Patinkar

Hardware description language

Page:
Date: 6-Jan

Verilog → hardware description language

Why HDL's?

- in software everything is sequential.
- in chip transistors are used not the gate.
  RTL → register transfer language.

Abstractions:

→ if & else
Behavioral (how I/P will be behave for a particular O/P)

↓

Data flow (boolean expression)

↓

Structural (if we want 4 bit adder by using and connect them we can make it)

One struc.

↓

Switch (in s/w nand gate → How many cmos are required)

- Behavioral methods are more preferable for design the chip.

Scanned by CamScanner

Advantages of HDL :

- designs can be described at various level of abstraction.
- Top-down approach
- functional simulation early in design flow
- automatic conversion of HDL to gates
- testing of various design implementation
- design reuse

- VHDL → very high speed integrated circuit hardware description language.
- veryllog HDL (Verification of logic)

Digital s/s design flow :

(from slide)

synthesis → our design to technology design.

Types of simulation:

① Eventdriven : when a change on an i/p causes a change on o/p, which cause a further change on another i/p)

Verilog HDL :

- verification of logic
- Used in Simulation and synthesis.
- verilog is case sensitive
- Keywords are lowercase
- statement end with semicolumn

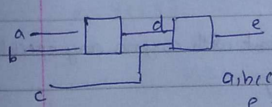
Verilog program structure :

module < module_name > (Port list)

① port declaration
(how much i/Ps or o/Ps, clock, enable Pin are in coming in the port dec)

② data type declaration
data type int, float

wire & register



a, b, c → i/p Port
e → o/p Port

a,b,c → are wire bcoz they don't handle any having value.

d → having the value register

③ Circuit functionally
(logic of program)

④ Time specification
end module

for example:

```
module and2 (a,b,c);        * module not_gate(in,out);
input = a,b;                   input in;
o/p = c;                       output out;
assign c = a or b;             assign out = ~in;
end module                     end module
```

operator	Name
&	and
|	OR
^	XOR
~&	NAND
~|	NOR
~^	NOT XOR
~	NOT

```
* module useless;
initial (keyword)
$ Display (" Hello! ");
endmodule
```

- module name can't be start with any no. any special char and it can be anything.

Number System

9-Jan

$r = 2, 8, 10, 16$

binary, octal, decimal, hexadecimal

any base ⟶ decimal

$$\left[a_n r^n + a_{n-1} r^{n-1} + a_{n-2} r^{n-2} + \ldots\ldots a_1 r + a_0 \right]$$

a_n → base, r^n → position

for example

(i) $(1010.011)_2 \longrightarrow (\quad)_{10}$

$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 2^{-2} + 2^{-3}$

$= 8 + 2 + \frac{1}{4} + \frac{1}{8}$

$= 10 + \frac{3}{8} = 10 + 0.375$

$= 10.375$

(ii) $(4021.2)_5$

$4 \times 5^3 + 2 \times 5^1 + 1 + 2 \times \frac{1}{5}$

$= 511.4$

(iii) $(B65F)_{16}$

$11 \times 16^3 + 6 \times 16^2 + 5 \times 16^1 + 15$

$= 46,687$

Scanned by CamScanner

Mondays ? why we do complement to do subtraction ?

$(41)_{10} \rightarrow (\quad)_2$

divided by that particular base, which we required to base in convert

r's and (r-1)'s complement addition

for ex: i) 72532 − 03250 (M = 72532, N = 03250)

03250 ⟶ 96749
+1
10's comp → 96750

72532
+ 96750
(1) 69282

carry is ignored

- if carry is generated → ignored
- if carry is not generated then take the r's complement of the ans and put −ve sign infront of the ans.

ii) 1010100 + 1000100 = 0011000

1010100
+ 1000100
(1) 0011000
↳ ignored

(r-1)'s complement method:

iii) 1010100 − 1000100 (r's complement)

1000100 ⟶ 2's comp

0111011
+1
0111100

1010100 → m
+ 0111100 → 2's
(1) 0010000 ⟶
↳ ignored

By method r's & (r-1)'s ans would be same.

(r-1)'s complement:

1010100 → m
+ 0111011 → 1's comp.
(1) 0001111
+1
0010000 ⟶

- if carry is generated → add carry to LSB postion of ans.
- if carry is not generated → take (r-1)'s comp. of ans and put (−) sign.

	Sign magnitude	2's comp	1's comp.
i) +51	0110011	0110011 ~~1001101~~	0110011 ~~1001100~~
(ii) −51	1110011	1001101	1001100

* sign bit same rahegi

* 2's and 1's comp. of +ve no can't be find its also same as binary representation of pos no.

2	51	
2	25	1
2	12	1
2	6	0
2	3	0
2	1	1
	0	1

$(51)_{10} = (110011)_2$

Ex: +73.75 12 bit 2's comp form

-73.75

2	73	
2	36	1
2	18	0
2	9	0
2	4	1
2	2	0
2	1	0
	0	1

$.75 \times 2 = 1.5$

$.5 \times 2 = 1.0$

$.0 \times 2 = 0$

$(01001001.1100)_2$

-73.75 → 01001001.1100

↓

10110110.0011 → 1's comp

↓ 0100

+1 +1

10110111.0100

for 2's comp. treat real and fractional part separately.

.1100 ←	01001001 ←
.0100	10110111 (by fliping)

Ex:

```
110 ) 101101 ( 111.1
      110
      1010
       110
       1001
        110
        0110
         110
         000
```

```
101 ) 110101111 ( 101011
      101
      00110
        101
        00111
          101
          101
          101
          000
```

Ex: 1101 × 110

```
    1101
  ×  110
    0000
   11010
  110100
 1001110
```

Date: 10-Jan

Boolean Algebra:

Demorgan's: $\overline{A+B+C+\dots} = \bar{A}\cdot\bar{B}\cdot\bar{C} \dots$

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

OR:

$A+0=A$

$A+\bar{A}=1$

$1+A=1$

$A+A=A$

AND:

$A\cdot 0=0$

$A\cdot 1=A$

$A\cdot\bar{A}=0$

$A\cdot A=A$

commulative: $A+B=B+A$ $\quad A\cdot B=B\cdot A$

Associative: $(A+B)+C = A+(B+C)$ $\quad (A\cdot B)\cdot C = (A\cdot B)\cdot C$

$A(B+C) = AB+AC$

$A+BC = (A+B)(A+C)$

$\bar{\bar{A}} = A$

$A+AB = A$ } Absorption law

$A(A+B) = A$

Examples:

1) $A[B+C(\overline{AB+AC})]$

$\overline{AB}\cdot\overline{AC}$

$A[B+C(\bar{A}+\bar{B})(\bar{A}+\bar{C})]$

$A(B+C(\bar{A}+\bar{B}\bar{C}))$

$A(B+\bar{A}C)$

AB

2) $A\bar{B}C + AB\bar{C} + ABC$

3) $\bar{A}\bar{C} + B\bar{C} + \bar{A}BC + ABC$

4) $A\bar{B}\bar{D} + AB\bar{C}\bar{D} + ABC\bar{D} + A\bar{B}\bar{D} + \bar{A}BCD$

5) $B\bar{C} + A\bar{C} + AB + BCD$

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

7) $AD + A\bar{B}\bar{C} + \bar{B}C\bar{D} + \bar{A}CD + \bar{A}BC\bar{D}$

2) $A\bar{B}C + AB\bar{C} + ABC$

$A(\bar{B}C + B\bar{C} + BC + BC)$

$A(B+C)$

3) $\bar{A}\bar{C} + B\bar{C} + BC$

$\bar{A}\bar{C} + B$

4) $A\bar{B}\bar{D} + AB\bar{D} + A\bar{B}\bar{D} + \bar{A}BCD$

$A\bar{B}\bar{D} + AB\bar{D} + \bar{A}BCD$

$A\bar{D} + \bar{A}BCD$

5) $BC + A\bar{C} + AB(C+\bar{C})$

$= ABC + AB\bar{C} + A\bar{C} + BC$

$= BC + A\bar{C}$

Page:
Date:

$A + \bar{B}CD$

(6) $\because (A+B)(A+C) = A+BC$

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

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

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

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

$A+\bar{B}C\cdot\bar{C}+\bar{B}CD$

$A+\bar{B}CD$

(7) $AD + A\bar{B}\bar{C} + \bar{B}C\bar{D} + \bar{A}CD + \bar{A}BC\bar{D}$

Page:
Date: 16-jan

SOP & POS :

x	y	z	min term		max term
0	0	0	$\bar{x}\bar{y}\bar{z}$	m_0	$x+y+z$
0	0	1	$\bar{x}\bar{y}z$		$x+y+\bar{z}$
0	1	0	$\bar{x}y\bar{z}$		$x+\bar{y}+z$
0	1	1	$\bar{x}yz$		$x+\bar{y}+\bar{z}$
1	0	0	$x\bar{y}\bar{z}$		$\bar{x}+y+z$
1	0	1	$x\bar{y}z$		$\bar{x}+y+\bar{z}$
1	1	0	$xy\bar{z}$		$\bar{x}+\bar{y}+z$
1	1	1	xyz	m_7	$\bar{x}+\bar{y}+\bar{z}$

NAND $\longrightarrow \overline{A\cdot B} \longrightarrow$ min term (more preferable)

NOR $\longrightarrow \overline{A+B} \longrightarrow$ max term

Example :

(1) $F = A+\bar{B}C$

$A(B+\bar{B}) = (AB+A\bar{B})(C+\bar{C}) = ABC + A\bar{B}C + AB\bar{C} + A\bar{B}\bar{C}$

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

$A+\bar{B}C = ABC + AB'C + ABC' + AB'C' + AB'C + A'B'C$

$= \Sigma m(7, 5, 6, 4, 1) \longrightarrow$ min term

max term $(0, 2, 3)$

(2) $F = xy + \bar{x}z$

~~$xy(1+z) + \bar{x}z(1+y)$~~

~~$= xy + xyz + \bar{x}z + \bar{x}yz$~~

POS:

$$xy + \bar{x}z$$
$$= (xy + x')(xy + z)$$
$$= (x + x')(y + x')\ (x + z)(y + z)$$
$$= (x' + y)\ (x + z)\ (y + z)$$
$$= (x' + y + z)(x' + y + z')\ (x + y + z)(x + y' + z)\ (x + y + z)$$
$$(x' + y + z)$$

$$(x' + y)(\ = \ x' + y + z \cdot \bar{z}$$
$$= (x' + y + z)\ (x' + y + z')$$

SOP:

$$xy + \bar{x} z$$
$$xy = xyz + xy\bar{z}$$
$$\bar{x}z = \bar{x}yz + \bar{x}\bar{y}z$$
$$xy + \bar{x}z = \Sigma m(7, 6, 3, 1)$$

K-Map:

Ex:

$$F = x'yz + x'yz' + xy'z' + x'y'z'$$

$$2^3 = 8$$

$x \backslash yz$	$\bar{y}\bar{z}$ 00	$\bar{y}z$ 01	$y\bar{z}$ 10	yz 11
$\bar{x}$ 0	m_0 1	m_1	m_3 1	m_2 1
x 1	m_4 1	m_5	m_7	m_6

$2^3 = 8 \rightarrow 1$

$2^2 = 4 \rightarrow 1$ - variable

$2^1 = 2 \rightarrow 2$ variable in each pair

$$\bar{y}\bar{z} + \bar{x}y$$

4-Variable K-Map:

$$F(w, x, y, z) = \Sigma(0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14)$$

$wx \backslash yz$	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{w}\bar{x}$	1	1	0	1
$\bar{w}x$	1	1	0	1
wx	1	1	0	1
$w\bar{x}$	1	1	0	0

1 pair of 8 $= \bar{y}$

2 pair of 4 $= \bar{w}\bar{z}, x\bar{z}$

$$F = \bar{y} + \bar{w}\bar{z} + x\bar{z}$$
$$= (\bar{y} + \bar{z})(\bar{y} + \bar{w} + x)$$

Ex:- $F(A, B, C, D) = \Sigma m(0, 1, 2, 5, 8, 9, 10)$

POS & SOP

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	0	1
$\bar{A}B$	0	1	0	0
AB	0	0	0	0
$A\bar{B}$	1	1	0	1

$$F = \bar{B}\bar{C} + \bar{B}\bar{D} + \bar{A}\bar{C}D \longrightarrow \text{min term SOP}$$
$$= (\bar{A} + \bar{B})(\bar{C} + \bar{D})(D + \bar{B}) \longrightarrow \text{max term POS}$$

Ex:- $F(w, x, y, z) = \Sigma m(1, 3, 7, 11, 15) + d(0, 2, 5)$

	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{w}\bar{x}$	X	1	1	X
$\bar{w}x$	0	X	1	0
wx	0	0	1	0
$w\bar{x}$	0	0	1	0

$$F = yz + \bar{w}\bar{x} \longrightarrow \text{SOP}$$

$z\bar{w}$

$$= z(y + \bar{w}) \longrightarrow \text{POS}$$

Date: 17-jan

Ex:- $F(A, B, C, D, E) = \Sigma(0, 2, 4, 6, 9, 11, 13, 15, 17, 21, 25, 27, 29, 31)$

BC \ DE	$\bar{D}\bar{E}$	$\bar{D}E$	DE	$D\bar{E}$
$\bar{B}\bar{C}$	1			1
$\bar{B}C$	1			1
BC		1	1	
$B\bar{C}$		1	1	

A = 0

$\bar{A}$

BC \ DE				
		1		
		1		
		1	1	
		1	1	

A = 1

$$F = \bar{A}BE + ABE + A\bar{D}E \quad + \bar{A}\bar{B}\bar{E}$$

$$= BE$$

Map entered variable :

	A	B	C	D	E	F	G (O/P)
m_0	0	0	0	0	X	X	1
m_1	0	0	0	1	X	X	X ← Don't care O/P
m_2	0	0	1	0	X	X	1
m_3	0	0	1	1	X	X	1
m_5	0	1	0	1	1	X	1
m_7	0	1	1	1	1	X	1
m_9	1	0	0	1	X	1	1
m_{11}	1	0	1	1	X	X	1
m_{12}	1	1	0	0	X	X	X ←
m_{13}	1	1	0	1	X	X	X ←
m_{15}	1	1	1	1	X	X	1

$$G(A, B, C, D, E, F) = m_0 + m_5 + m_2 + m_3 + E\,m_5 + E\,m_7 + F\,m_9 + m_{11} + d(m_1, m_{12}, m_{13})$$

in case of m_7, E term is there

in case of m_9 F term is 1

max. term is m_{15} → so 4-variable K-map

AB \ CD				
	1	X	1	1
		E·1	E	
	X	X	1	
		F	1	

Standard equation:

P → is the no. of variables

$$F = \left(MS_0 + \underset{E}{P_1} \cdot MS_1 + \underset{F}{P_2}\, MS_2\right) + \cdots\; P_n + MS_n$$

it would be our ans. for upper given problem

MS_0 is the when E & F are not there and solve the K map solution.

m_{S0}

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	X	1	1
$\bar{A}B$				
AB	X	X	1	
$A\bar{B}$			1	

– do not consider any variable (E & F)

$m_{S0} = A'B' + ACD$

m_{S1}

AB \ CD				
	X	X	X	X
		1	1	
	X	X	X	
			X	

↳ 1 is replaced by don't care
↳ don't care same as
↳ another variable is not present
↳ and that particular variable is replaced by 1

$P_1 M_{S1} = (\bar{A}\,D)E$

m_{S2}

X	X	X	X
X	X	X	
	1	X	

$P_2 M_{S2} = ADF$

SO; $F = A'B' + ACD + \bar{A}DE + ADF$

__Example__ $Z(A,B,C,D,E,F,G) = \Sigma m(2,5,6,9) + \Sigma d(1,3,13,14)$
$+ E(m_{11} + m_{12})$
$+ F(m_{10}) + G(m_0)$

$Z = m_{S0} + P_1 m_{S1} + P_2 m_{S2} + P_3 m_{S3}$

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	G	X	X	1
$\bar{A}B$		1		1
AB	E	X		X
$A\bar{B}$		1	E	F

m_{S0}

	X	X	1
	1		1
	X		X
	1		

$m_{S0} = \bar{C}D + \bar{A}C\bar{D}$

m_{S1}

	X	X	X
	X		X
1	X		X
	X	1	

$E\, m_{S1} = E\,[AB\bar{C} + [illegible]\bar{B}D]$

m_{S2}

	X	X	X
	X		X
	X		X
	X		1

$F\, m_{S2} = FC\bar{D}$

m_{S3}

1	X	X	X
	X		X
	X		X
	X		

$G\, m_{S3} = G\bar{A}\bar{B}$

Scanned by CamScanner

$$Z = C'D + A'CD' + ABC'E + EBD' + FCD' + GA'B'$$

Example :

$$G = C'E'F + DEF + AD'E'F' + BC'E'F + AD'EF'$$

Solve using map entered method.

$\bar C\bar D E\bar F$

$C\bar D E\bar F$

Sol :

two variable is A & B

$$= C'E'FD + C'E'F\bar D + CDEF + \bar C DEF + A[CD'E'F' + C'D'E'F'] + B[C'D'E'F + C'DE'F] + A[C'D'EF' + CD'EF']$$

$$= 0101 + 0001 + 1111 + 0111 + B[0001 + 0101] + A(1000 + 0000 + 0010 + 1010)$$

1+B=1

$$= \Sigma m(1, 5, 7, 15) + B(1+5) + A(0+2+8+10)$$

$$= m_5 + m_1 + m_7 + m_{15} + Bm_1 + Bm_5 + A(m_0 + m_2 + m_8 + m_{10})$$

23-Jan

$$= m_1 + m_5 + m_7 + m_{15} + A(m_0 + m_2 + m_8 + m_{10})$$

$$= \Sigma m(1, 15, 7, 5) + A(m_0 + m_8 + m_{10} + m_2)$$

CD \ EF	$\bar E\bar F$	$\bar E F$	EF	$E\bar F$
$\bar C\bar D$	A	1		A
$\bar C D$	1	1	1	
CD	[illegible]		1	
$C\bar D$	A			A

$$m_{s_0} = \bar C\bar E F + DEF$$

23-Jan

m_{s_1}

1	x		1
	x	x	
D		x	
1			1

$$m_{s_1} = \bar F\bar D$$

$$F = \bar C\bar E F + DEF + A\bar F\bar D$$

Example :

$$Z = A'B'CDF' + A'CD + A'B'CD'E + BCDF'$$

Sol : two variables are E & F

$$Z = A'B'CDF' + A'BCD + A'B'CD + A'B'CD'E + ABCDF' + \bar A BCDE'$$

$$= (0011)F' + 0111 + 0011 + (0010)E + F'(1111) + (0111)F'$$

$$= F'.m_3 + m_7 + m_3 + Em_2 + F'm_{15} + m_7F'$$

$$= m_3 + m_7 + Em_2 + F'm_{15}$$

		1	E
		1	
		F'	

$$m_{s_0} = \bar A C D$$

$$Em_{s_1} = E.(\bar A\bar B C)$$

$$F'm_{s_1} = \bar F(BCD)$$

$$Z = \bar A CD + \bar A\bar B CE + BCD\bar F$$

8 4 2 1

Ex: $W = \bar{A}B\bar{E}F + AEF + \bar{B}F\bar{E} + \bar{A}\bar{B}CEF + \bar{A}D\bar{F}\bar{E} + \bar{A}\bar{B}\bar{C}\bar{E}\bar{F}$

Sol: two variables are C & D

$= \bar{A}B\bar{E}F + ABE\bar{F} + A\bar{B}E\bar{F} + \bar{A}\bar{B}F\bar{E} + A\bar{B}F\bar{E} + \bar{A}\bar{B}EF.C + \bar{A}B\bar{E}\bar{F}.D + \bar{A}\bar{B}\bar{E}\bar{F}.D + \bar{A}\bar{B}\bar{C}\bar{E}\bar{F}$

$= 0101 + 1110 + 1010 + 0001 + 1001 + 0011.C + 0100.D + 0000.D + 0000.\bar{C}$

$= m_5 + m_{14} + m_{10} + m_1 + m_9 + m_3.C + m_4.D + m_0.D + m_0.\bar{C}$

$m_0(\bar{C}+D) \rightarrow z(\bar{C}+D)$

AB \ EF	$\bar{E}\bar{F}$	$\bar{E}F$	EF	$E\bar{F}$
$\bar{A}\bar{B}$	$(\bar{C}+D)$	1	C	
$\bar{A}B$	D	1		
AB				1
$A\bar{B}$		1		1

assume $\bar{C}+D = z$

$m_{s0} = ~~B\bar{E}F + AE\bar{F}~~ \; AE\bar{F} + \bar{A}\bar{E}F + \bar{B}\bar{E}F$

$D.m_{s1} = D.\bar{A}B\bar{E}$

$C m_{s2} = C.\bar{A}\bar{B}F$

$z\, m_{s3} = (\bar{C}+D)\bar{A}\bar{B}\bar{E}$

$W = AE\bar{F} + \bar{A}\bar{E}F + \bar{B}\bar{E}F + \bar{A}B\bar{E}D + \bar{A}\bar{B}FC + \bar{A}\bar{B}\bar{C}\bar{E} + \bar{A}\bar{B}\bar{E}D$

Combinational Circuit:

Ex: A Hacker does not want anyone to hack his data so he wants to design a secure he wants his signal to be transmitted in 16 diff line one by one. at Rx, he wants his data back as it was transmitted. He also wants to make a circuit that helps him in the selection of the channel out of avaible 16 channels on which he transmit data. This ckt will accept two bit binary no. and the channel which is Tx is the addition of this no., for ex: if two no. are 011 & 110 → addition will be 1001 then channel line will be Tx, the same ckt with the same I/P available at the Rx to get back original data back. As an ec engineer design the ckt in the LSI form ~~from~~ to solve hacker problem.

Sol:

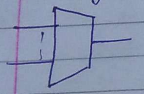
Example: Mr. X is the scientist and wants to design a Digital modulator. the i/p in the modulator comes from the digital carrier genr and modulate signal coming from the freq. genr. Modulating signal is totally independent of the carrier and signal is change irrespective of it as any analog signal. In the carrier genr six diffr o/p are available. So one can generatate six diffr freq. Those freq. are 50 Hz, 75 Hz, 100 Hz, 125 Hz, 150, 175 Hz. And adder ckt is also avaible with 3-bit o/p namely a_0, a_1, a_2. The modulator will ~~be~~ provided the carriers signal from the avaible 6 freq depending upon the ans of adder. More-over adder ckt is not simple adder. Mr. X wants o/p in binary form i/p to the adder ckt is the XS-3 form. An ec eng. solve the Mr. X's problem.

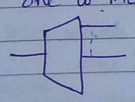
Date: 24-Jan

Encoder:
- interrupt priority encoder
- 2^n inputs, n outputs

Decoder:
- n inputs, 2^n O/Ps

Multiplexer	De-mux
many to one	one to many





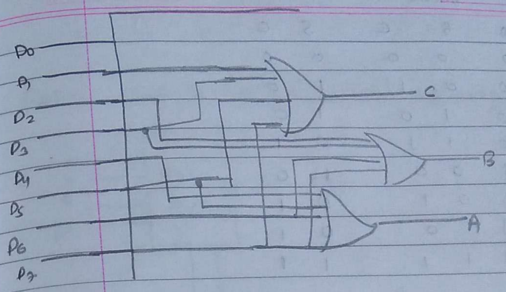
Octal to binary Encoder:

D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	A	B	C
1	0	0	0	0	0	0	0	0	0	0
	1							0	0	1
		1						0	1	0
			1					0	1	1
				1				1	0	0
					1	0	0	1	0	1
						1	0	1	1	0
							1	1	1	1

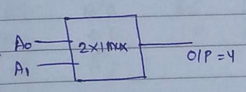
$A = D_4 + D_5 + D_6 + D_7$

$B = D_2 + D_3 + D_6 + D_7$

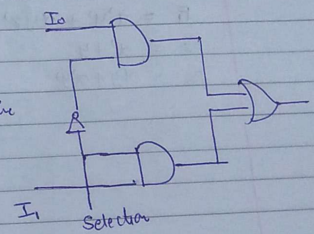
$C = D_1 + D_3 + D_5 + D_7$



: Design a 2×1 mux internal struc:



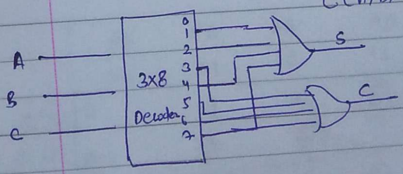
A_0	A_1	selection line
0	0	
0	1	
1	0	
1	1	



Design a full adder using 3 to 8 line decoder and OR gate:

$S(A,B,C) = \Sigma(1,2,4,7)$

$C(A,B,C) = \Sigma(3,5,6,7)$



Bubble and → NOR
Bubble OR → NAND

	A	B	C	S	C
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

Ex: implement this func using decoder and NAND gate

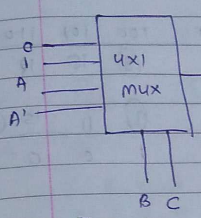
$F_1 = x'y' + xyz'$

$F_2 = x' + y$

$F_3 = xy + x'y'$

why in RTL view → all if else conditions are converted into mux (PP)

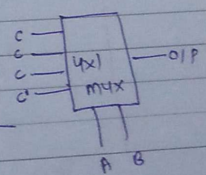
Ex:- $f(A,B,C) = \Sigma(1,3,5,6)$ by using 4x1 mux



		I_0 B'C' 00	I_1 B'C 01	I_2 BC' 10	I_3 BC 11
0	A'	0	1	2	3
1	A	4	5	6	7
I/P		0	1	A	A'

if two circles → 1
if not circles → 0
if one circle → irrespective variable

- AB → selection line

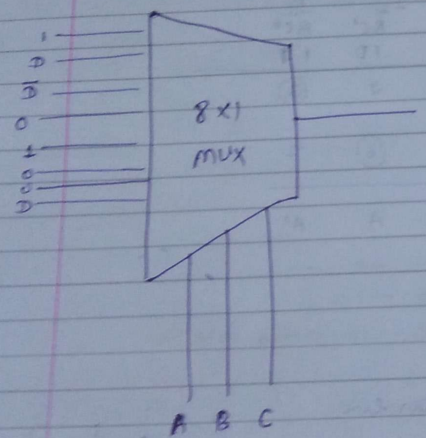


	AB' 00	A'B 01	AB' 10	AB 11
C'(0)	0	2	4	6
C (1)	1	3	5	7
	C	C	C	C'

Ex: $F(A,B,C,D) = \Sigma(0,1,3,4,8,9,15)$ using 8x1 mux

Sol: $2^3 \times 1$

$\bar{A}\bar{B}\bar{C}$	000	001	010	011	100	101	110	111
$\bar{D}$	0	2	4	6	8	10	12	14
D	1	3	5	7	9	11	13	15
	1	D	$\bar{D}$	0	1	0	0	D



using 2 (4x1 → mux)

	I_0	I_1	I_2	I_3
A'B'	0	1	2	3
A'B	4	5	6	7
AB'	8	9	10	11
AB	12	13	14	15

A B C D
0 0 0 0
0 0 0 1
0 0 1 0
0 0 1 1
0 1 0 0
0 1 0 1
0 1 1 0
0 1 1 1
1 0 0 0
1 0 0 1
1 0 1 0
1 0 1 1
1 1 0 0
1 1 0 1
1 1 1 0
1 1 1 1