

Logic Optimization

Virendra Singh

Professor

Computer Architecture and Dependable Systems Lab

Department of Electrical Engineering

Indian Institute of Technology Bombay

<http://www.ee.iitb.ac.in/~viren/>

E-mail: viren@ee.iitb.ac.in

CS-226: Digital Logic Design



Lecture 13-B: 15 February 2021

CADSL

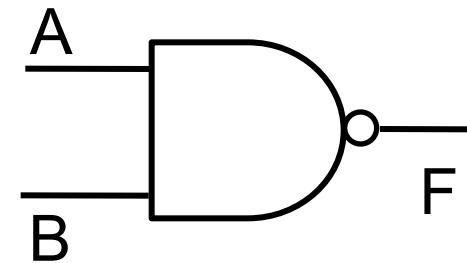
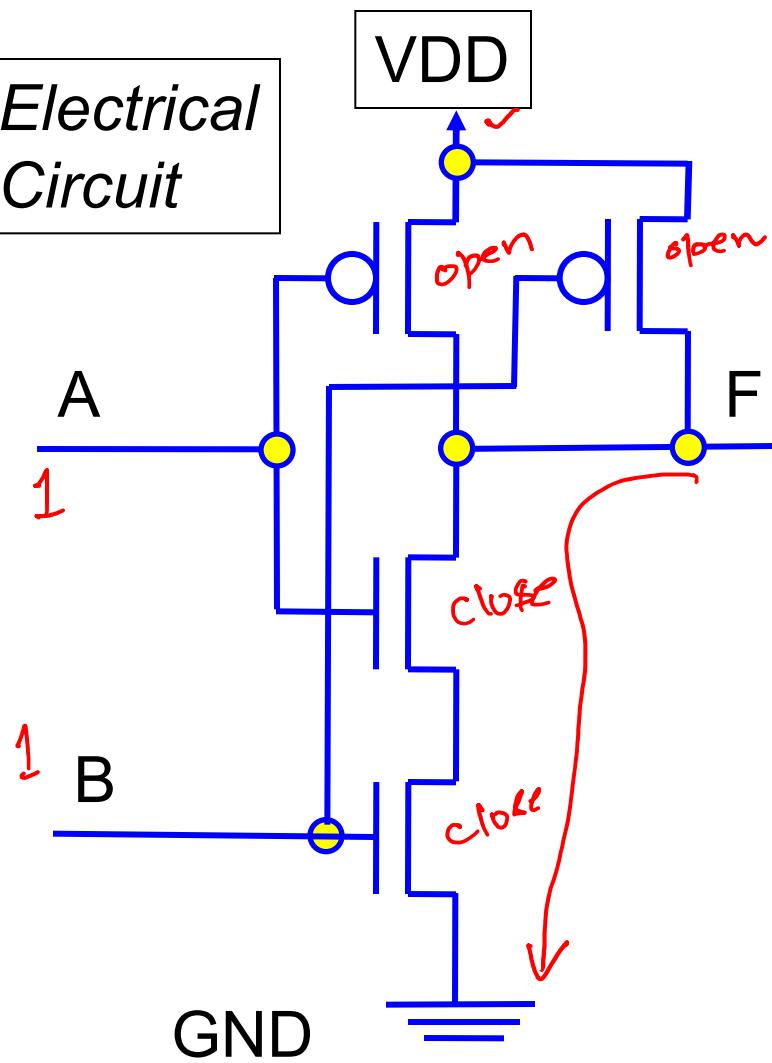
CMOS Gates

Logic function	Number of transistors		
	1 or 2 inputs <i>(cost)</i>	N inputs <i>(cost)</i>	<i>delay</i>
NOT	✓ 2	1	-
AND	✓ 6	3	$2N + 2$
OR	✓ 6	3	$2N + 2$
NAND	✓ 4	2	$2N$
NOR	✓ 4	2	$2N$



CMOS Logic Gate: NAND

Electrical Circuit



Transistor delay = τ (1 unit)

delay (2 input NAND gate)
= $\underline{2\tau}$ (2 units)

delay (N input NAND - gate)
= $\underline{N\tau}$ (N units)



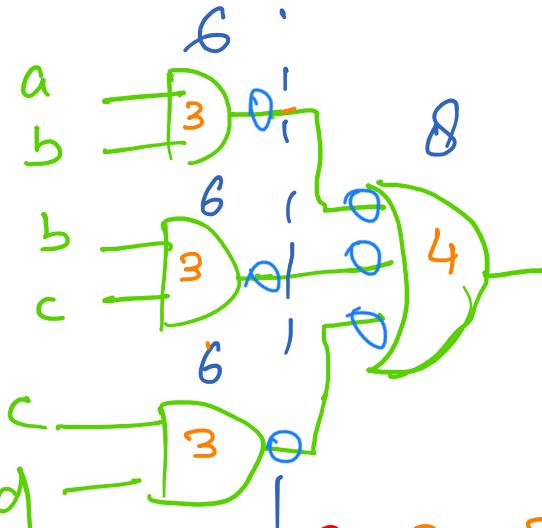
Function Minimization

Optimization

- ① Area/Cost
- ② Performance/Delay
- ③ Feasibility
- ④ Power

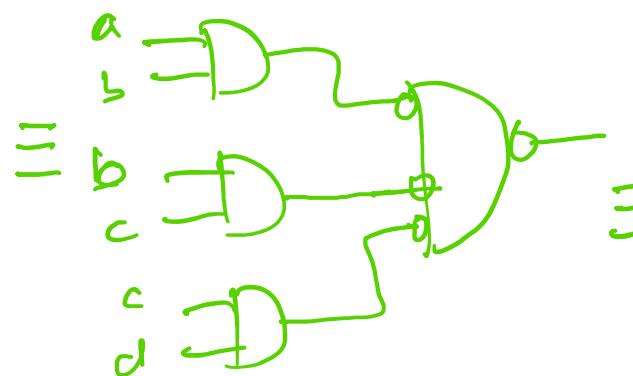
SOL

$$f(a, b, c, d) = \overbrace{ab}^6 + \overbrace{bc}^6 + \overbrace{cd}^6 = \frac{6 \times 2}{3 \times 2} = \frac{12}{6}$$



$$\text{Cost} = 3 \times 6 + 8 = 26$$

$$\text{delay} = 3 + 4 = 7$$



$$\text{Cost} = 4 \times 3 + 6 = 18$$

$$\text{delay} = 2 + 3 = 5$$



Function Minimization

$$f(a, b, c, d) = ab + bc + cd$$

cost \propto # literal
 \propto # product terms

Cost \propto (# literals + # product terms)

delay \propto max{# literals in
product terms}

\propto # literals product terms

delay \propto (max{# literals in } + # product terms)

min (# product terms) \rightarrow minimize C



Function Minimization

$$\min (\# \text{ literals}) \rightarrow \text{Cost}$$
$$\min (\max \{\# \text{ literals in prod. terms}\}) \rightarrow \text{delay}$$

$$\boxed{\min (\# \text{ product terms})}$$
 - minimizes both cost & delay.

CMOS Technology	Cost \propto # inputs	delay \propto # input
TTL	Cost \propto # inputs	delay = constant



minimize (# product terms)

Function → Truth table / SOP / POS / POM / SOM / RM / RDD.

$$f(a,b) = \bar{a}\bar{b} + a\bar{b}$$

a	b	f(a,b)
0	0	1
0	1	0
1	0	1
1	1	0

$$\bar{a}\bar{b} \rightarrow f.$$

Implicant

$$a\bar{b} \rightarrow f$$

Implicant



Logic Minimization

- Generally means
 - In SOP form:
 - Minimize number of products (reduce gates) and
 - Minimize literals (reduce gate inputs)
 - In POS form:
 - Minimize number of sums (reduce gates) and
 - Minimize literals (reduce gate inputs)

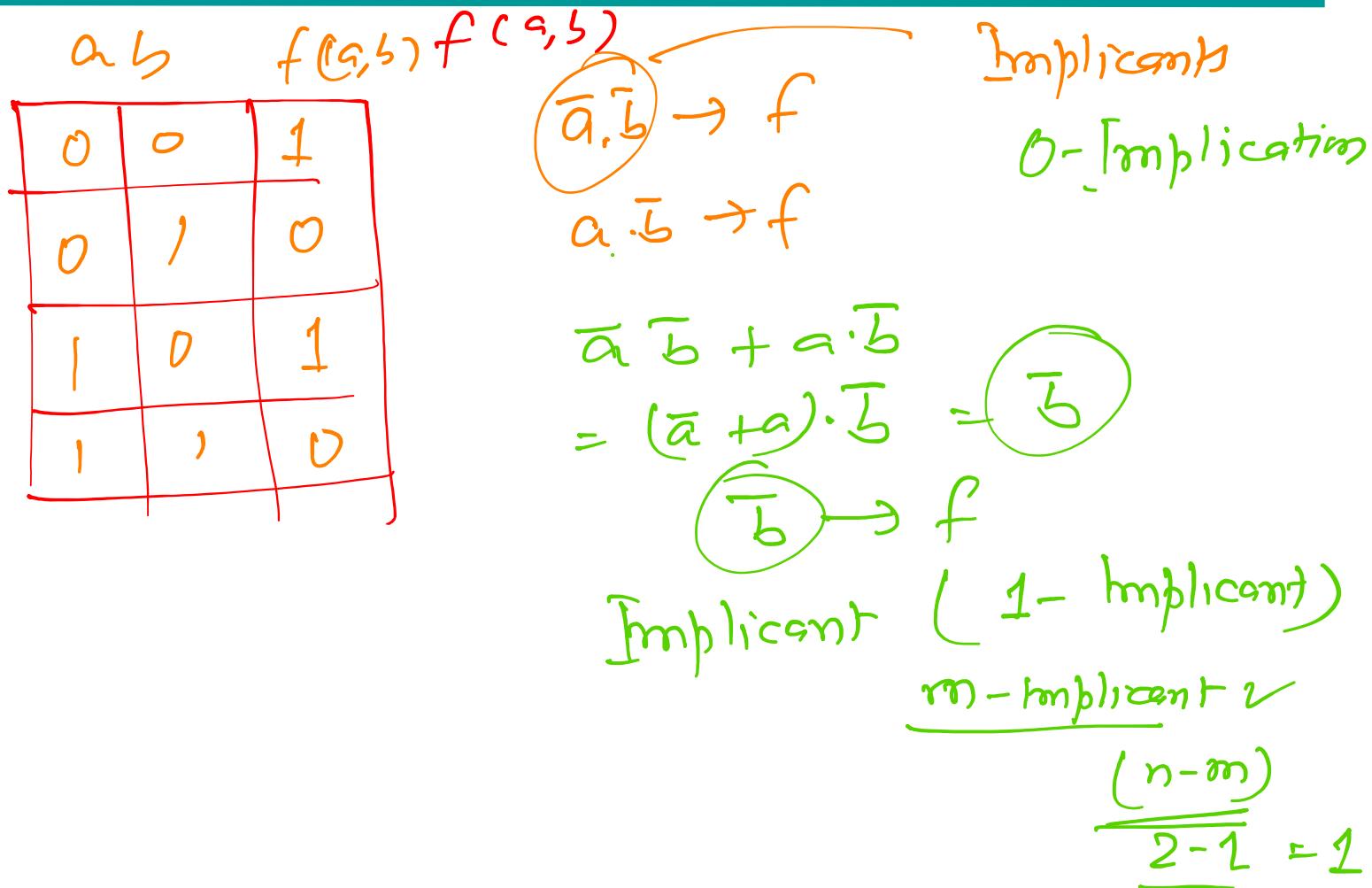
$\min(\# \text{ Sum clauses}) \rightarrow \text{minimize cost delay}$

$\min(\# \text{ literals}) \rightarrow \text{minimize cost}$

$\min(\max\{\# \text{ literals in sum clauses}\}) \rightarrow \text{minimize delay}$



Function Minimization



Product or Implicant or Cube

- Any set of literals ANDed together.
- Minterm is a **special case** where all variables are present. It is the largest product.
- A minterm is also called a 0-implicant of 0-cube.
- A 1-implicant or 1-cube is a product with one variable eliminated:

- Obtained by combining two adjacent 0-cubes

$$\underbrace{ABCD}_{\checkmark} + \underbrace{ABC\bar{D}}_{\checkmark} = ABC(D + \bar{D}) = ABC$$

$ABC \rightarrow f$
1- implicant



Truth Table: Two Bit Adder

X	Y	Binary Sum (C)(S)
0	0	0 0
0	1	0 1
1	0	0 1
1	1	1 0

$$x \cdot y \rightarrow c$$

$\circ\text{-Implicon}$

A diagram illustrating a binary addition operation. It features two blue arrows originating from the bottom left and pointing upwards and to the right. The word "CARRY" is positioned below the first arrow, and the word "SUM" is positioned below the second arrow. Above the arrows, the digits "1" and "0" are displayed in red and blue respectively, representing the bits being added.

$$\bar{x} \cdot y \rightarrow f$$



Logic Minimization

$\min(\# \text{ literals})$

$\min(\# \text{ prod. terms})$

minimization

1 Algebraic technique (Boolean Algebra) \Rightarrow SCALABILITY

2 Algorithmic techniques
(Graph technique) \rightarrow K-Map
QM



Thank You

