

Customer's Need



Requirement



Specification

(formal or semi-formal)

mathematics \leftrightarrow HDL



Design Synthesis

(Mathematical Transformation)



Design Verification



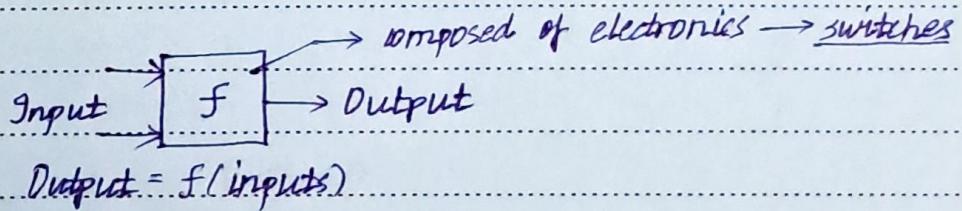
Fabricate



Test



Customer Delivery



Factors: (considered by design engineer)

• Cost

• Power consumption

• Security

• Performance

• Testability

• Embedded Intelligence

Propositions evaluate to true or false UNIVERSALLY (not ambiguous)

Truth value : T F
1 0

$P \rightarrow$ proposition.

P	0/F	1/T

e.g.

You can eat cake or ice cream.

OR

You can eat either cake or ice cream.

XOR

operators:

Negation : \neg , $\sim p$, \tilde{p} , $p!$, \bar{p}

NOT

Conjunction : \wedge , \cdot

AND

Disjunction : \vee , $+$

OR

Exclusive OR : \oplus

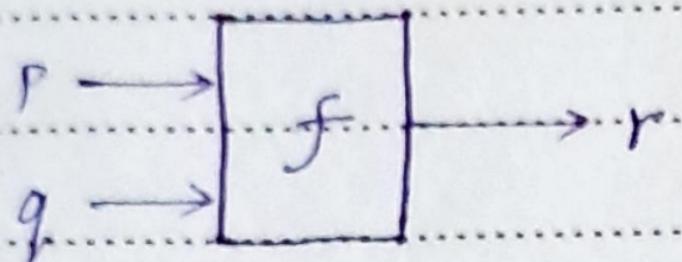
XOR

Implication : \rightarrow

Equivalence : \leftrightarrow

P	q	\wedge	\vee	\oplus	\rightarrow	\leftrightarrow
0	0	0	0	0	1	I DON'T CARE
0	1	0	1	1	1	I can be possible
1	0	0	1	1	0	0
1	1	1	1	0	1	1

$\sim, \cdot, + \rightarrow$ Basic operators \rightarrow all operations can be defined using these

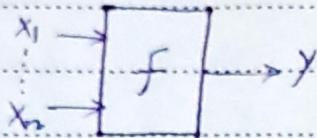


For n inputs, 2^{2^n} functions are possible.

$\hookrightarrow 2^n$ combinations $\rightarrow 2$ cases for each $\rightarrow 2^{2^n}$
 $(0, 1)$

$f \rightarrow$ represented as equation or truth table

EE 224



$$Y = f(x_1, \dots, x_n)$$

2^n possibilities each with 2 outcomes

2^{2^n} outcomes possible

e.g:

To start a car: $\rightarrow Y$

① Ignition system DN $\rightarrow p$
&

② Car in Parking $\rightarrow q$
OR

Break is applied (DN) $\rightarrow r$
&

③ Seat belt fastened $\rightarrow s$
OR

Car in Parking $\rightarrow q$

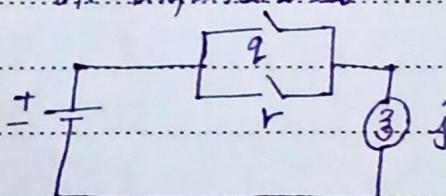
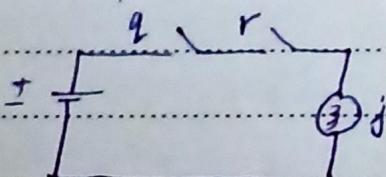
$$\therefore Y = p \cdot (q+r) \cdot (q+s) = f(p, q, r, s)$$

[NOT AND OR] XOR imply Equivalence

$\sim \& \oplus$ \leftrightarrow
BASIC

USING A

AND implementation: NETWORK OF SWITCHES DR implementation:



Number systems:

Decimal : 10-ary : 10 symbols \rightarrow digit

Binary : 2-ary : 2 symbols \rightarrow bit.

Say there are r symbols, then the system is defined as:

$$A_n A_{n-1} \cdots A_2 A_1 A_0 \cdot A_{-1} A_{-2} \cdots A_{-m}$$

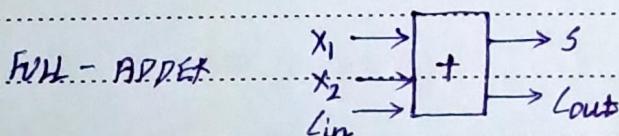
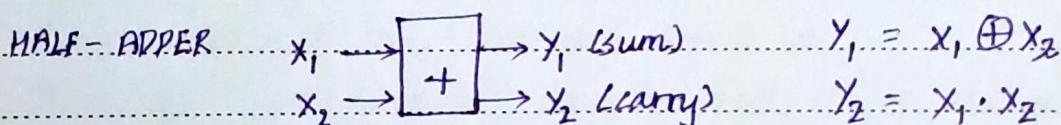
$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$

$$r^n r^{n-1} \cdots r^2 r^1 r^0 \cdot r^{-1} r^{-2} \cdots r^{-m}$$

Binary:

$$\begin{array}{r}
 0 \quad 0 \quad 1 \quad 1 \\
 + \quad 0 \quad +1 \quad +0 \quad +1 \\
 \hline
 00 \quad 01 \quad 01 \quad 10
 \end{array}$$

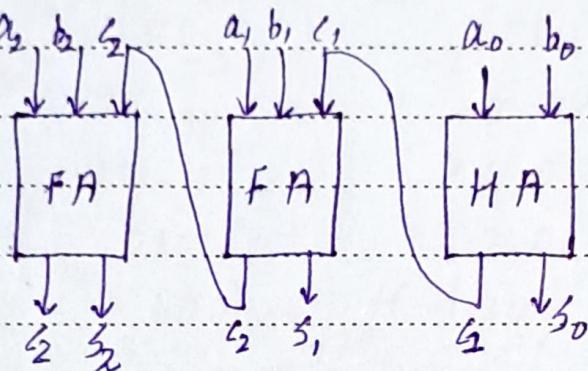
carry sum



X ₁	X ₂	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

EE 224

Addition:



Subtraction:

$$\begin{array}{r} 0 \quad 0 \quad 1 \quad 1 \\ - 0 \quad - 1 \quad - 0 \quad - 1 \\ \hline 00 \quad \overset{1}{\cancel{1}} \quad 00 \quad 00 \\ \text{borrow result} \end{array} \qquad \begin{array}{r} a \quad b \quad a-b \quad B \\ 0 \quad 0 \quad 0 \quad 0 \\ 0 \quad 1 \quad 1 \quad 1 \\ 1 \quad 0 \quad 1 \quad 0 \\ 1 \quad 1 \quad 0 \quad 0 \end{array}$$

Multiplication:

$$\begin{array}{r} 0 \quad 0 \quad 1 \quad 1 \\ \times 0 \quad \times 1 \quad \times 0 \quad \times 1 \\ \hline 0 \quad 0 \quad 0 \quad 1 \\ \end{array} \qquad \begin{array}{r} a \quad b \quad a*b \\ 0 \quad 0 \quad 0 \\ 0 \quad 1 \quad 0 \\ 1 \quad 0 \quad 0 \\ 1 \quad 1 \quad 1 \end{array}$$

For digital systems,

Fixed length

n -bits \rightarrow unsigned \rightarrow 0 to $2^n - 1$
signed $\rightarrow -2^{n-1}$ to $2^{n-1} - 1$

			D
15	1111	0000	0
14	1110	+0	1
13	1101	+1	0001
12	1100	+2	0010
	1011	+3	0011
11	1010	+4	0100
10	1001	+5	0101
9	1000	+6	0110
8		+7	0111
			1

0001

1110

→ 1's complement

0001

1110 + 1

1111 → 2's complement

$$\begin{array}{r}
 0010 \\
 + 0011 \\
 \hline
 0101
 \end{array}
 \quad
 \begin{array}{r}
 0011 \quad 3 \\
 - 0010 \quad -2 \\
 \hline
 0001 \quad 1
 \end{array}
 = 3 + (-2)$$

$$= 3 + 14 = 1$$

Signed Number Representation:

→ Exclusive sign bit

→ 1's complement (Decimal - 9's complement)

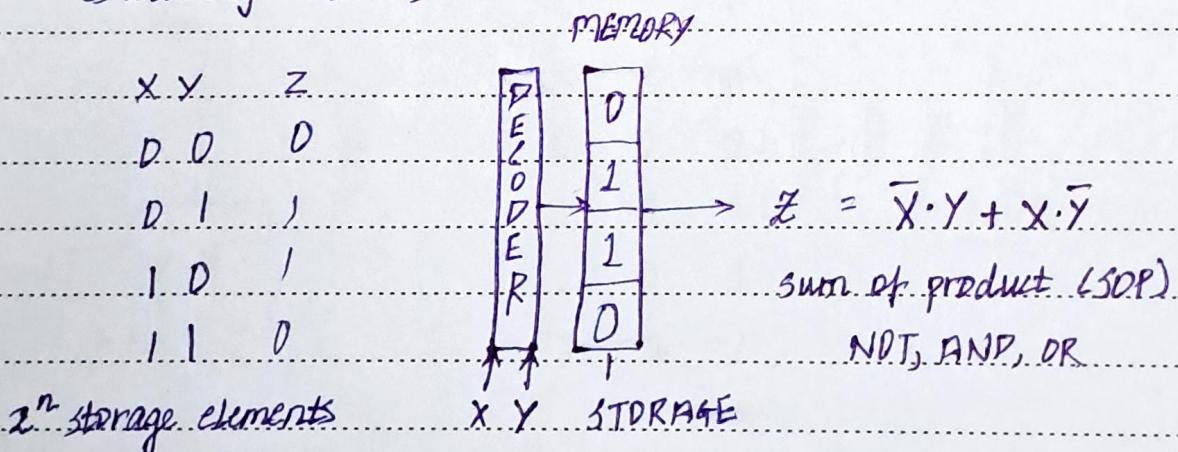
→ 2's complement (Decimal - 10's complement)

Representing signed numbers:

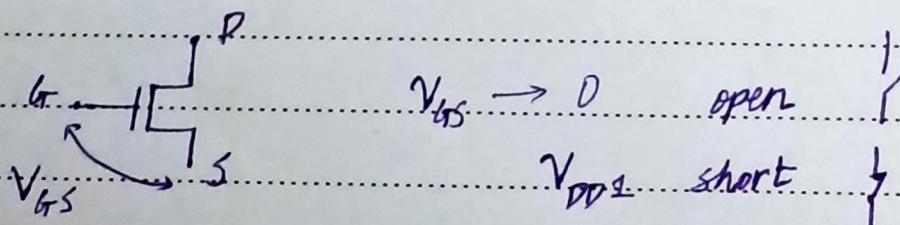
- sign bit
- 1's complement
- 2's complement

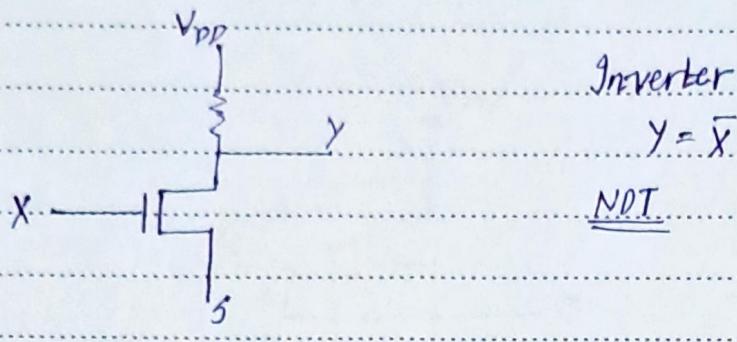
$$\begin{array}{l} \text{MSB} \rightarrow 1 \rightarrow -\text{ve} \\ \quad 0 \rightarrow +\text{ve} \end{array}$$

Representation
 (Truth Table) \rightarrow LOGIC
 Synthesis ARITHMETIC
 (Switching Network) OPERATION

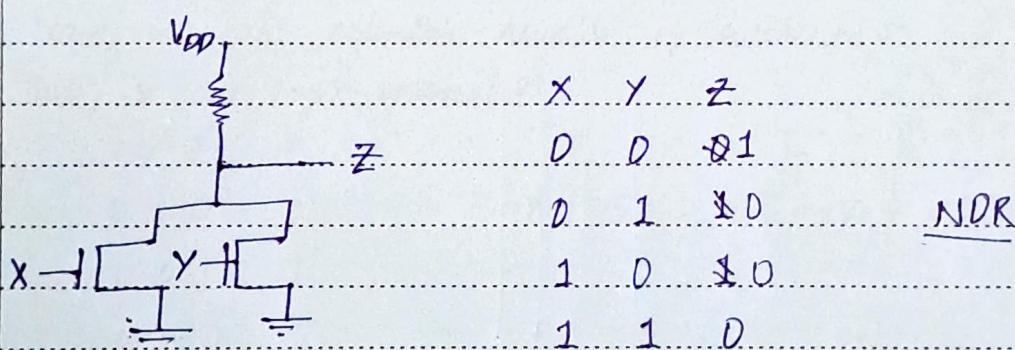
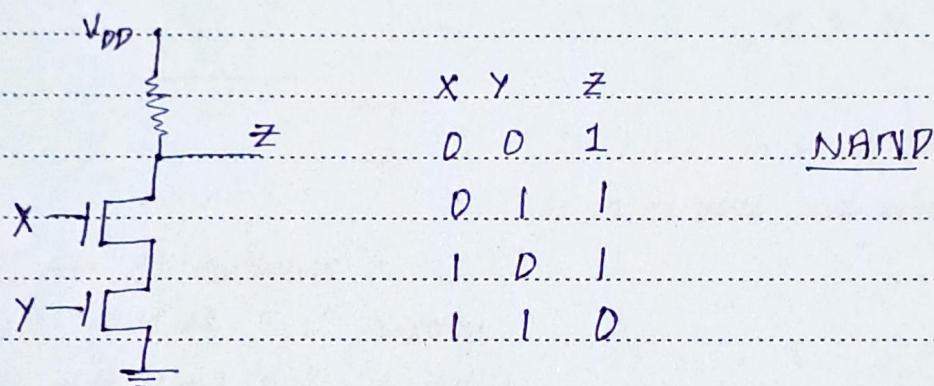


Throughout the course, we consider X, Y, \bar{X} & \bar{Y} are available to us. can be implemented using normally open, normally closed & electromechanical switches.

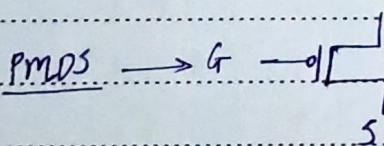




X	Y
0	1
1	0 (short)

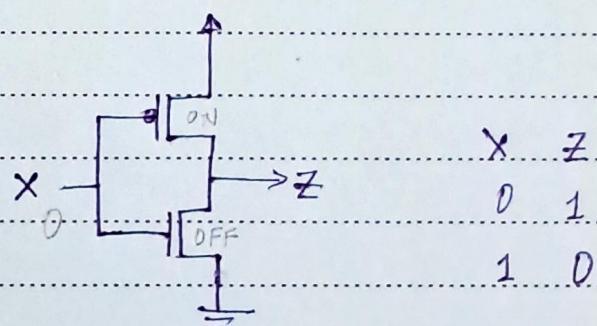


NMOS switches



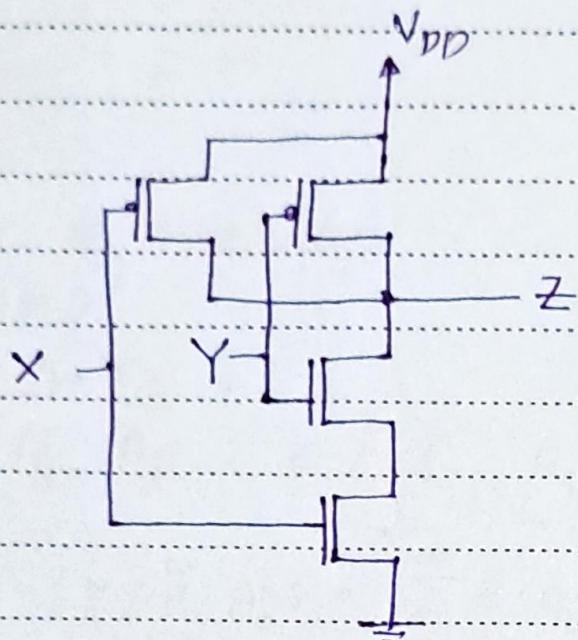
0 → Closed

1 → Open



NMOS

Inverter



NAND

XY Z

DD 1

D 1 1

1 D 1

1 1 D

If NMOS in series, PMOS should
be in parallel and vice-versa.

NDT → 2 switches

AND → $4+2 = 6$ switches

OR → $4+2 = 6$ switches

Now, we can calculate number of switches.

But, is our logic optimal?

XYZ P

0 0 0 0

$$P = \bar{X}\bar{Y}Z + XY\bar{Z} + X\bar{Y}Z + X\bar{Y}\bar{Z} + XYZ$$

0 0 1 1

0 1 0 0

$$P = X + Y\bar{Z} \rightarrow \text{optimal}$$

0 1 1 0

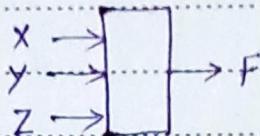
1 0 0 1

1 0 1 1

1 1 0 1

1 1 1 1

X	Y	Z	F
0	0	0	0
0	0	1	1



D	I	R	D
0	1	1	0

$$F = \bar{X}\bar{Y}Z + X\bar{Y}\bar{Z} + X\bar{Y}Z + XY\bar{Z} + XYZ$$

I	D	O	I
1	0	0	1

$$F = X + \bar{Y}Z$$

I	D	I	I
1	1	0	1

→ Cost (# switches)

I	I	I	I
1	1	1	1

→ Performance (Decay)

→ Power

→ Testability

→ Security

→ Intelligence

ALGEBRA:

Elements : $S = \{a, b, c, d\}$

Operators : *

Postulates :

Commutativity : $a * b = b * a$

Associativity : $a * (b * c) = (a * b) * c$

Identity : $a * e = e * a = a$

Inverse : $a * b = e$

* , + → Distributivity : $a * (b + c) = a * b + a * c \quad \forall a, b, c \in S$

George Boole - Boolean Algebra

$B = \{0, 1\}$ Operator: + ; ; ~ → ∨, ∧, ~

Closure : $L = a + b \quad a, b \in B \rightarrow L \in B$

Commutative : $a + b = b + a ; a \cdot b = b \cdot a$

Associative : $a + (b + c) = (a + b) + c ; a \cdot (b \cdot c) = (a \cdot b) \cdot c$

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$

Distributivity : $a \cdot (b + c) = a \cdot b + a \cdot c$ $a + (b \cdot c) = (a + b) \cdot (a + c)$

Identity : $a + 0 = a$ $a \cdot 1 = a$

Complementation : $a + \bar{a} = 1$ $a \cdot \bar{a} = 0$

Theorem:

1] Idempotency (Invariance):

$$a + a = a$$

$$a \cdot a = a$$

2] Null existence:

$$a + 1 = 1$$

$$a \cdot 0 = 0$$

Absorption

c] Involution:

$$\bar{\bar{a}} = a$$

d] Absorption:

$$a + ab = a$$

$$e] a + \bar{a} \cdot b = a + b$$

$$a \cdot (\bar{a} + b) = ab$$

$$a(a+b) = a$$

f] Uniting:

$$a \cdot b + \bar{a} \cdot b = b$$

$$(a+b) \cdot (\bar{a}+b) = b$$

g] DeMorgan's Theorem

$$\overline{a+b} = \bar{a} \cdot \bar{b}$$

$$\overline{a \cdot b} = \bar{a} + \bar{b}$$

$$\{, +, \cdot, \sim$$

h] Consensus Theorem:

$$a \cdot b + \bar{a} \cdot c + b \cdot c = a \cdot b + \bar{a} \cdot c$$

$$(a+b) \cdot (\bar{a}+c) \cdot (b+c) = (a+b) \cdot (\bar{a}+c)$$

Representation

1. Truth table \leftarrow Canonical
2. Function as SOP \rightarrow Sum of Product

Space Requirement - 2^n bits

Full adder:

X	Y	Z	S	F
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	0
1	1	0	1	1
1	1	1	1	1

$$F = \bar{X}\bar{Y}Z + X\bar{Y}\bar{Z} + X\bar{Y}Z + XY\bar{Z} + XYZ$$

$$= X + \bar{Y}Z$$

Cost:

6 (NOT)

$+ 5 \times 8$ (AND)

$+ 12$ (OR)

11

58

Too high

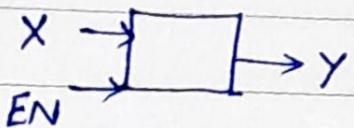
N input OR $\rightarrow 2N+2$ switches

N input AND $\rightarrow 2N+2$ switches

NOT $\rightarrow 2$ switches

Common Functions:

1] Enabler:



If $EN = 1$, then $Y = X$

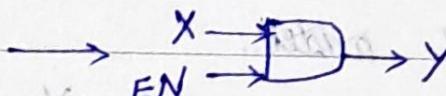
If: $EN \quad X \quad Y$

0 D D

D 1 D

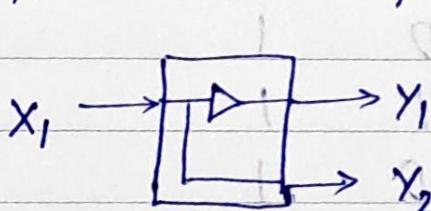
1 D D

1 1 0

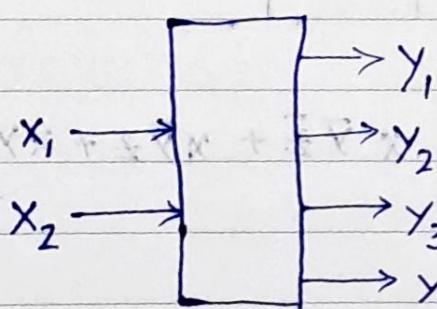


2] Decoder:

n inputs $\rightarrow 2^n$ outputs

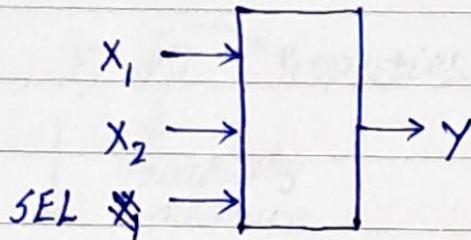


X_1	Y_1	Y_2
0	1	0
1	0	1



X_1	X_2	Y_1	Y_2	Y_3	Y_4
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0

3] Selector:



if $\bar{SEL} == 1$

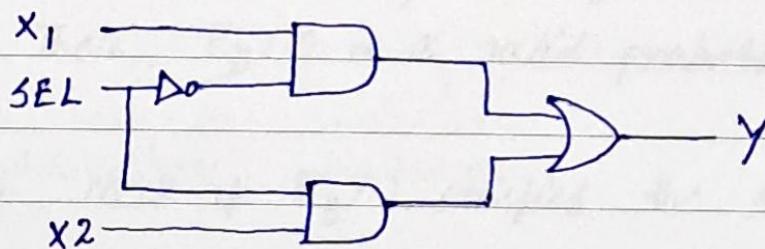
$$Y = X_2$$

else

$$Y = X_1$$



$$Y = \bar{SEL} \cdot X_1 + SEL \cdot X_2$$



For 4 inputs, 2 selectors 31, 30

4] Encoder:

2^n inputs \longrightarrow N outputs

(opposite of decoder)