

control

on/off

1/0

Number Systems & Codes:

Decimal - $b=10$ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

$$145.36_{10} = 1 \times 10^2 + 4 \times 10^1 + 5 \times 10^0 + 3 \times 10^{-1} + 6 \times 10^{-2}$$

$$N = \frac{a_0 + a_1 b_1 + a_2 b_1^2 + \dots + a_{p-1} b_1^{p-1}}{b_1^p}$$
$$N = \sum_{i=0}^{p-1} a_i \times b_1^i + \sum_{i=-1}^{-p} a_i \times b_1^i$$

$b=16$ — 0, 1, 2, ..., 9, A, B, C, D, E, F

$$342.5_8 = 3 \times 8^2 + 4 \times 8^1 + 2 \times 8^0 + 5 \times \frac{1}{8}$$
$$= 226.625_{10}$$

$$N = a_{q-1} b_2^{q-1} + a_{q-2} b_2^{q-2} + \dots + a_0 b_2^0$$
$$\frac{N_{b_1}}{b_2} = a_{q-1} b_2^{q-2} + a_{q-2} b_2^{q-3} + \dots + a_1 + \frac{a_0}{b_2}$$
$$\therefore N_{b_1} = b_2 \left(a_{q-1} b_2^{q-2} + a_{q-2} b_2^{q-3} + \dots + a_1 \right) + a_0$$

$$\begin{array}{r} 226 \\ 2^8 - 2 \uparrow \\ 3 - 4 \uparrow \\ 0 - 3 \end{array}$$

$$\begin{array}{r} 10101101_2 \\ \downarrow \\ 2535_8 \\ 550_{10} \end{array}$$

BCD				2's Complement			
decimal	8 4 2 1	2 4 2 1	2 4 2 1	2 4 3 -2	2 -3 2 1	2 -3 2 1	
0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 1 0 1	1 1 1 0	
1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1	1 1 1 1	0 0 0 0	
2	0 0 1 0	0 0 1 0	0 0 1 0	0 1 0 1	0 1 0 1	1 1 1 0	
3	0 0 1 1	0 0 1 1	0 1 0 0	0 1 0 1	0 1 0 1	1 1 1 0	
4	0 1 0 0	0 1 0 0	0 1 0 1 1 0 1 1	0 1 0 1 1 0 1 1	0 1 0 1 1 0 1 1	1 1 1 0	
5	0 1 0 1	0 1 0 1	0 1 1 0	0 1 1 0	0 1 1 0	1 1 1 0	
6	0 1 1 0	0 1 1 0	0 1 1 1	0 1 1 1	0 1 1 1	1 1 1 0	
7	0 1 1 1	0 1 1 1	1 1 1 0	1 1 1 0	1 1 1 0	1 1 1 0	
8	1 0 0 0	1 0 0 0	1 1 1 0	1 1 1 0	1 1 1 0	1 1 1 0	
9	1 0 0 1	1 0 0 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	

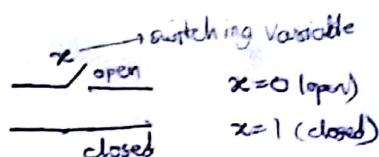
Non-weighted code

gray code

0	0000	0
1	0001	1
2	0011	3
3	0010	2
4	0110	6
5	0111	7
6	0101	5
7	0100	4
8	1100	8
9	1101	5

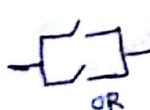
10111010

Switching Algebra :-



set $\{0, 1\}$ operation $+, \cdot, ', \bar{x}$
 OR AND NOT

Idempotency



$$x+x=x$$

$$x \cdot x = x$$

$$1+1=1$$

$$0+0=0$$

$$1 \cdot 1=1$$

$$0 \cdot 0=0$$

AND

$$x+1=1$$

$$x \cdot 0=0$$

$$x+0=x$$

$$x \cdot 1=x$$

Perfect Induction :-

$$x=1 \text{ iff } x \neq 0$$

$$x=0 \text{ iff } x \neq 1$$

$$x + x \cdot y = x \cdot (1+y) = x + x$$

absorption: $x + x \cdot y = x$

commutative: $x + 1 = 1 + x$ $x \cdot 0 = 0 \cdot x$

associative: $x + (y + z) = (x + y) + z$

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z$$

Distributive:

$$+, \cdot$$

$$x \cdot (y + z) = x \cdot y + x \cdot z$$

$$x + y \cdot z = (x + y) \cdot (x + z)$$

$$x \cdot (x + y) = x$$

Perfect Induction prove with all possible values

x	y	z	$x+y$	$x+z$	$(x+y) \cdot (x+z)$	$x \cdot y \cdot z$
0	0	0	0	0	0	0
1	0	0	1	0	0	0
2	0	1	1	0	0	0
3	0	1	1	1	1	1
4	1	0	1	1	1	1
5	1	0	1	1	1	1
6	1	1	1	1	1	1
7	1	1	1	1	1	1

Consensus:

$$\textcircled{1} \quad xy + x'z + yz = xy + x'z$$

$$\textcircled{2} \quad (x+y) \cdot (x'+z) \cdot (y+z) = (x+y) \cdot (x'+z)$$

$$\begin{aligned}
 & (x+y) \cdot (x'+z) \cdot (y+z+x+x') \\
 &= (x+y) \cdot (x'+z) \cdot (x+z) + (x+y) \cdot (x'+z) \cdot (x+y) \\
 &= (x+y) \cdot (x'+z) \quad \left(\begin{array}{l} \text{xy+x'z+yz} \\ \text{= xy+x'z+yz(x+x')} \\ \text{= xy(1+z) + x'z(1+y)} \\ \text{= xy + x'z} \end{array} \right)
 \end{aligned}$$

Dual form $\rightarrow +, \cdot$ exchanged

Three Basic Rules

involution $(x^l)^l = x$

De Morgan's: $(x+y)^l = x^l \cdot y^l$
 $(x \cdot y)^l = x^l + y^l$

Can. Canonical Form:

A (x_1, x_2, \dots, x_n)
 Switching expression
 $f(x_1, x_2, \dots, x_n)$
 Switching function
 of n variables

Combination circuits
 present output depends only
 on present input.

$$f(x, y, z) = xy + y^l z$$

x	y	z	$xy + y^l z$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$f_1 = x^l y^l z + x^l y^l z + x^l y^l z + x^l y^l z$$

↑ SOP
 This is canonical form

Canonical Form of Switching Functions:-

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

$$f(x, y, z) = xy + y^l z + xz^l$$

$$= \underline{xyz} + \underline{xyz^l} + \underline{y^lzx} + \underline{y^lzx^l} + \underline{x^lyz^l} + \underline{x^ly^l z}$$

$$f_{SOP} = xyz + xyz^l + xy^l z + x^ly^l z + x^ly^l z$$

Sum of Product form (SOP)

let f_1 & f_2 be two canonical forms of same function f .
 ⇒ if $f_1 = 1$ then $f_2 = 0$

product term is called the minterm

$$f_{POS} = (x^l + y^l + z^l) \cdot (x^l + y^l + z) \cdot (x + y^l + z^l)$$

Product of sum terms is called the maxterm

How to get the canonical form:

$$f(x, y, z) = xy + y'z \\ = xyz + xyz' + xy'z + xy'z'$$

- ① Check each product term for all the variables
- ② Introduce the missing variable in the term as ($x \cdot x = x$)
- ③ Check the redundancy ($x + x = x$)

$$\begin{aligned} f(x, y, z) &= xy' + xz \\ &= x \cdot (y' + z) \\ &\cancel{=} x \cdot (x + y' + z) \\ &= x \cdot (y' + z) + x \cdot x \\ &\cancel{=} x \cdot (x' + y' + z) \\ &= (x + y + z) \cdot (xx' + y' + z) \\ &= ((x + y)(x + y') \cancel{+ xz}) - (x + y' + z)(x' + y' + z) \\ &= (x + y + z) \cdot (x + y + z') \cdot (x + y' + z) \cdot (x' + y' + z) \\ &= (x + y + z) \cdot (x + y + z') \cdot (x + y + z) \cdot (x + y' + z) \end{aligned}$$

Shanon's Expansion Theorem :-

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &= x_1 \cdot f(1, x_2, \dots, x_n) + x_1' \cdot f(0, x_2, \dots, x_n) \\ &= [x_1 + f(0, x_2, \dots, x_n)] \cdot [x_1' + f(1, x_2, \dots, x_n)] \end{aligned}$$

Proof by perfect induction:

Consider expansion against one variable (say x_1)

$$\text{when } x_1 = 1, \quad f(x_1, x_2, \dots, x_n) = f(1, x_2, \dots, x_n)$$

$$\text{if } x_1 = 0 \quad f(x_1, x_2, \dots, x_n) = f(0, x_2, \dots, x_n)$$

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &= x_1 x_2 f(1, 1, x_3, \dots, x_n) + x_1 x_2' f(1, 0, x_3, \dots, x_n) \\ &\quad + x_1' x_2 f(0, 1, x_3, \dots, x_n) + x_1' x_2' f(0, 0, x_3, \dots, x_n) \end{aligned}$$

$$f(x_1, x_2, \dots, x_n) \quad \text{No. of switching functions} = 2^n$$

$$f(x,y) = \alpha xy + \alpha xy' + \alpha x'y + \alpha x'y'$$

value

$$A \otimes B = A \oplus A$$

$$A \otimes (B \otimes C) = (A \otimes B) \otimes C$$

$$A \otimes A \otimes C = A \cdot (B \otimes C)$$

$$A \otimes B \cdot C$$

$$A \otimes C = S$$

$$S \otimes C = A$$

$$A \otimes B \otimes C = 0$$

Functionally Complete:

, +, *, '

Duality of SW algebra {OR, NOT}

{AND, NOT}

$$x \text{ NOR } y = (x+y)'$$

$$x \text{ NOR } x = (x+x)' = x' \text{ NOT}$$

$$(x+y) \text{ NOR } (x+y)' = (x+y) \text{ OR}$$

Isomorphic system

- ① Series-Parallel switching circuits
- ② Switching algebra are isomorphic



- ③ Propositional logic is isomorphic to SW algebra

Syst sets for \neg and $\neg\neg$

$(\alpha_1, \alpha_2, \dots, \alpha_n)$

Syst set B $\{b_1, b_2, \dots, b_n\}$

$(\alpha_1 \alpha_2 \dots \alpha_n)$

Syst \rightarrow Syst

Boolean algebra is a distributive, complemented lattice

$$\begin{array}{l} \text{Set } A \{a, b, c = 0, 1\} \\ \text{op}^n \{+, \cdot\} \end{array}$$

idempotent, commutative,
associative, absorption & mutually distributive

Def'n of complementation:

$$\begin{array}{l} a+a' = 1 \\ a \cdot a' = 0 \end{array}$$

↑
↑
a+b
a+b'

Let 'a' has two complements

$$a+b_1=1 \quad a \cdot b_1=0$$

$$a+b_2=1 \quad a \cdot b_2=0$$

$$b_2 = b_2 \cdot 1 = b_2 \cdot (a+b_1)$$

$$= b_2 \cdot a + b_2 \cdot b_1 = b_2 \cdot b_1 \cancel{+} b_2 \cdot b_2$$

$$\cancel{= b_2 \cdot a + b_1 \cdot b_2}$$

$$= 0 + b_2 \cdot b_1 \neq$$

$$= a \cdot b_1 + b_2 \cdot b_1$$

$$= (a+b_2) \cdot b_1$$

$$= b_1$$

introduce two elements 0 & 1 as bounds $a \cdot b = 1$
 $a + b = 1$

DeMorgan's Theorem:

$$(a+b)' = a' \cdot b'$$

$$(a \cdot b)' = a' + b'$$

$$(a+b)' \cdot (a' \cdot b') = a \cdot a' b' + b \cdot a' b' = 0$$

	0	1	a	b
+	0	1	a	b
0	0	1	a	b
1	1	1	1	1
a	a	1	a	$a+b \rightarrow 1$
b	b	1	$a+b \downarrow$	b

	0	1	a	b
0	0	0	0	0
1	0	1	a	b
a	0	a	a	$a \cdot b \rightarrow 0$
b	0	b	$a \cdot b$	b

1.1 Combination Logic

AND
OR
NOT
NAND
NOR

output depends only on present input

$$f = xyz + xy'z + x'y'z + x'y'z' + xyz'$$

$$f' = x'y'z' + x'y'z + x'yz$$

$$\begin{aligned} \text{Minimize } f &= A + A'B + A'B'C + A'B'C'D + \dots \\ &= A + B + C + D + \dots \end{aligned}$$

Check the equivalence of

$$f_1 = AB'C + A'C'D + AC + ABC$$

$$f_2 = B'CD + BD + AC = AB'CD + A'B'CD + ABCD + A'BCD + A'BD + A'CD$$

$$AB'C + A'C'D + ABC = \cancel{AC} + \cancel{A'C'D}$$

$$= A'B'C + A'C'D + AC$$

$$= A'B'CD + A'B'CD' + A'BC'D + A'BC'D' + AC$$

max 3.6

Chapter 3, Kohavi Book

3.3 (c) Simplify
 \checkmark abc, d, e

$$xy + wxyz' + x'y$$

$$= xy(1+wz') + x'y = xy + x'y = y$$

$$\begin{aligned} (d) \quad & a + a'b + ab'c + a'b'c'd + \dots \\ & = a + a'b + b'c + b'c'd + \dots \\ & = a + b + b'(c + b'c'd + \dots) \\ & = a + b + c + d + \dots \end{aligned}$$

3.4

$$f = w \\ f' =$$

3.5 (c) With

wz

3.7

A, B

3.12 By a

A

=

=

=

=

=

$$3.4 \quad f = w^1 + (x^1+y^1+z^1) (x^1+y^1+z^1)$$

$$f' = w \cdot (x^1z^1 + x^1y^1 + x^1z^1)$$

Without perfect induction, check if the eq is valid

3.5 (a)

$$xyz + wy^1z^1 + wxz^1 = xy^1z + wy^1z^1 + wxyz \quad (\text{True})$$

$$\begin{aligned} wxyz + w^1xyz^1 + wxyz^1 + wxy^1z^1 &= wxyz + w^1xyz^1 + wxyz^1 + wxy^1z^1 \\ &+ wxyz^1 \end{aligned}$$

3.7 A, B, C, D

$$A^1 + AB = 0$$

$$AB = AC$$

$$AB + AC^1 + CD = C^1D$$

$$A^1 + AB = 0 \Rightarrow A^1 + B = 0$$

$$\Rightarrow \cancel{A} \cancel{B} \quad A^1 = 0, B = 0$$

$$\Rightarrow A = 1, B = 0$$

$$AC = AB \Rightarrow C = 0$$

$$D = 0 + 1 + 0 = 1$$

$$A = 1, B = 0, C = 0, D = 1$$

3.12 By adding redundant terms, convert SOP to POS

$$AC + A^1BD^1 + A^1BE + A^1C^1DE$$

$$= AC + \cancel{A(BD^1 + BE)} + A^1(BD^1 + BE + C^1DE)$$

$$= (AC + A^1)(AC + BD^1 + BE + C^1DE)$$

$$= (A + C) \cdot (AC + C^1DE + B \cdot (D^1 + E))$$

$$= (A + C) \cdot (AC + C^1DE + B) \cdot (AC + C^1DE + D^1 + E)$$

$$= (A + C) \cdot (AC + C^1DE + B) \cdot (AC + D^1 + E)$$

$$= (A + C) \cdot (A + D^1 + E) \cdot (C + D^1 + E) \cdot (AC + C^1DE + B)$$

$$\begin{aligned}
 & uvw + uwxz + uwxz + wxyz \\
 &= uv(w+xz) + xy(uw+zx) \\
 &= uv(uw+xz) + xy(uw+xz) \\
 &= (uw+xy)(uw+xz) \\
 &= (uw+xy)(uw+xz)(v+y)(u+z)(w+x)(w+z)
 \end{aligned}$$

3.14. Show that $f(A, B, C) = A'BC + AB' + B'C'$ is a universal opⁿ

$$f(A, A, B) = A'B'$$

3.15. Prove or give counter example

$$(d) (A \oplus B)' = A' \oplus B = A \oplus B'$$

$$\begin{aligned}
 (e) A \oplus (B+C) &= A \oplus B + A \oplus C \\
 &= AB' + BA' + AC' + CA' \\
 &= A \cdot (B' + C') + (B + C) \cdot A' \\
 A=1, B=1, C=0
 \end{aligned}$$

3.20 A safe have 5 locks v, w, x, y, z all of which must be unlocked for the safe to open. The keys to the locks are distributed among the executives like

A	has keys for locks	$v \& x$	$w \& y \& z$	A	✓ \bullet ✓ →
B	"	$v \& y$		B	✓
C	"	$w \& y$		C	✓
D	"	$x \& z$		D	✓
E	"	$v \& z$		E	✓

~~crossed out~~

C is essential

B is not required

min. no. of executives?

Minimization

Reduces the

Karnaugh

- pict
- it

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

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

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

∴ Answer

$$= ACD + ACE + BCD + BCE$$

Minimization of switching Function:

Reduces the no. of literals
Methods

- Boolean
- Pseudo
- Karnaugh
- Quine

$$\begin{aligned} f &= \alpha^1 \bar{b} \bar{c} + \alpha^1 b \bar{c} + \alpha^1 b c \\ &\Rightarrow \alpha^1 \bar{b} + \alpha^1 b \\ &\Rightarrow \alpha^1 = 0 \end{aligned}$$

(no. of terms = 2)
 $\alpha^1, \beta, \gamma, \dots$ - literals
minimisation of
① no. of literals
② no. of terms
③ no. of variables in one product term

adjacent blocks vary
only in one literal

Karnaugh map:-

- pictorial representation of truth table minimisation
- it applies $\alpha^1=1$ (SOP) & $\alpha^1=0$ (POS) to minimise the function

$$f = xy^1 z + x^1 y^1 z + x^1 y^1 z + x^1 y^1 z$$

$$\begin{array}{|c|c|c|c|} \hline x \backslash y & 00 & 01 & 11 & 10 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 1 & 1 \\ \hline \end{array}$$

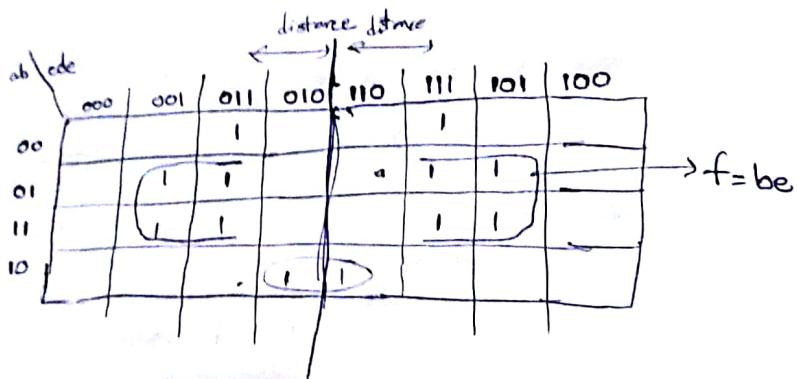
$$f(x,y,z) = x^1 y^1 z + x^1 y^1 z + x^1 y^1 z + x^1 y^1 z$$

$$= x^1 y^1 z + x^1 y^1 z$$

abcd	00	01	11	10
00				
01				
11				
10				

for n-variable function,
 2^n minterms

Try to find a cube of
 2^m cells, then the expression
reduces to $(n-m)$ for n variables



K-map works efficiently upto 6-variable

$$f = \sum(0, 4, 5, 7, 8, 9, 13, 15)$$

ab\cd'e'	00	01	11	10
00	1			
01	1	1		
11	1		1	
10	1		1	1

$$f_1 = ac'd' + bd + ab'c'$$

$$f_2 = b'c'd' + abc' + bd + ac'd$$

Each one is
covered only by
one cube
to minimize

Code converter

BCD to excess-3-code

Binary
coded decimal
digits

f₃

f₃ =

BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001
10-15										

12-18

BCD to

a	b	c	d
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	1	0	0
6	1	0	0
7	1	0	0
8	1	0	0
9	1	0	0

function
forms

	Excess-3 code
0000	0000
0001	0100
0010	0101
0011	0110
0100	0111
0101	1000
0110	1001
0111	1010
1000	1011
1001	1100
1010	1101
1011	1110
1100	1111

10-15 don't care

variables

17.1.16

ab\cd	00	01	11	10
00	0	0	0	0
01	1	0	1	0
11	0	0	0	0
10	1	0	1	0

$$f_{pos} = (a+b)(\bar{a}+\bar{b})(\bar{b}+d)(\bar{b}\bar{c}+d)$$

BCD to Excess-3-code converter

a b c d	f ₃	f ₂	f ₁	f ₀
0 0 0 0	0	0	0	1
0 0 0 1	0	1	0	0
0 0 1 0	0	1	0	1
0 0 1 1	0	1	1	0
0 1 0 0	0	1	1	1
0 1 0 1	1	0	0	0
0 1 1 0	1	0	0	1
0 1 1 1	1	0	1	0
1 0 0 0	1	0	1	1
1 0 0 1	1	1	0	0

$$\begin{aligned} f_0 &= \bar{d} \\ f_1 &= cd + \bar{c}\bar{d} \\ f_2 &= b\bar{c}\bar{d} + \bar{b}\bar{d} + \bar{b}c \\ f_3 &= a + b\bar{c} + b\bar{d} \end{aligned}$$

Multi-output circuit

ab\cd	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	0	0	0	0
10	1	0	0	0

$f_0 = \bar{d}$

ab\cd	00	01	11	10
00	1	0	1	0
01	1	0	0	0
11	0	0	0	0
10	1	0	0	0

$f_1 = \bar{c}\bar{d} + \bar{a}cd$

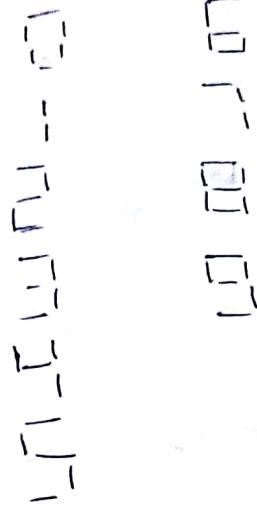
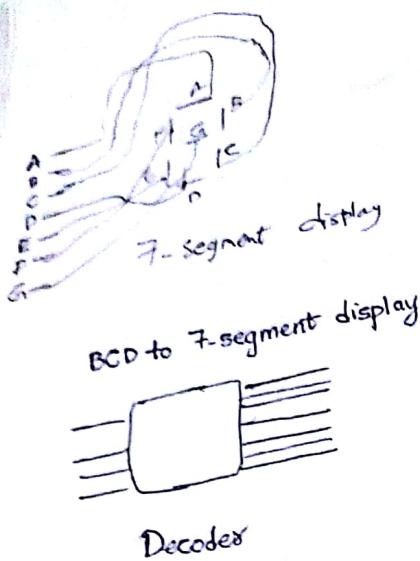
ab\cd	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	x	x	x	x
10	1	1	x	x

$$f_3 = ac + bc + bd$$

ab\cd	00	01	11	10
00	0	1	1	1
01	1	0	0	0
11	0	0	0	0
10	0	1	1	0

$$f_2 = b\bar{c}\bar{d} + \bar{b}\bar{d} + \bar{b}c$$

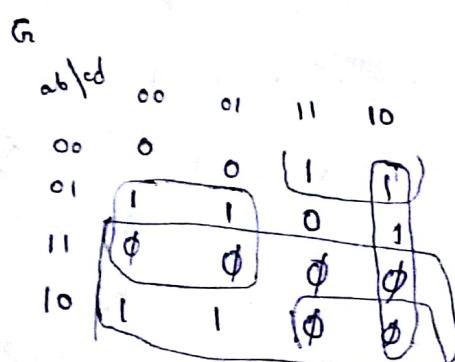
K-map
can handle



- ① Make minimi
- ② Make Make Get
- ③ Ref
- ④ Tote

BCD	A	B	C	D	E	F	G
0 0000	1	1	1	1	1	1	0
0 0001	0	1	1	0	0	0	0
1 0010	1	1	0	1	1	0	1
3 0011	1	1	1	1	0	0	0
4 0100	0	1	1	0	0	1	1
5 0101	1	0	1	1	0	1	1
6 0110	1	0	1	1	1	1	1
7 0111	1	1	1	0	0	0	0
8 1000	1	1	1	1	1	1	1
9 1001	1	1	1	1	0	1	1

Step 1:



$$G = a + b\bar{c} + bc + \bar{c}d$$

F
E
D
C
B
A

Step 2:

Kmap can handle functions with more variables

wx ^{yz}		00	01	11	10
00	w	0	1	A	0
01	0	1	0	1	1
11	A	b	φ	B'	
10	0	B	φ	0	

variable - A

f depends on A

Makes -

- ① Make all the map-entered variables zero and get the minimised function
- ② Make one map-entered variable as 1 and others zero.
Make all the ones as φ.
- ③ Repeat step 2 for all map-entered variables complemented & uncomplemented
- ④ Take the sum of all 1, 2, 3

Step 1:

wx ^{yz}		00	01	11	10
00	0	1	0	0	
01	0	1	1	1	
11	0	φ	φ	0	
10	0	0	φ	0	

$f_1 = wx'y'z + w'xz' + w'xy$

Step 2:

wx ^{yz}		00	01	11	10
00	0	φ	1	0	
01	φ	0	φ	φ	
11	1	φ	φ	0	
10	0	0	φ	0	

$f_2 = yz + wx'y'$

f_A

wx ^{yz}		00	01	11	10
00	0	φ	0	0	
01	φ	0	φ	φ	
11	0	φ	φ	0	
10	0	1	φ	0	

$$f_3 = wz$$

wx ^{yz}		00	01	11	10
00	0	φ	0	0	
01	φ	0	0	φ	
11	0	φ	φ	1	
10	0	0	φ	0	

$f_4 = xy$

$$\begin{array}{ccccc}
 w & x & y & z & \\
 \text{m} & 0 & 0 & 0 & 1 \\
 \text{m} & 1 & 0 & 0 & 0 \\
 \text{m} & 0 & 1 & 0 & 0 \\
 \text{m} & 1 & 1 & 0 & 0 \\
 \text{m} & 0 & 0 & 1 & 0 \\
 \text{m} & 0 & 0 & 0 & 1 \\
 \text{m} & 1 & 0 & 1 & 0 \\
 \text{m} & 0 & 1 & 1 & 0 \\
 \text{m} & 1 & 1 & 1 & 0
 \end{array}
 \quad f_S = w'x'y'z' + w'x'z' + w'xy' + A(yz + wxy') + B\cdot wz + B'xy + C\cdot w'z'y$$

Step 9 - $f = w'x'y'z' + w'x'z' + w'xy'$
 $+ A(yz + wxy') + B\cdot wz + B'xy + C\cdot w'z'y$

$f(x,y,z)$, $g(x,y,z)$

f covers g if f is 1 whenever g is 1.

If f covers g and g covers f , then f and g are equivalent.

Implicant - If f covers one product term, then the product term called implicant and h is said to imply f .

$$f = w'x + y'z, h = wxy'$$

f covers h

$$h \rightarrow f$$

$h \rightarrow f$

Prime Implicant: Prime implicant is the implicant in which if one literal is deleted then the new term does not imply f .

Theorem: Irredundant function are sum of all prime implicants of f_{SOP}

Proof: Let irredundant function contain one implicant which is not prime.

If I delete one literal for the implicant, then also it implies f .

f_{SOP} is not irredundant

wxyz
0000
0100
1100
1000

$f =$

Quine

Step 1 - Make with

Step 2 - Try adjacent can |

Give a minute

Repeat
can be until

on/alone

$w \setminus x$	00	01	11	10
00	1	0	1	0
01	1	1	0	1
11	0	1	1	0
10	0	0	0	1

$$f = \underline{wx} + w'y'z + w'yz$$

($w \setminus x$)

essential

prime implicants: covers a '1' which is not covered by others

Quine Mc Clusky:

Step 1: Make groups of minterms

with same no. of 1's (index)

$$f = \sum (0, 1, 2, 5, 7, 8, 9, 10, 13, 15)$$

Step 2: Try to identify the pairs from adjacent groups so that they can be combined. ($A \oplus A' = A$)

Give a tick ~~A \oplus A'~~ if the minterm is combinable

Repeat until no two implicants can be combined. Take all the unticked implicants

$$f = \underline{x'y'} + \underline{z'z} + \underline{y'z} + \underline{xz}$$

Step 1		Step 2	
index	minterms	indices	check
0	0000✓	0, 1	000-✓
1	0001✓		00-✓
2	0010✓		-000✓
3	1000✓		
4	0101✓	1, 2	0-01-
5	0001✓		-001-
6	1010✓		-010-
7	0101✓	2, 3	0-01-
8	0001✓		-001-
9	1010✓		-010-
10	0111✓	3, 4	100-
11	1101✓		-101-
13	1111✓		-111-
15			11-1-

Step 3

indices	wxyz
0, 1, 2	-00-
	-0-0
	-0-
	f = $x'y' + z'z$
1, 2, 3	-0-
	-01
	-01
	+ $y'z + zx$
2, 3, 4	-1-
	-1-

R.M

- ① Binary Representation
- ② Decimal Representation

20/1/18

$$f = \sum(0, 1, 2, 5, 7, 8, 9, 10, 13, 15)$$

	w x y z
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
5	0 1 0 1
8	1 0 0 0
7	0 1 1 1
13	1 1 0 1
15	1 1 1 1

	w x y z
0, 1	0 0 0 -
0, 2	0 0 - 0
0, 8	- 0 0 0
1, 5	0 - 0 1
1, 9	- 0 0 1
2, 10	- 0 1 0
8, 9	1 0 0 -
8, 10	1 0 - 0
5, 7	0 1 - 1
9, 13	1 - 0 1
5, 13	0 - 1 0 1
7, 15	- 1 1 1
13, 15	1 1 - 1

	w x y z
0, 1, 8, 9	- 0 0 -
0, 2, 8, 10	- 0 - 0
1, 5, 9, 13	- - 0 1
5, 7, 13, 15	- 1 - 1

$$f = x'y' + x'z' + y'z + xz$$

Prime Implicant chart :-

	0	1	2	5	7	8	9	10	13	15
A	✓					✓				
B				⊕						
C										
D						⊕			⊕	

B, D essential

$$(A+C) \cdot (A+C)$$

Either A ⊕ C

$$f_1 = x'y' + x'z' + xz$$

$$f_2 = x'z + y'z + xz$$

$$\begin{aligned} 17 &\rightarrow 10001 \\ 13 &\rightarrow 1101 \end{aligned}$$

A
B
C
D
E
F
G
H

Quine Mc Class

$$f(v, w, x, y, z)$$

Step 1

index	minit
1	1
2	1
3	1
4	1
5	1

Durchsetzung
Quine-McCluskey Methode

$$f(v, w, x, y, z) = \sum (1, 2, 12, 13, 15, 19, 16, 17, 20, 21, 23, 24, 25, 26, 27, 28, 29, 31)$$

index	minterms	Step 1	Step 2	Step 3
1	1, 2		1, 17 (1, 2)	16 A — H
2	12, 17, 18, 20, 24	2, 12 (16)	16 B — G, 12	
3	13, 19, 21, 25	12, 13 17, 21	1 — F 17, 19, 23 4 — 19, 23	2, 14 /
4	15, 23, 27, 29	17, 25 20, 21 24, 25 18, 19	8 — 17, 25, 1 — D 17, 25, 1 — C 19, 27 1 — E — 18, 25, 27	17, 21, 19, 23, 25, 29, 27, 31 2, 8 / A
5	31	13, 15 13, 29 19, 23	2 — 13, 27, 18 — 21, 29 4 — 17, 25, 8 — 21, 29 21, 23 — 20 21, 29 — 8 25, 27 — 2 25, 29 — 4	2, 8 / 13, 27, 21, 29 — B
		15, 31 23, 31 27, 31 29, 31	16 — 19, 27 8 — 23, 31 4 — 25, 29 2 — 27, 31	4, 8 / (2, 16) — B
		v w x y z		
A	1 - - - 1	vz		
B	- 1 1 - 1	wxz		
C	1 00 00 -	vwxyz	21, 23, 29, 31	2, 8 /
D	1 0 1 0 -	vwxyz		
E	1 0 0 1 -	vwxyz		
F	0 1 1 0 -	vwxyz		
G	- 0 0 1 0	wxyz		
H	- 0 0 0 1	wxyz		

(A+C)
✓

Prime Implicant chart

	1	2	12	13	15	17	19	20	21	23	24	25	27	29	
A	✓					✓		✓	✓	✓	✓	✓	✓	✓	✓
B	✓					✓									
C	✓														
D	✓														
E															
F						✓	✓								
G															
H								✓							

$$f = A + B + C + D + F + G + H$$

$$f(v, w, x, y, z) = \sum (1, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14, 15, 18, 19, 20, 21, 22, 23, 25, 26, 27)$$

$$A(4, 5, 6, 7, 20, 21, 22, 23)$$

$$B(4, 5, 6, 7, 12, 13, 14, 15)$$

$$C(18, 19, 26, 27)$$

$$D(18, 19, 22, 23)$$

$$E(10, 11, 26, 27)$$

$$F(10, 11, 14, 15)$$

$$G(3, 11, 19, 27)$$

$$H(3, 7, 19, 23)$$

$$I(3, 7, 11, 15)$$

$$J(1, 3, 5, 7)$$

$$K(25, 27)$$

	10	11	12	19	2G
C			x	x	x
D			x	x	
E	x	x			x
F	x	x			
G		x		x	
H				x	
I		x			

sow i covers
sow j

dominating rows

to be identified

$$f_i = A + B + J + K + C + E$$

$$f_2 = A + B + J + K + C + F$$

$$f(w, x, y, z) = \sum (0, 1, 5, 7, 8, 10, 14, 15)$$

Variable	Value
A	0
B	1
C	5
D	7
E	8
F	10
G	14
H	15

no essential prime implicant

write
software
for Quine
McCluskey

12/18 Logic Design :-

Combinational
Sequential

Propagation delay: Time required from input to output

fanout & fanin

fanout: Maximum current can be drawn from the gate without affecting the output

fanin: Minimum current is required to drive a gate

fanout: the no. of outputs that it can drive

fanin: The no. of inputs to be fed without restriction ^{to} for a gate.

One gate/few gates \rightarrow Circuits with millions of gates

SSI \rightarrow small scale integration

1/2/4/8-10 basic gates

MSI \rightarrow Medium scale integration

100 gates

LSI \rightarrow Large "

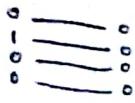
"

VLSI \rightarrow Very Large "

" few thousands of gates

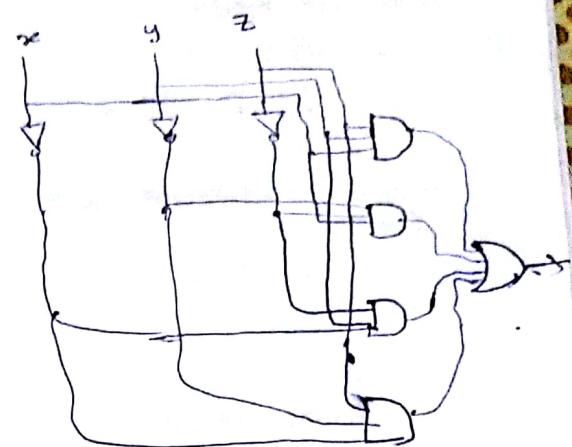
" Millions of gates

Parity Bit:



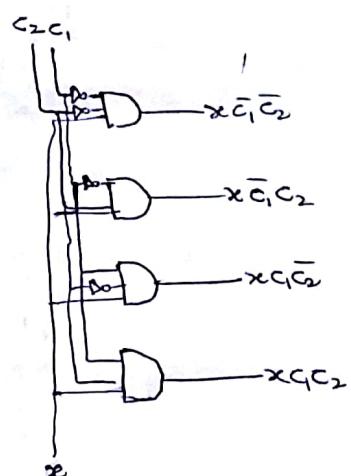
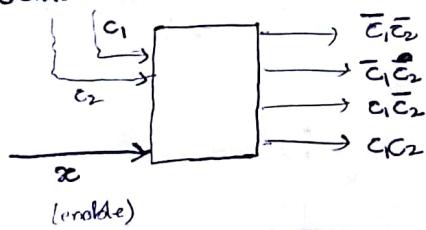
also send
odd/even
no. of 1's

x	y	z	Parity Bit (odd)
1	0	0	1
0	1	0	1
0	0	1	1
1	1	1	0



Serial to Parallel Converter:

Control lines



Comparators:

2-bit comparator:

$x_1 x_2 \quad y_1 y_2$

$x_1 x_2$	$y_1 y_2$	00	01	10	11
00	00	f_1	f_2	f_2	f_2
00	01	f_0	f_1	f_2	f_2
01	10	f_0	f_0	f_1	f_0
10	10	f_0	f_0	f_2	f_0

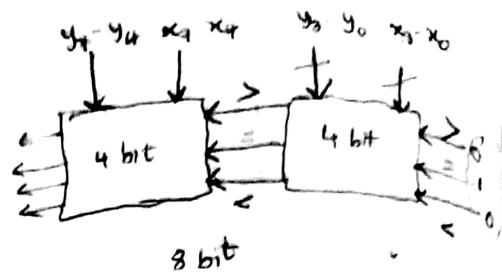
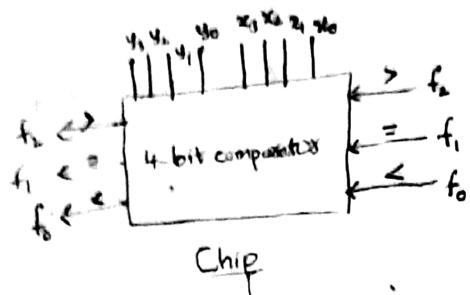
$$f_2 = x_1 y_1 + x_2 y_1 y_2 + x_1 x_2 y_2$$

$$f_1 = x_1 x_2 y_1 y_2 + x_1 x_2 y_1 y_2 + x_1 x_2 y_1 y_2 + x_1 x_2 y_1 y_2$$

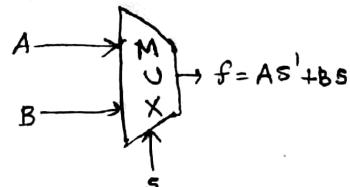
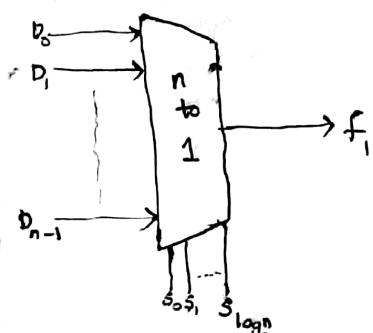
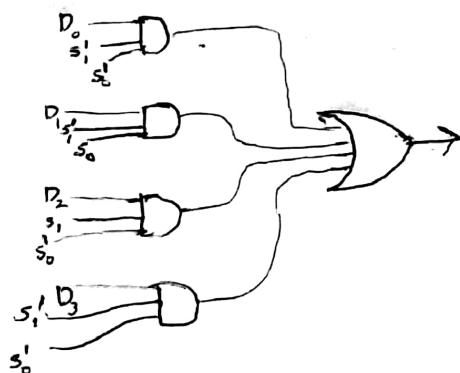
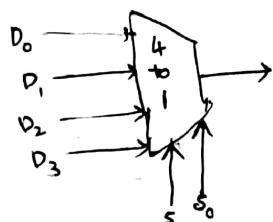
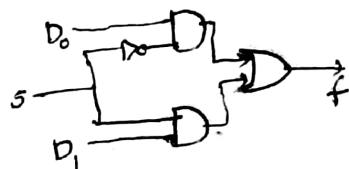
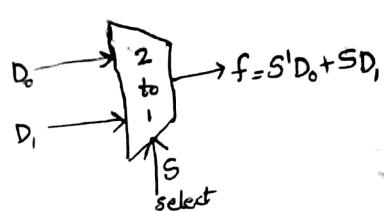
$$f_0 = x_1 y_1 + x_2 y_1 y_2 + x_1 x_2 y_2$$

$x_1 x_2$	$y_1 y_2$	f_2	f_1	f_0
00	00	0	1	0
00	01	0	0	1
00	10	0	0	1
00	11	0	0	1
01	00	1	0	0
01	01	0	1	0
01	10	0	0	1
01	11	0	0	1
10	00	1	0	0
10	01	1	1	0
10	10	0	1	0
10	11	0	0	1
11	00	1	0	0
11	01	1	1	0
11	010	1	0	0
11	11	0	1	0

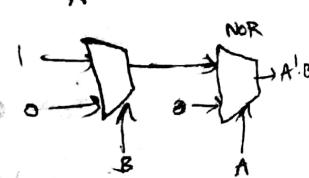
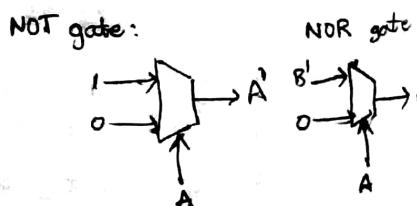
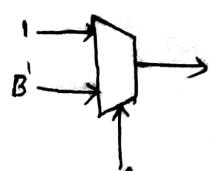
4-Bit Comparators



Data Selector



$$\text{NAND gate: } A' + AB' = A' + B' = (A \cdot B)'$$

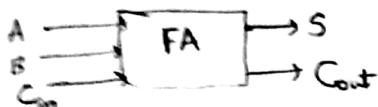


4-Bit ALU

$$\begin{array}{r}
 A \quad A_3 A_2 A_1 A_0 \quad 0101 \\
 B \quad B_3 B_2 B_1 B_0 \quad 1010 \\
 \hline
 & \underline{\quad\quad\quad\quad} \\
 & 1111
 \end{array}$$

S	C
0	0
0	1
1	0
1	1

$$S = A \oplus B \quad C = A \cdot B$$



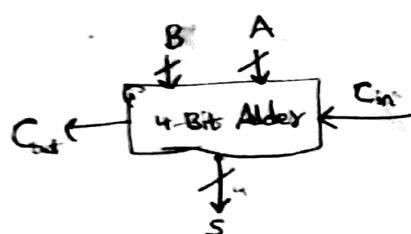
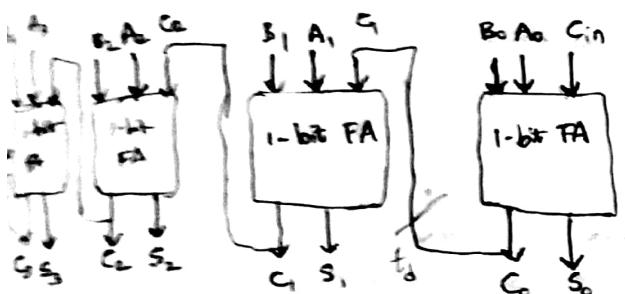
$$S = A \oplus B \oplus C_{in}$$

$$S = \overline{A}BC_{in} + \overline{A}B'C_{in} + A'BC_{in} + AB'C_{in}$$

$$C_{out} = A \cdot B + B \cdot C_{in} + C_{in} \cdot A$$



$$\begin{aligned}
 A(B \oplus C_{in}) + BC_{in} &= ABC_{in}^1 + BC_{in} + A'BC_{in}^1 + BC_{in} \\
 &= AB + AC_{in} + BC_{in}
 \end{aligned}$$



Ripple Carry Adder

n-bit Adder

Delay is $(n-1)t_g$

Challenge - How to reduce delay

Carry Look-ahead Adder - Generates carry well before

Full adder

$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = AB + BC_{int} + AC_{in}$$

$$= g + C_{in} \cdot p$$

Carry generate $g_i = A_i B_i$

Carry Propagate $p_i = A_i + B_i$

$$C_0 = g_0 + p_0 C_{in}$$

$$C_1 = g_1 + p_1 g_0$$

$$= g_1 + p_1 (g_0 + p_0 C_{in})$$

$$= g_1 + p_1 g_0 + p_1 p_0 C_{in}$$

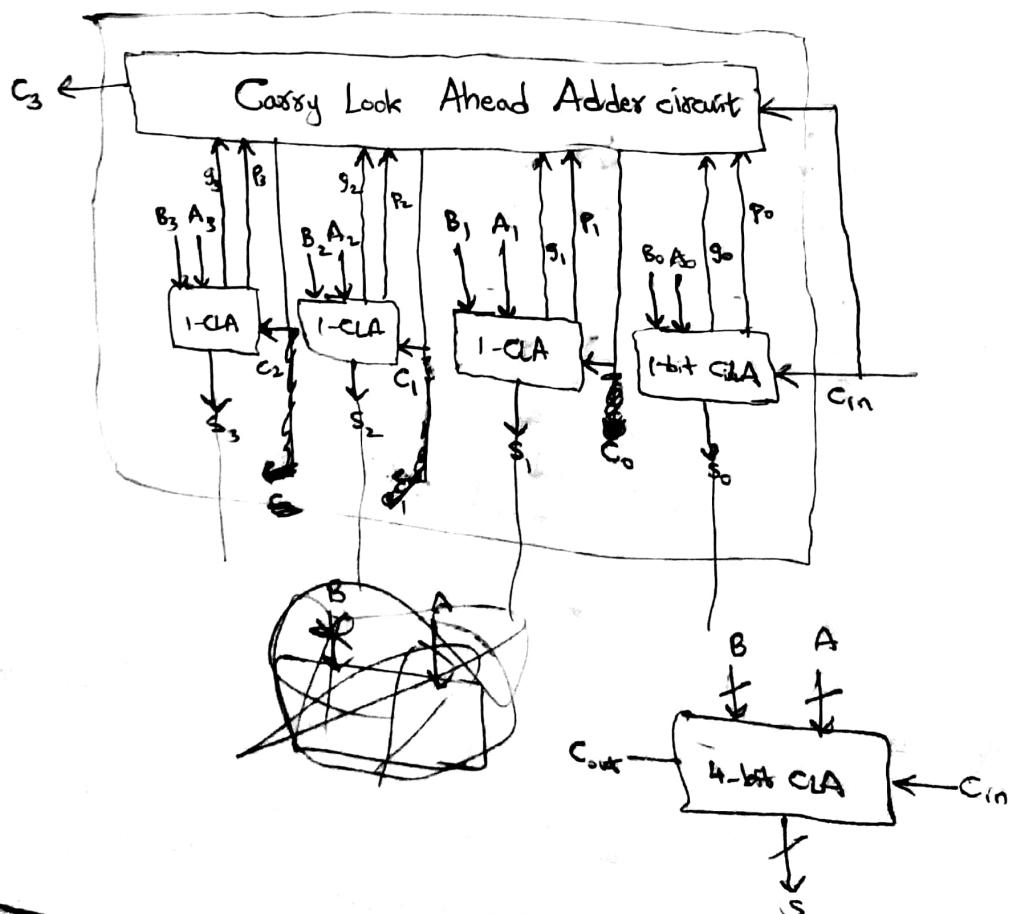
$$C_2 = g_2 + p_2 C_1$$

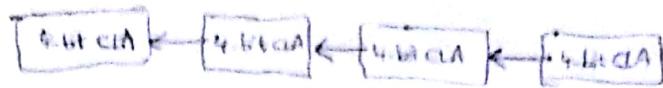
$$= g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 C_{in}$$

$$C_3 = g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0 + p_3 p_2 p_1 p_0 C_{in}$$

$$C_i = f(A_i, B_i, C_{in})$$

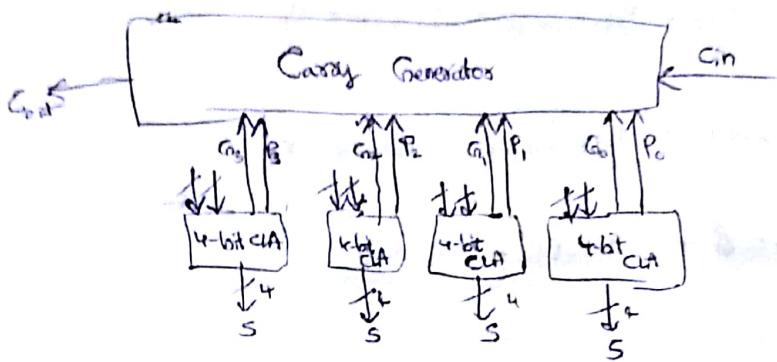
$C_i = f(A_i, B_i, C_{in})$ Requires more hardware





Instead of $13t_g$, it becomes $3t_g$

4-bit CLA & Ripple Carry.

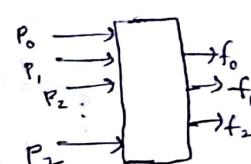
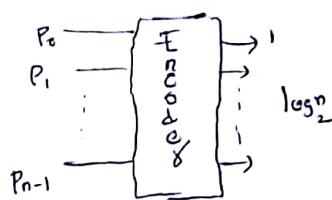


1/2/18

Combinational Circuits :-

- ① Priority Encoders
- ② Decodes

n no. of inputs $\log_2 n$ outputs



P_i, P_j works service

if $i > j$
 P_i will get served

o.w. P_j will get served

P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	f_2	f_1	f_0
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	1	0	1
0	0	0	0	0	1	0	0	1	1	0
0	0	0	0	0	0	1	0	1	0	1
0	0	0	0	0	0	0	1	1	1	1

$$f_2 = P_4 P_5' P_6' P_7 + P_5 P_6' P_7'$$

$$+ P_6 P_7 + P_7$$

$$= P_4 + P_5 + P_6 + P_7$$

~~for~~

Memory Decoder

$$f_1 = P_1 P_2' P_3' P_4' P_5' P_6' + P_3 P_4' P_5' P_6' P_7' + P_6 P_7' + P_7$$

$$= P_2 P_4' P_5' P_6' P_7' + P_3 P_4' P_5' P_6' P_7' + P_6 + P_7$$

$$= P_2 P_4' P_5' + P_3 P_4' P_5' + P_6 + P_7$$

$$f_0 = P_1 P_2' P_3' P_4' P_5' P_6' P_7' + P_3 P_4' P_5' P_6' P_7' + P_5 P_6' P_7' + P_7$$

$$= P_1 P_2' P_3' P_5' P_6' P_7' + P_3 P_4' P_5' P_6' P_7' + P_5 P_6' P_7' + P_7$$

$$= P_1 P_2' P_4' P_5' + P_3 P_4' P_5' + P_5 P_6' + P_7$$



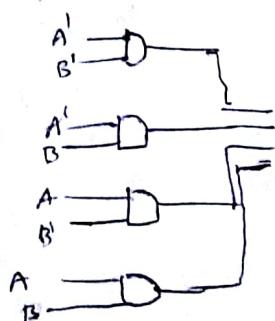
Decoder :-



Encoder
Decoder

4.6. Let $f =$

f_1



4.5

f_1

f_2

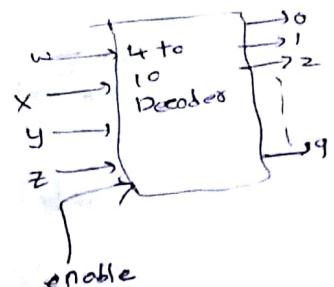
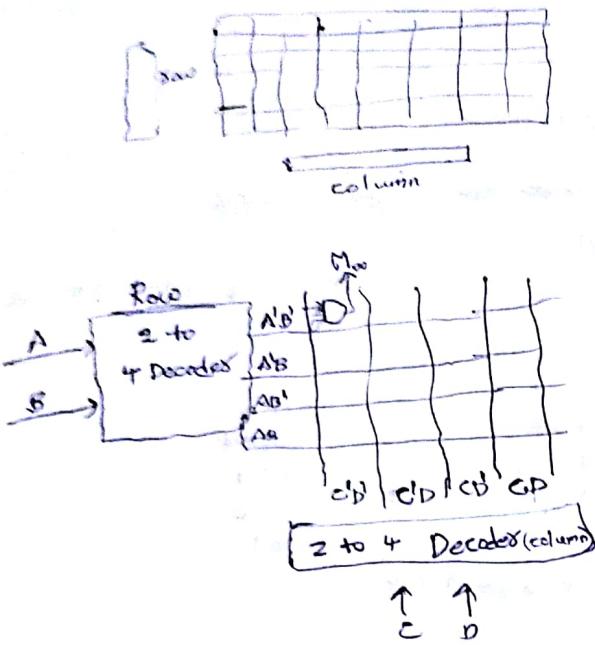
4.26

A

(a) P_8

f

Memory Decoder



Chapter 4:

4.4 Let $f = \sum(5, 6, 13)$

$$f_1 = \sum(1, 2, 3, 5, 6, 8, 9, 10, 11, 13)$$

Find f_2 such that $f = f_1 \cdot f_2'$

$$\left. \begin{array}{l} f_2' = \sum(5, 6, 13, \underline{4, 7, 12, 14, 15}) \\ \text{optional} \end{array} \right\}$$

Is f_2 unique?

If not, indicate all possibilities

4.5

$$f_1 = \sum(1, 3, 4, 5, 9, 10, 11) + \sum_{\emptyset}^{(4, 8)}$$

(a) $f_3 = f_1 \cdot f_2'$

$$f_2 = \sum(0, 2, 4, 7, 8, 15) + \sum_{\emptyset}^{(9, 12)}$$

(b) $f_4 = f_1 + f_2'$

4.26

A gate T is defined as follows



		AB	00	01	11	10
		C	0	0 0	1	0
			0	1	0	1
0	1		0	1	0	1

- ② Prove that if the logic value '1' is given then T is functionally complete

(b) Realize $f(w, x, y, z) = \sum(0, 1, 2, 4, 7, 8, 9, 10, 12, 15)$ by means of T gates.

The following

5.5

$$T(A, B, C) = A'BC + ABC' + AB'C$$

$$T(AB, 1) = A'B + AB' = \overline{(AB)}' = A \oplus B$$

$$T(A, 1, 1) = A'$$

$$T(T(A, 1, 1), B, B) = AB$$

wx\yz	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	1	0	0	0
10	0	1	0	1

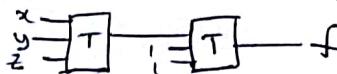
yz\wx	00	01	11	10
00	0	1	-	-
01	1	0	-	-
11	0	-	-	-
10	1	-	-	-

$$f = y'z' + x'y' + xyz + x'z'$$

$$= xy'z' + x'y'z' + x'y'z + xyz + x'y'z'$$

$$f' = (x + y + z') \cdot (x' + y + z) \cdot (x + y + z)$$

$$\Rightarrow f' = xyz' + x'yz + xy'z$$



5.4 You are supplied with just one NOT gate and an unlimited amount of AND & OR gates. Design a circuit that realizes

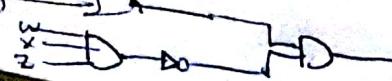
$$T(w, x, y, z) = w'x + wy + yz$$

Inputs are available in uncomplemented form.

$$T(w, x, y, z) = w'x + x'y + \cancel{yz}$$

wx\yz	00	01	11	10
00	0	0	1	1
01	1	0	1	1
11	1	0	0	1
10	0	1	1	1

$$(x+y) \cdot (w' + x' + z') = (x+y) \cdot (wxz)'$$

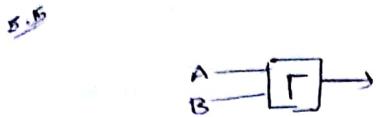


Eq:- $g(x_1, x_2, x_3)$

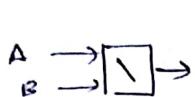
Do 5.6 5.

A, B, C

The following tables define two devices.



	A		
	0	1	2
0	0	2	0
1	0	1	1
2	2	1	0



	A		
	0	1	2
0	0	2	0
1	0	2	0
2	0	0	2

Find $f(A, B)$ that realizes



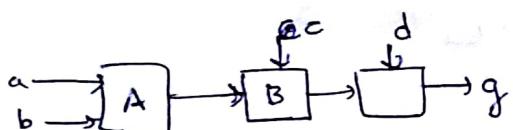
Q.

	A		
	0	1	2
0	2	2	2
1	0	1	0
2	0	2	0

Do

5.6 5.7 5.8 5.9

$$g(x_1, x_2, x_3, x_4) = \sum(4, 6, 7, 15) + \sum_{\emptyset}(2, 3, 5, 11)$$



Find the correspondence between x_i & a, b, c, d & determine

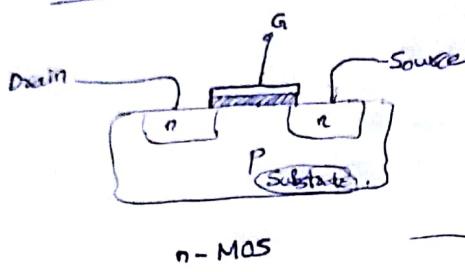
A, B, C

	x_3x_4			
	00	01	11	10
x_1x_2	00	0	0	ϕ
01	1	ϕ	1	1
11	0	0	1	0
10	0	0	ϕ	0

$$x_1x_2 + x_3x_4$$

9/21/18

CMOS

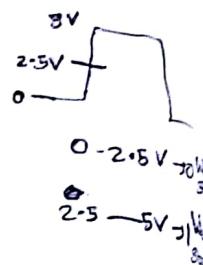
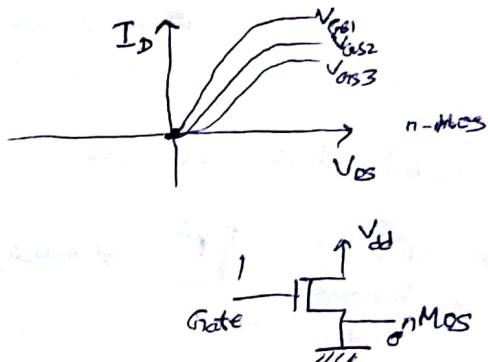


n-MOS

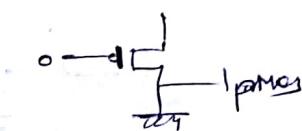
nMOS: Passes strong 0,
Weak 1

pMOS: passes a strong 1,
Weak 0

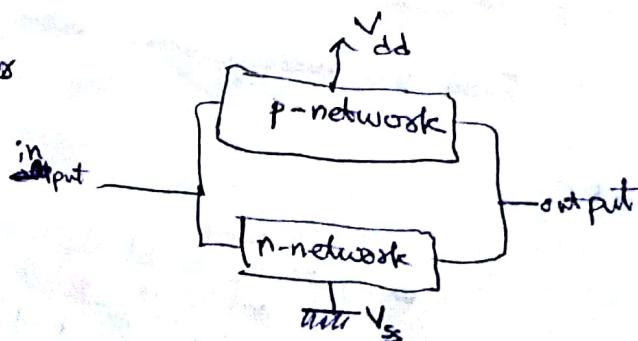
CMOS
Complementary



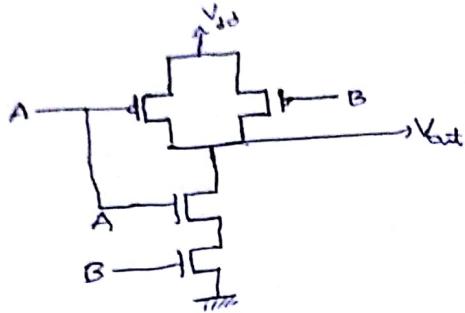
nMOS pMOS



NOT/
CMOS inverter

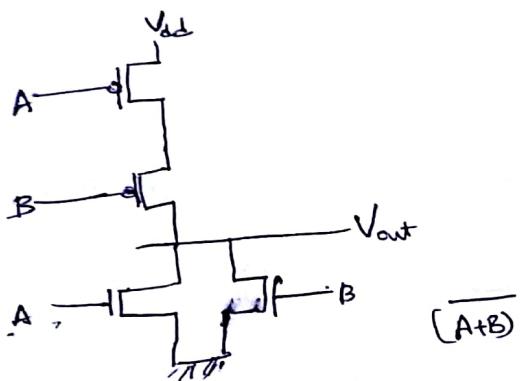


2-input NAND



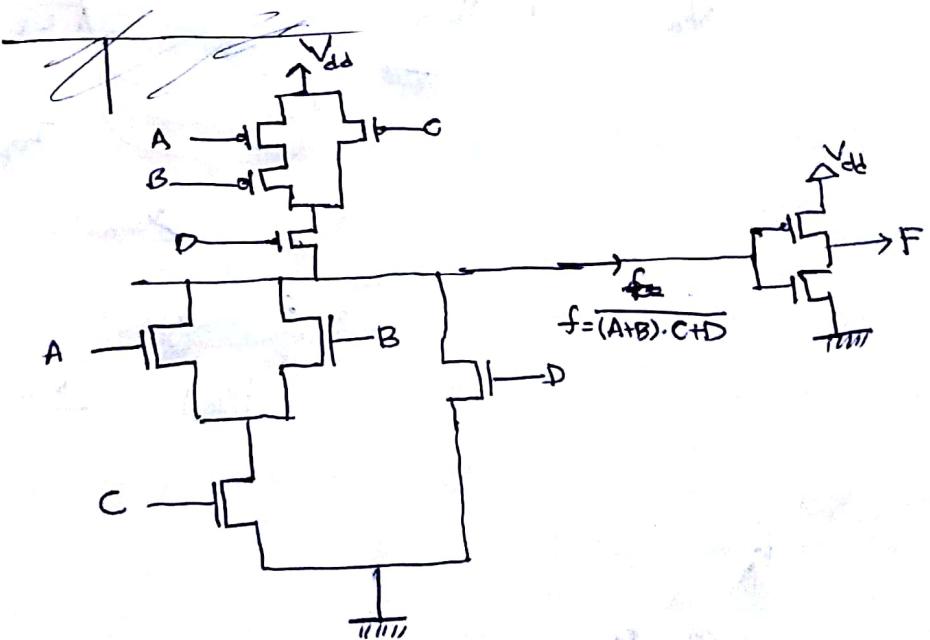
A	B	Vout
0	0	1
0	1	1
1	0	1
1	1	0

2-input NOR



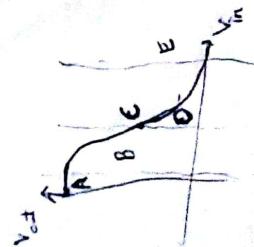
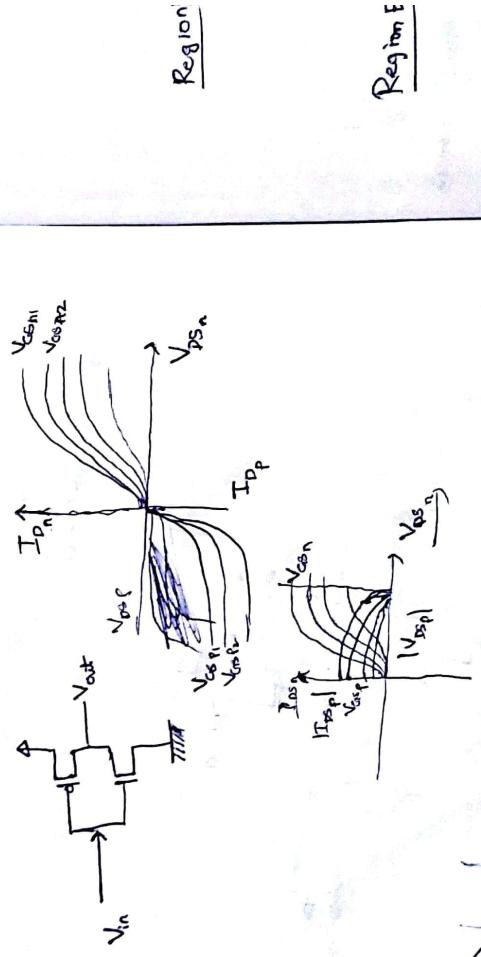
A	B	Vout
0	0	1
0	1	0
1	0	0
1	1	0

$$f = \overline{(A+B) \cdot C + D}$$





Cmos Inverter Characteristics :-



Region A: $V_{in} \leq V_{th}$

p-off

n-off

p-on

$$V_{out} = V_{dd}$$

Region B:

$$V_{th} \leq V_{in} \leq \frac{V_{dd}}{2}$$

p - non saturated

n - saturated

$$I_{DSp} = -I_{DSn}$$

$$V_{out} = (V_{in} - V_{tp}) + \sqrt{(V_{in} - V_{tp})^2 - 2(V_{in} - \frac{V_{dd}}{2} + V_{tp})V_{dd} - \frac{\beta_n}{\beta_p} (V_{in} - V_{tp})^2}$$

$$\beta = \frac{mc}{t} \left(\frac{W}{L} \right) \frac{\text{Width}}{\text{Length}}$$

Region C:

n & p both are saturation

$$V_{in} = \frac{V_{dd}}{2}$$

$$V_{out} > V_{in} - V_{tp}$$

$$V_{out} < V_{in} - V_{tn}$$

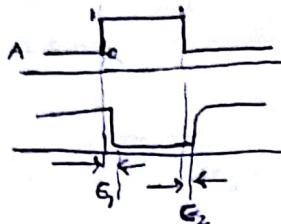
Region D:-

$$\frac{V_{dd}}{2} < V_{in} < V_{dd} + V_{tp}$$

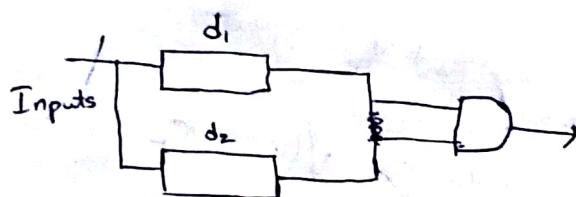
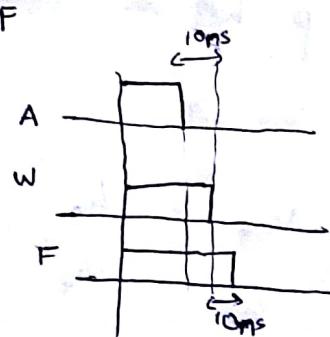
$$V_{out} = (V_{in} - V_{tn}) + \sqrt{(V_{in} - V_{tn})^2 - \frac{\beta_p}{\beta_n} (V_{in} - V_{dd} - V_{tp})^2}$$

Region E:- $V_{in} > V_{dd} + V_{tp}$, $V_{out} = 0$

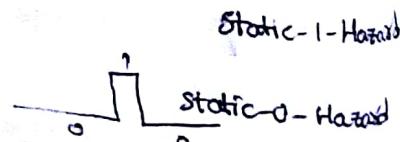
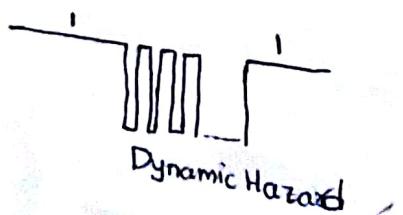
Gates Delays and Hazards

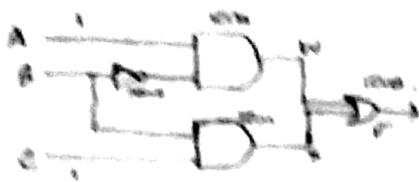


E_1 - Inverter Delay
 E_2 - " "
 slightly higher



$d_1 \sim d_2$ is not negligible
 Momentumarity, the output will be wrong

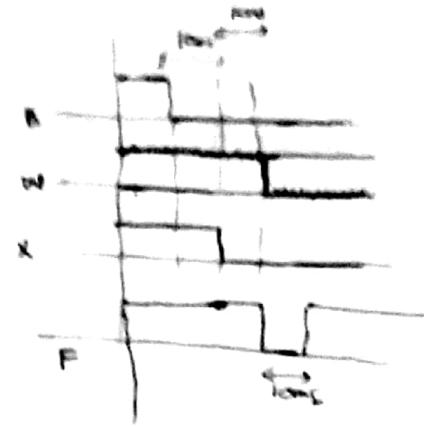




$$F = AB' + BC$$

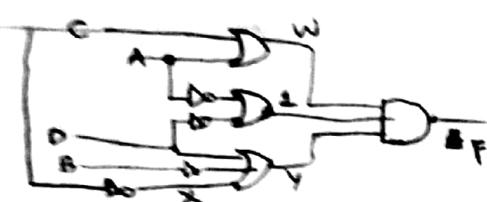
$$\text{If } A = C = 1,$$

$$F = B + B' = 1$$



AB'C	00	01	11	10
0	0	0	0	0
1	1	0	1	0

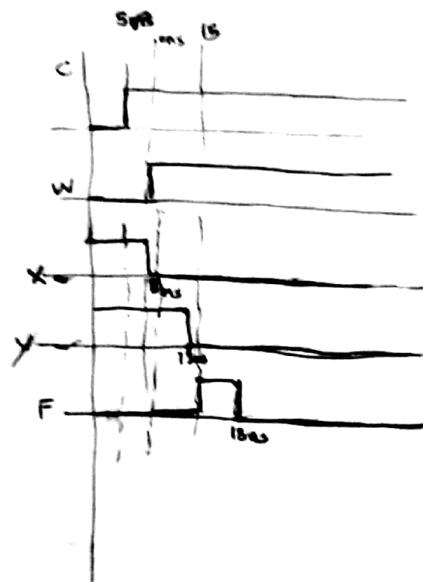
$$F = (A+C)(A'+D')(B'+C'+D)$$



$$\text{Delay } \xrightarrow{3\text{ ms}} \quad A=0, D=0, B=1 \quad C \rightarrow 1$$

$$\Rightarrow D, \xrightarrow{5\text{ ms}}$$

$$W=C$$



AB	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	1	1	0	0
10	1	0	0	1

Hazards occur at these changes

$$F = (A+C)(A'+D') \xrightarrow{(B'+C+D)} (C+D')(A+B'+D)(C'+B'+A')$$

Hazard Free circuit

NAND - NAND
 NOR - NOR circuit

$$F = (F')'$$

Two-level realization & Multi-level realization

$$F = A + BC' + B'CD' \quad \text{— AND-OR}$$

$$\begin{aligned} (F')' &= [(A + BC' + B'CD')']' \\ &= ((A)' \cdot (BC')' \cdot (B'CD')')' \quad \text{NAND-NAND} \\ &\approx (\overline{A} \cdot \overline{B} \cdot \overline{C}) \cdot (\overline{B} \cdot \overline{C} \cdot \overline{D}) \\ &= A + (B' + C)' + (B + C' + D)' \quad \text{NOR-OR} \end{aligned}$$

$$= ((A + (B' + C)' + (B + C' + D)')')' \quad \text{NOR-NOR}$$

~~(B+C)~~

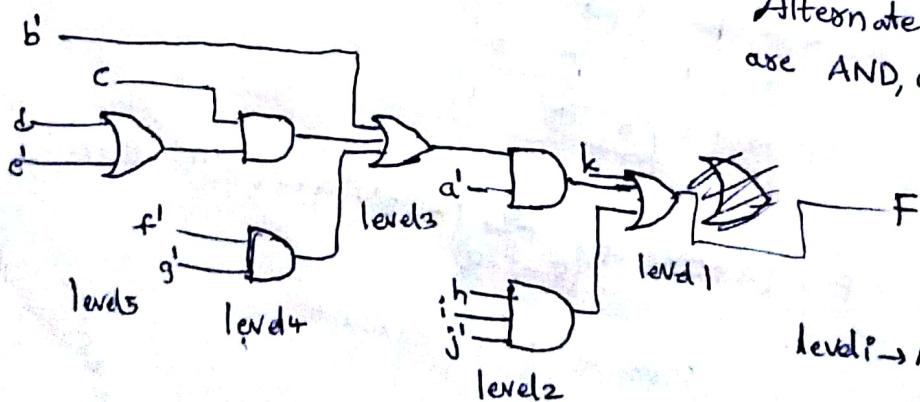
$$F = (A + B + C)(A + B' + D')(A' + C + D)$$

~~A'B'C'D' + A'B'C'D + A'BC'D'~~

$$= ((A + B + C)' + (A + B' + D')' + (A' + C + D)')'$$

$$F = a'[b' + c(d + e') + f'g'] + hij' + k'$$

Alternate levels
are AND, OR



level 1 → All odd gates
are OR

* Replace all gates by
NAND for

NAND-NAND realization

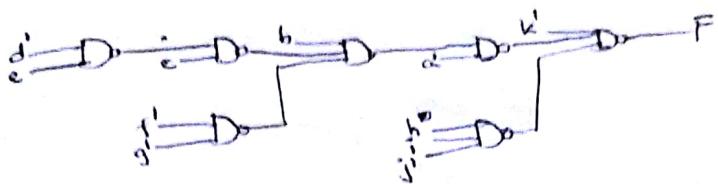
* Don't change the in...
out

→ For ad

d
e

Techno

For odd number levels, any inputs (literals), change it as its complement



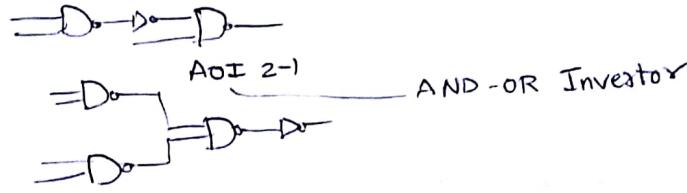
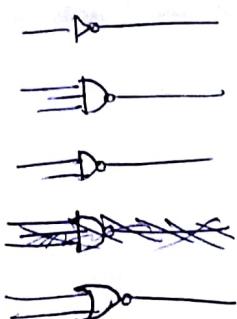
Cost of one basic gate : Area, Gate delay (propagation delay)

Technology Mapping :-

Hardware Development	Description Language	Verilog VHDL
----------------------	----------------------	-----------------

Cell Library

All base functions



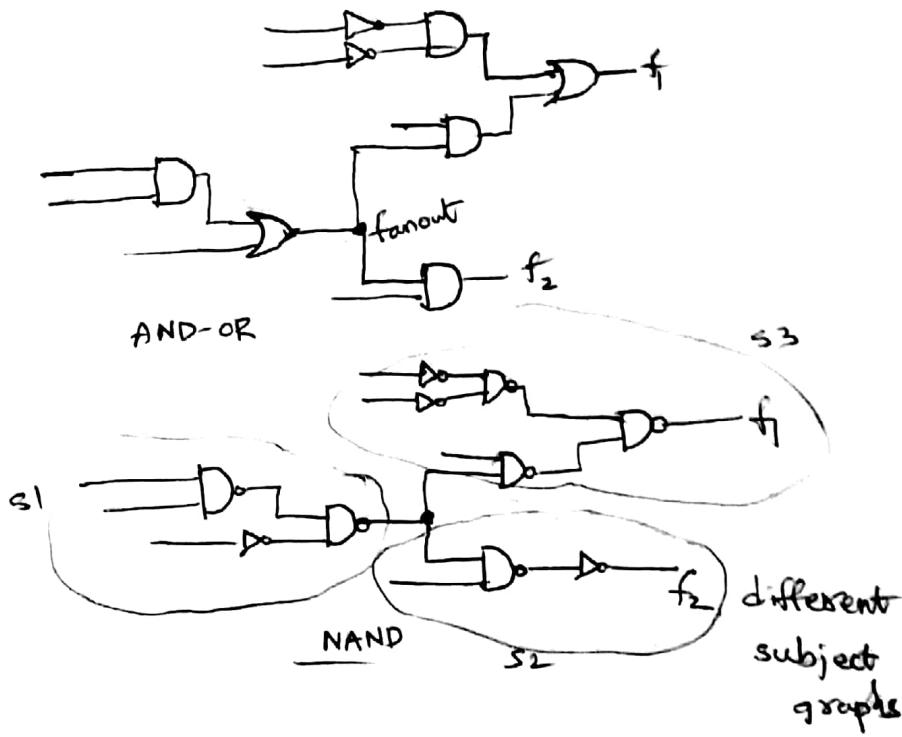
Pattern - Graph

$$\lambda = 0 \text{ to } 13 \text{ nm}$$

$$\approx 0.8 \mu$$

Cells	Area	Delay
INV	1	2
NAND2	2	3
NAND3	3	3
NOR2	2	3
NOR3	3	3
AOI 2-1	4	1
AOI 2-2	5	1

- Decompose a network into Base function (circuit)
- Partition a network into subject graph
- Obtaining a network cover i.e.
try to match the subject graph & pattern graph



Always break the fanout point to get different subject graphs.

Compute the delay & area of subject graph