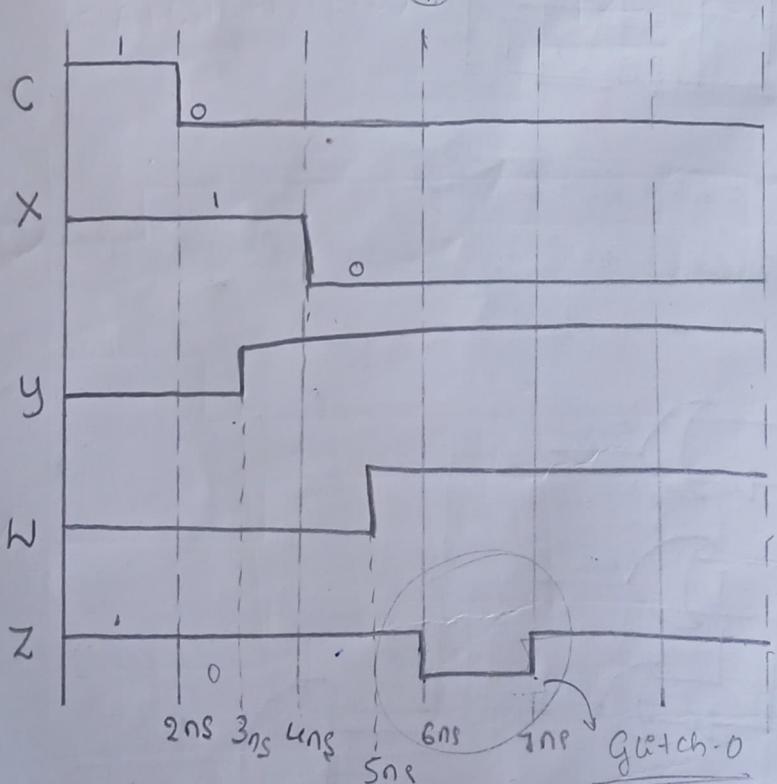
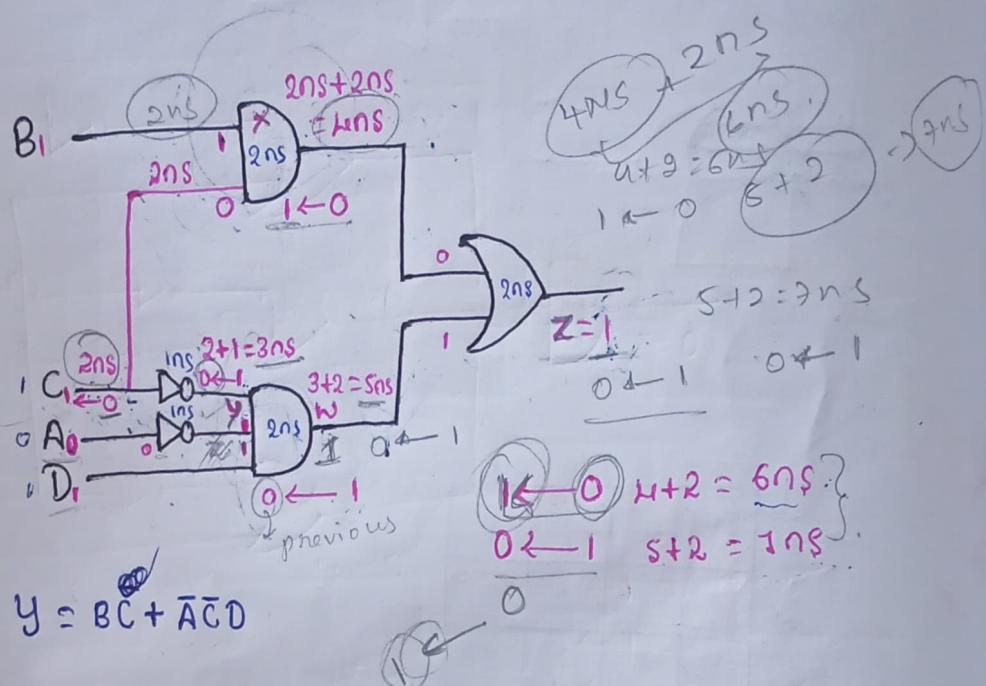
module test (A, B, C, y);
input A, B, C;
output y;
wire Y1, Y2;
OR g1 (Y1, A, B);
OR g2 (Y2, B, C);
NAND g3 (y, Y1, Y2);
endmodule

```

A	B	C	y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

AND-OR

For the following circuit in verilog has a delay of 1 ns and other gates has a delay of 2 ns initially $a=0, b=c=d=1$. C changes to at 2 ns. Draw the timing diagram and identify the transient occurring in the modified circuit diagram. (1-Hazard)



Timing diagram

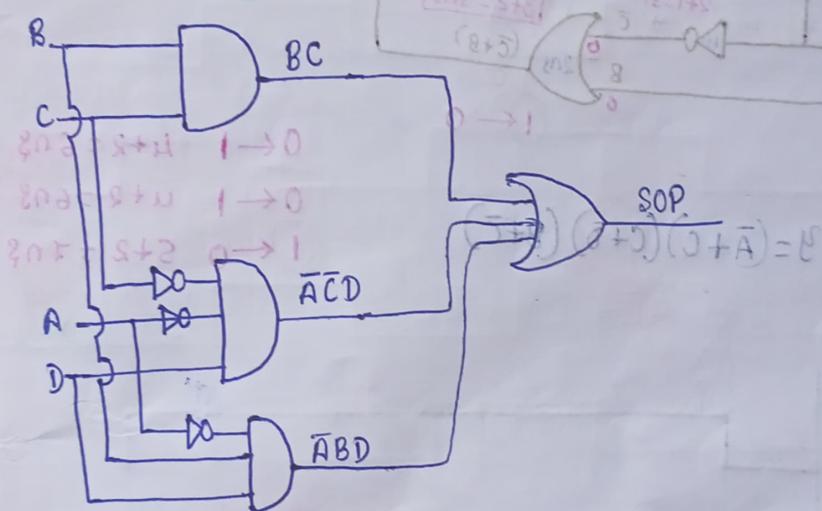
Hazard - 1

②

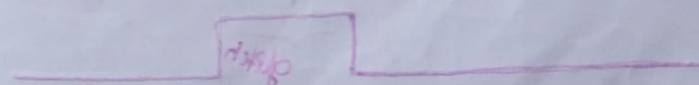
\bar{AB}	\bar{CD}	CD	\bar{D}	\bar{CD}
\bar{AB}	0	1	0	0
\bar{AB}	0	1	1	1
\bar{AB}	1	0	1	1
\bar{AB}	1	0	0	0

$$y = BC + \bar{A}\bar{C}D + \bar{A}BD$$

Hazard free circuit

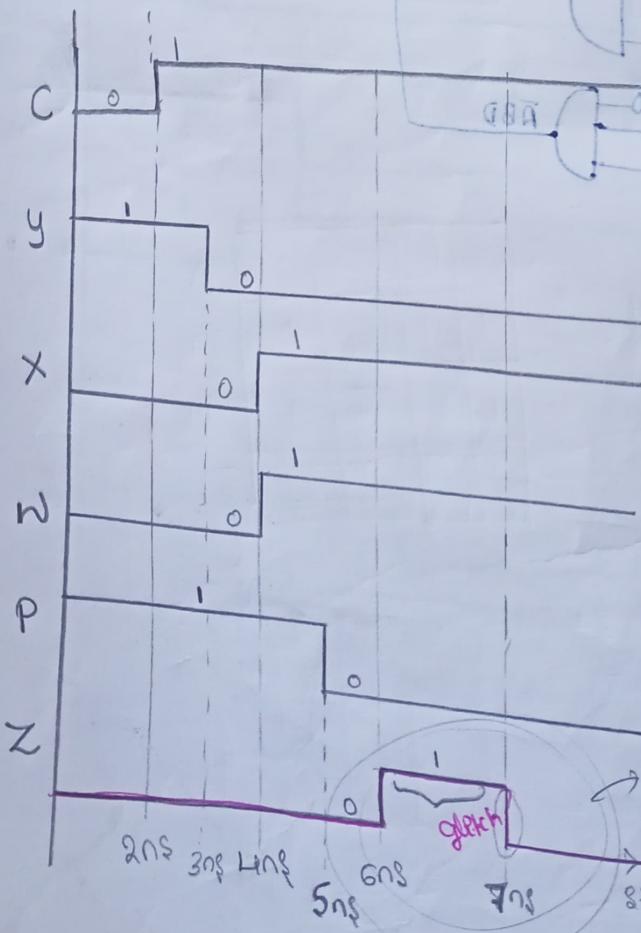
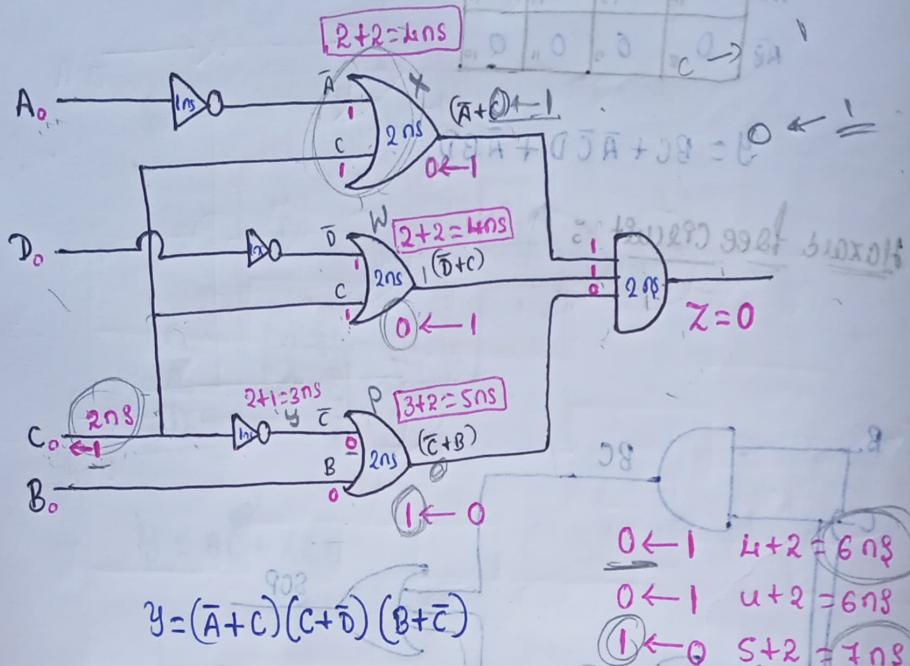
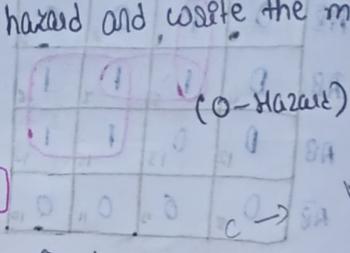


①



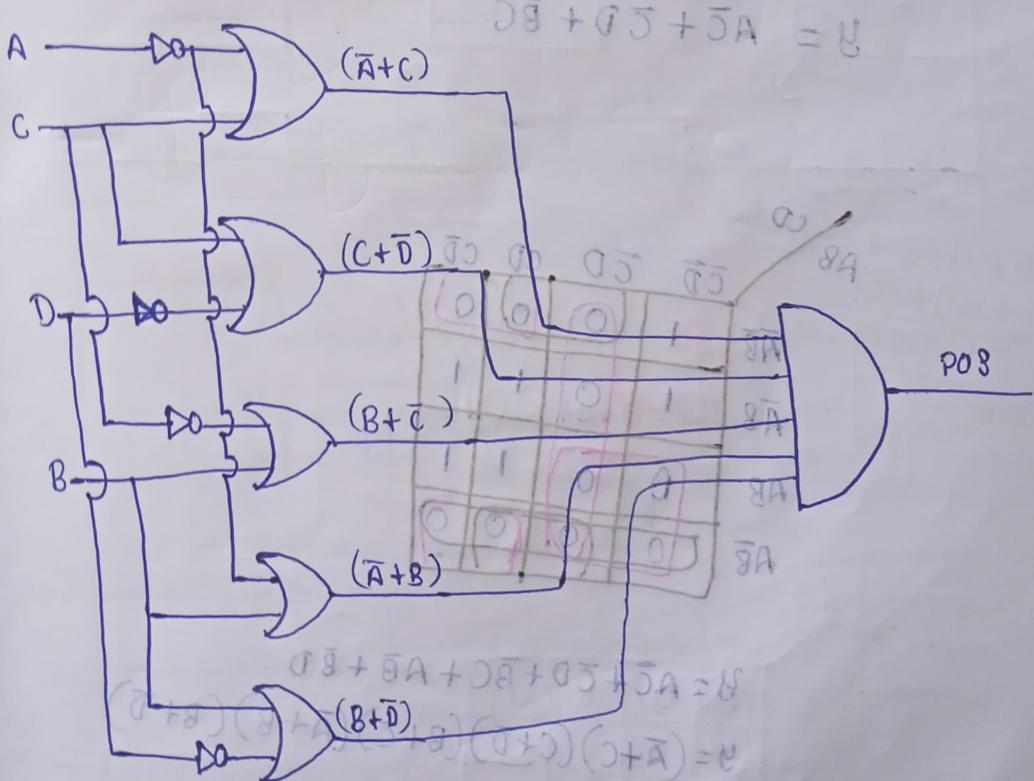
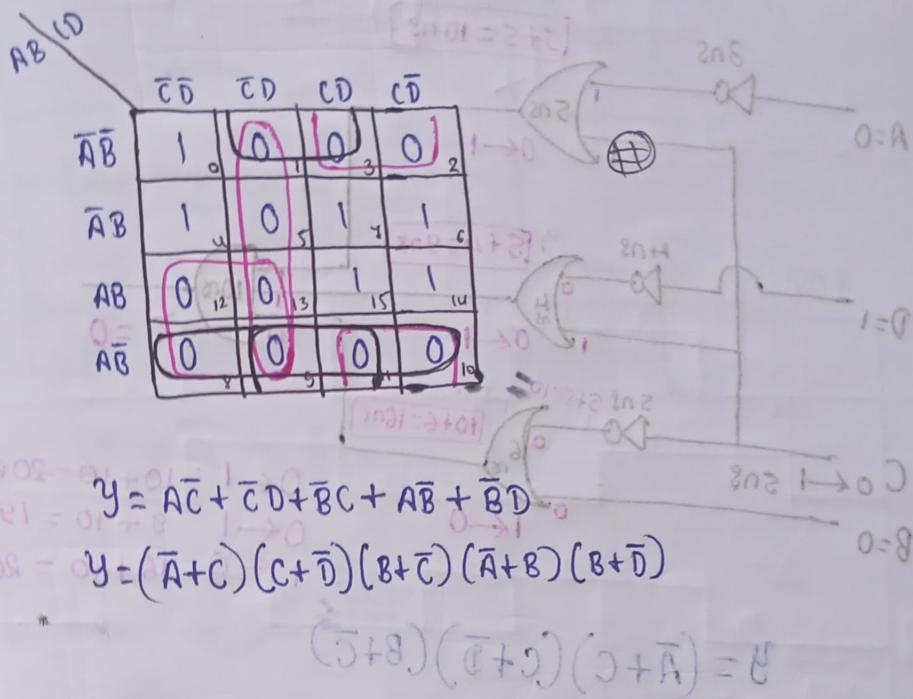
Q

Assume the inverter has a delay of 1 ns and other gates has a delay of 2 ns initially $a=b=c=d=0$. Ct changes to 1 at 2 ns. Draw the timing diagram and identify the hazard and write the modified circuit hazard free circuit.

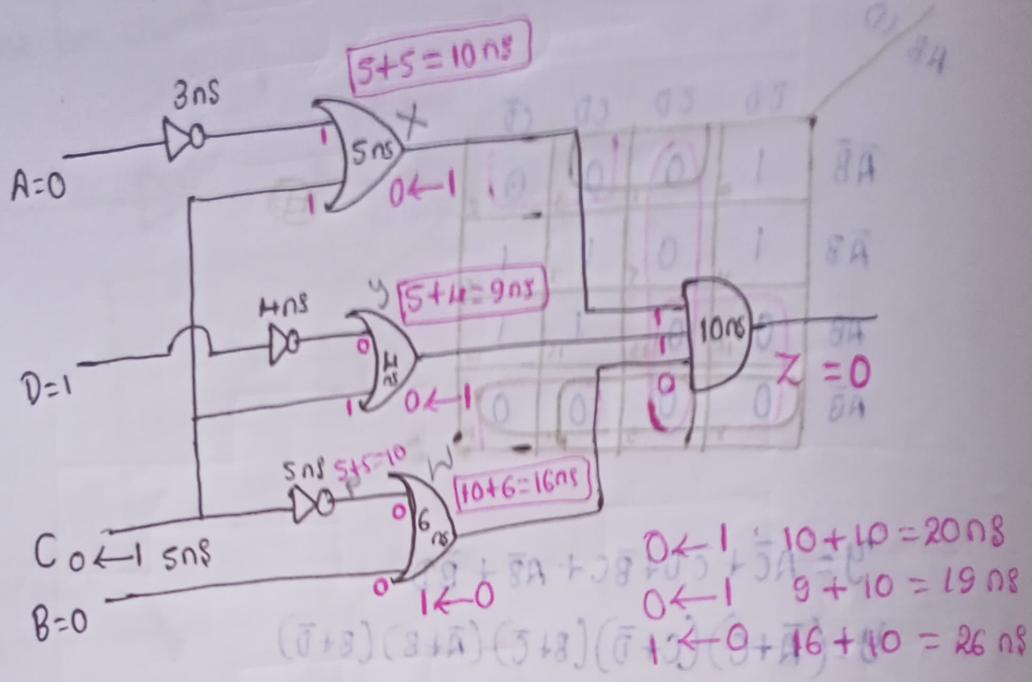
OR \rightarrow AND

$$Y = (\bar{A} + C)(C + \bar{D})(B + \bar{C})$$

$$Y = A\bar{C} + \bar{C}D + \bar{B}\bar{C}$$

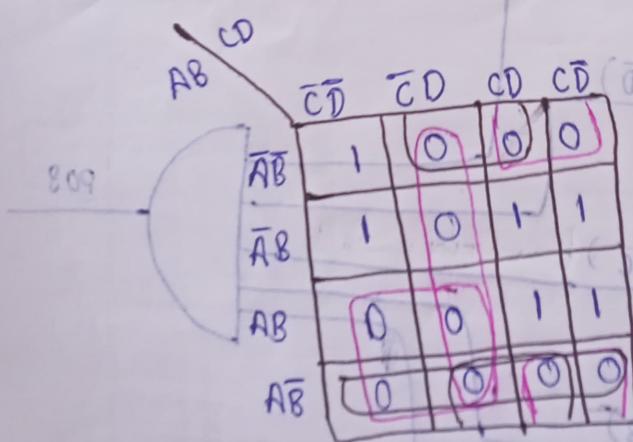


③ Design the Hazard's Spec'd. [0' Hazard]



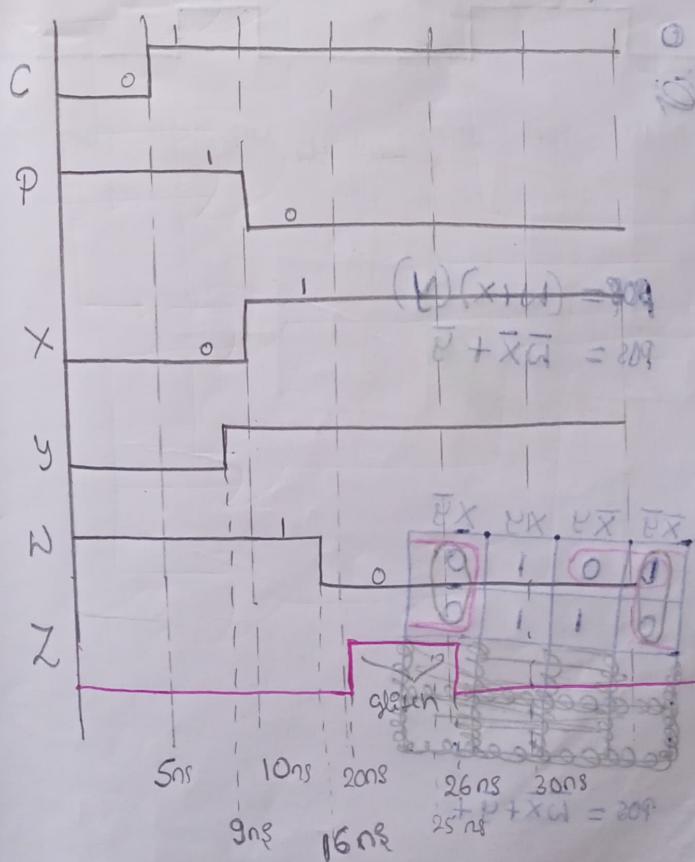
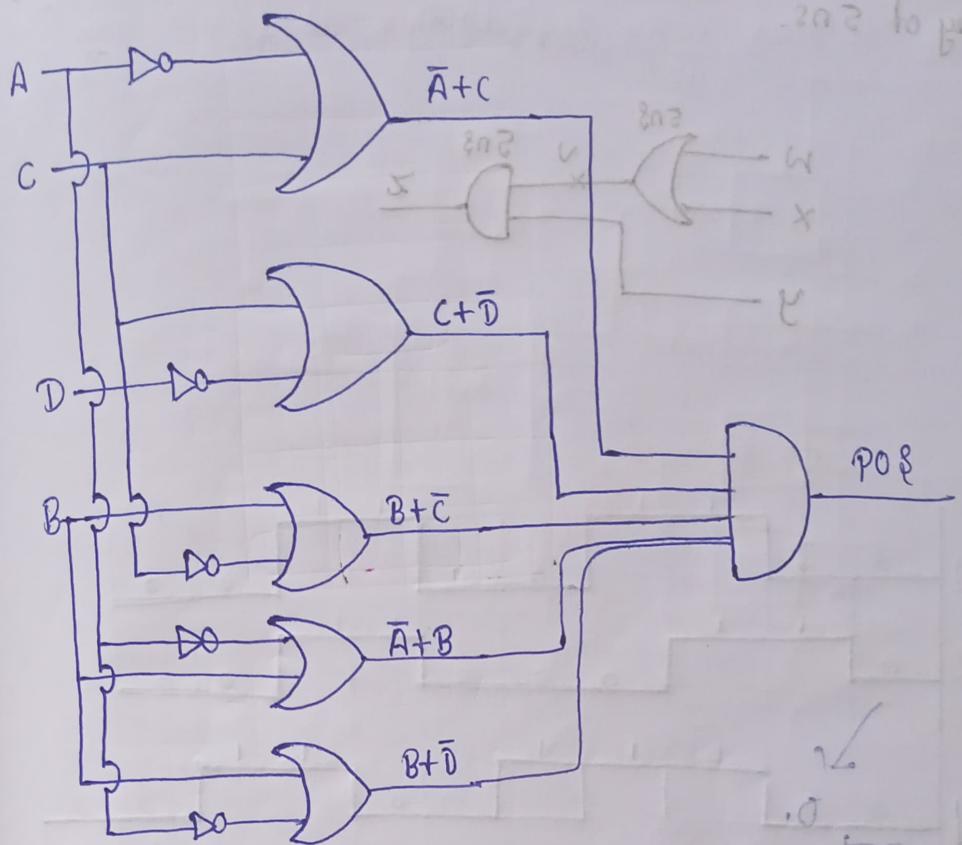
$$y = (\bar{A} + C)(C + \bar{D})(B + \bar{C})$$

$$y = A\bar{C} + \bar{C}D + \bar{B}C$$

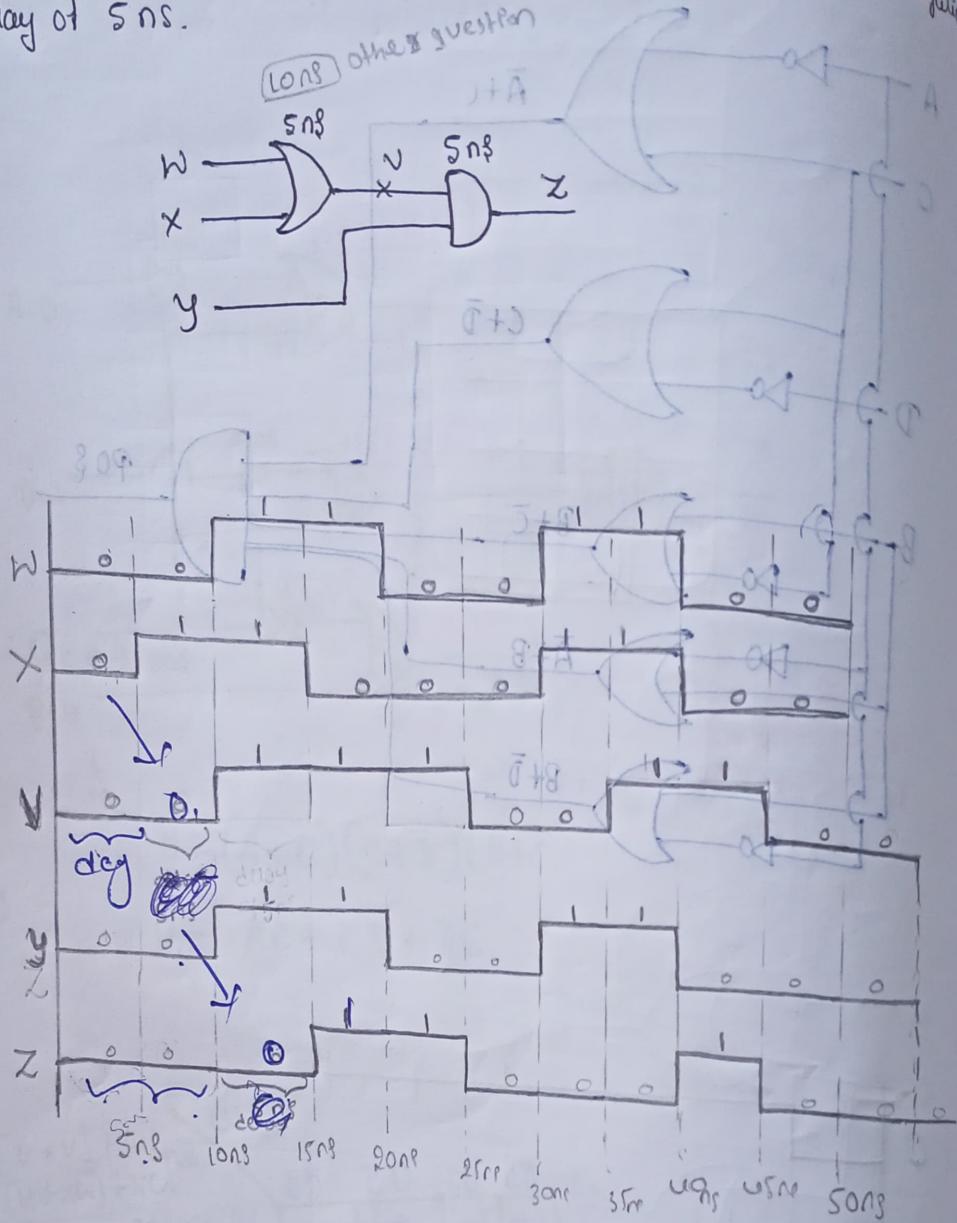


$$y = A\bar{C} + \bar{C}D + \bar{B}C + A\bar{B} + \bar{B}D$$

$$y = (\bar{A} + C)(C + \bar{D})(B + \bar{C})(\bar{A} + \bar{B})(B + \bar{D})$$



Compute the timing diagram. Assume both the gates has propagation delay of 5 ns.



$$POP = (W+X)(Y)$$

$$POS = \bar{W}\bar{X} + \bar{Y}$$

	$\bar{W}\bar{X}Y$	$\bar{W}XY$	$W\bar{X}Y$	$WX\bar{Y}$
\bar{W}	0	0	1	0
W	0	1	1	0
X	0	0	0	1
Y	0	1	1	0

$$POS = \bar{W}\bar{X} + \bar{Y} +$$