# 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

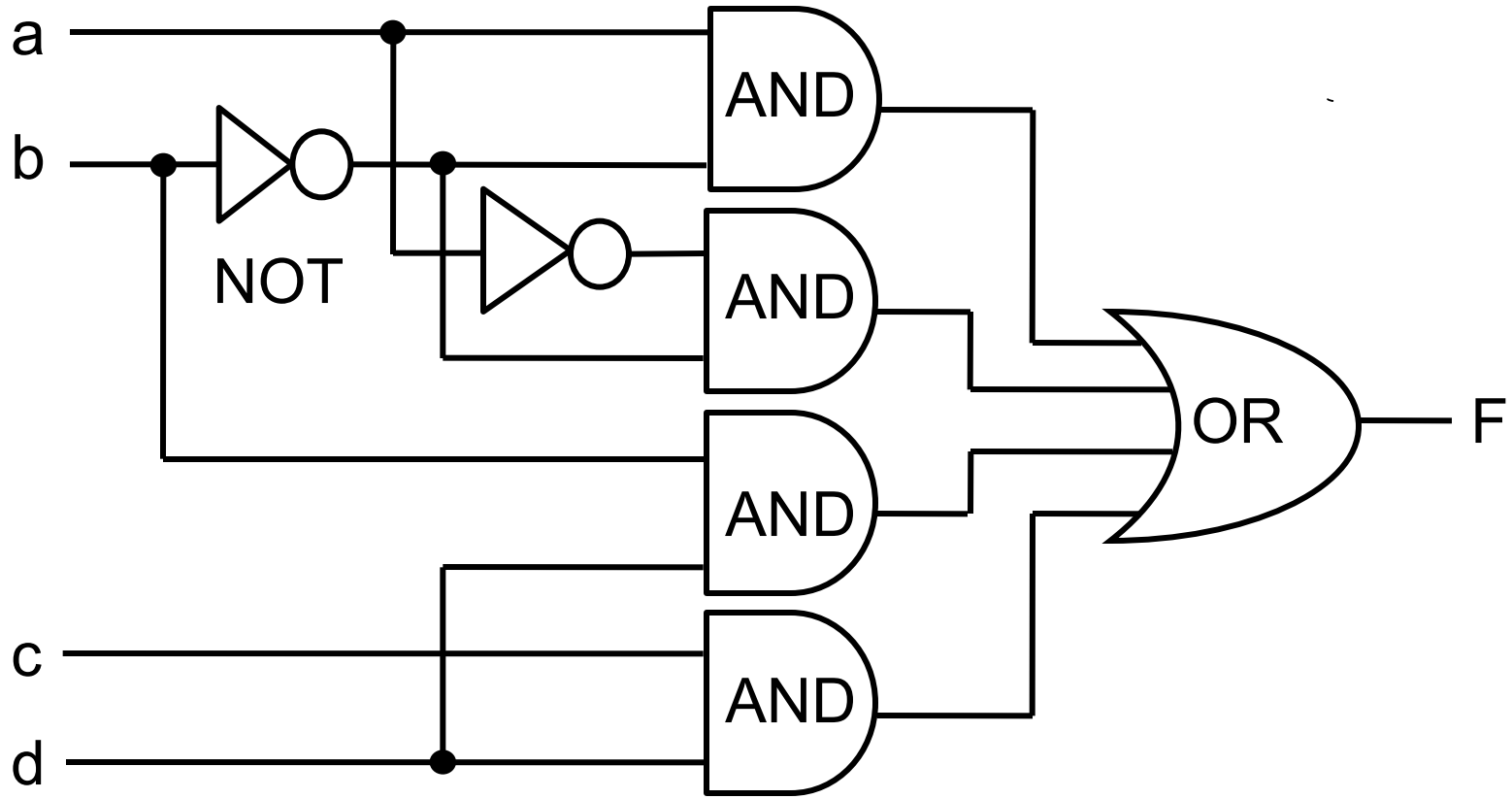
*CS-226: Digital Logic Design*

*Lecture 13-A: 15 February 2021*

CADSL

# Understanding Logic Minimization

- Logic function: $F = a\bar{b} + \bar{a}\bar{b} + bd + cd$



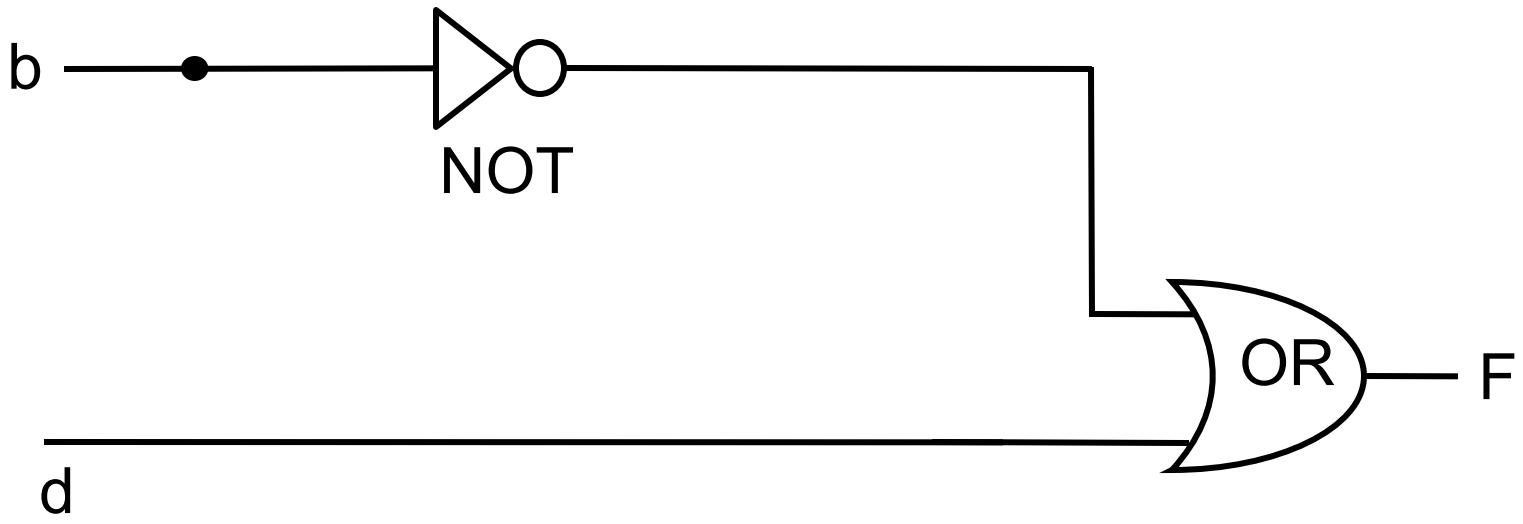
# Algebraic Logic Minimization

- Reducing products:

| | |
|---|---|
| $F = a\bar{b} + \bar{a}\bar{b} + bd + cd$ | Distributivity |
| $= \bar{b}(a + \bar{a}) + bd + cd$ | Complementation |
| $= \bar{b}1 + bd + cd$ | Identity |
| $= \bar{b}(c + \bar{c}) + bd + cd$ | Complementation |
| $= bd + \bar{b}c + cd + \bar{b}\bar{c}$ | Distribitivity |
| $= bd + \bar{b}c + \bar{b}\bar{c}$ | Consensus theorem |
| $= bd + \bar{b}(c + \bar{c})$ | Distributivity |
| $= bd + \bar{b}$ | Complement, identity |
| $F = \bar{b} + d$ | Adsorption |

# Minimized Circuit

- Minimized expression: $F = \bar{b} + d$




Cost, performance

# Standard Sum-of-Products (SOP)

- A Simplification Example:
- $F(A, B, C) = \Sigma m(1,4,5,6,7)$
- Writing the minterm expression:

  F = A'B'C + AB'C' + AB'C + ABC' + ABC

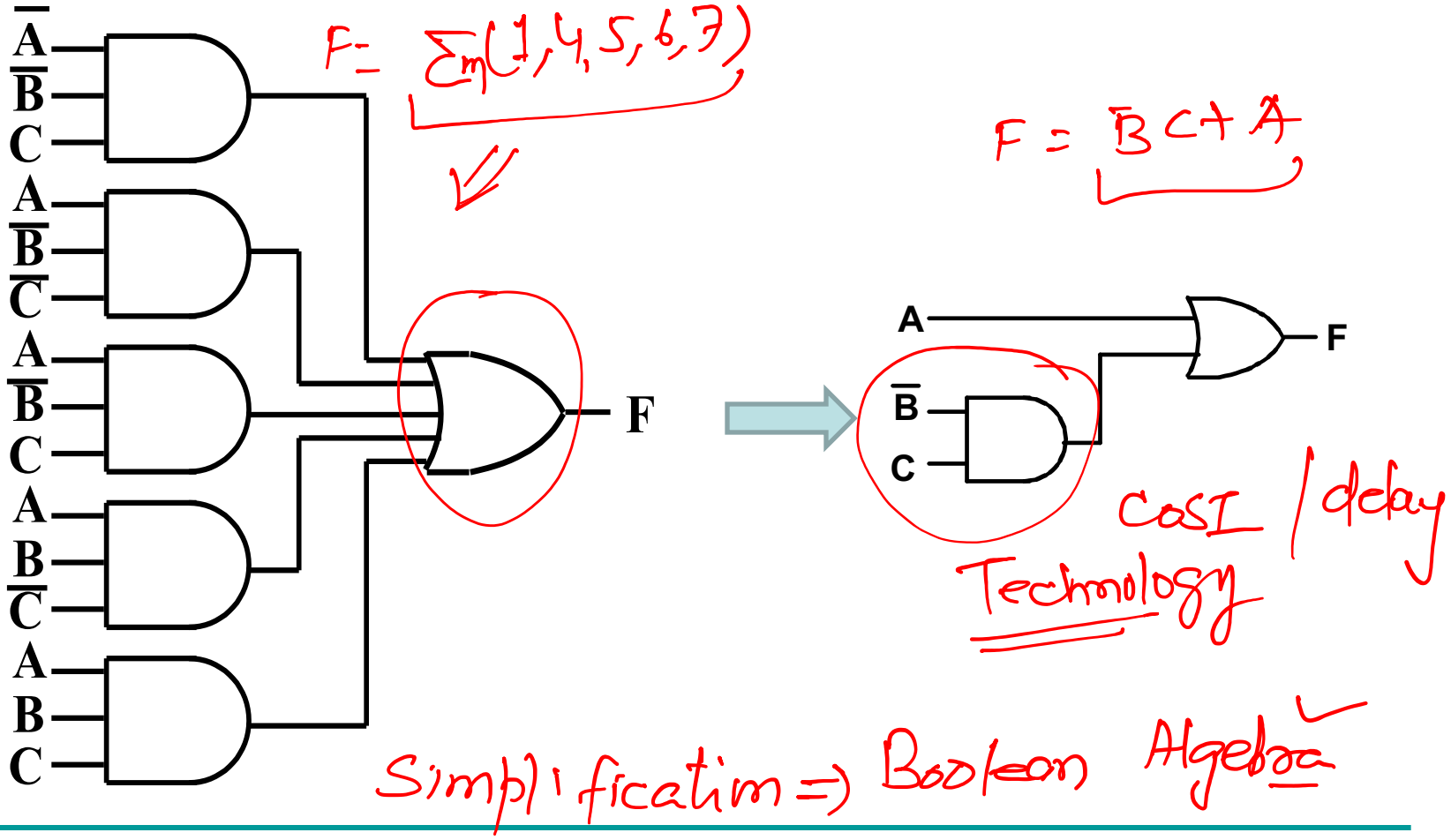
- Simplifying:

```
F = A′ B′ C + A (B′ C′ + B C′ + B′ C + B C)
  = A′ B′ C + A (B′ + B) (C′ + C)
  = A′ B′ C + A·1·1
  = A′ B′ C + A
  = B′C + A
```

- Simplified F contains 3 literals compared to 15 in minterm F

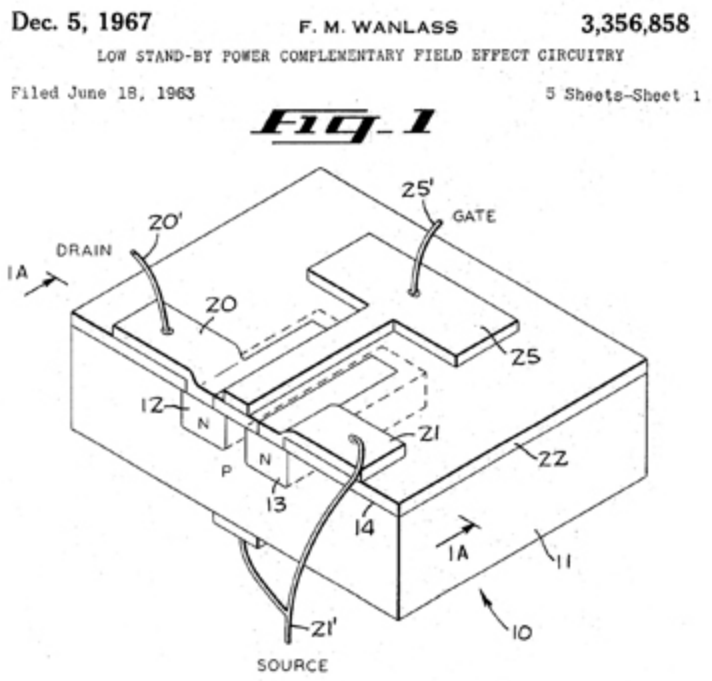
# AND/OR Two-level Implementation of SOP Expression

- The two implementations for F are shown below – it is quite apparent which is simpler!



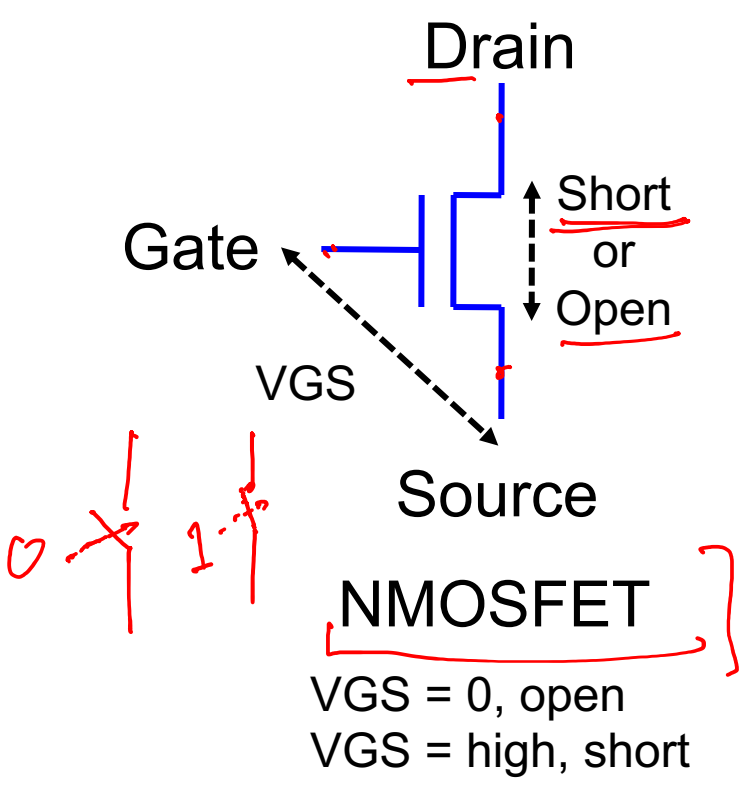
# CMOS Circuit

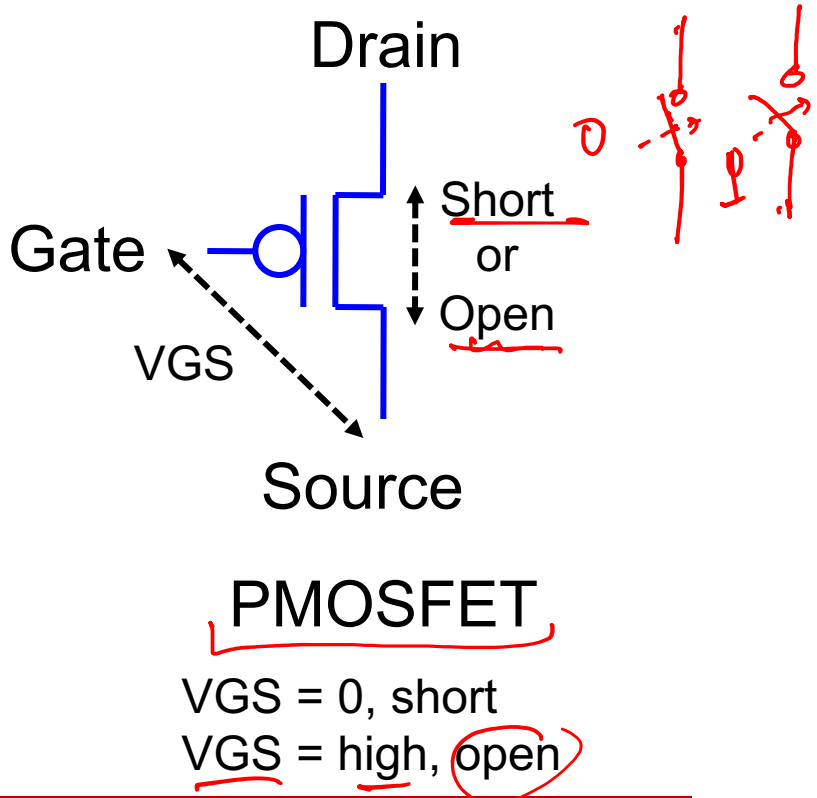
TTL




Wanlass, F. M. "Low Stand-By Power Complementary Field Effect Circuitry." *U. S. Patent 3,356,858* (Filed June 18, 1963. Issued December 5, 1967).

# MOSFET (Metal Oxide Semiconductor Field Effect Transistor)



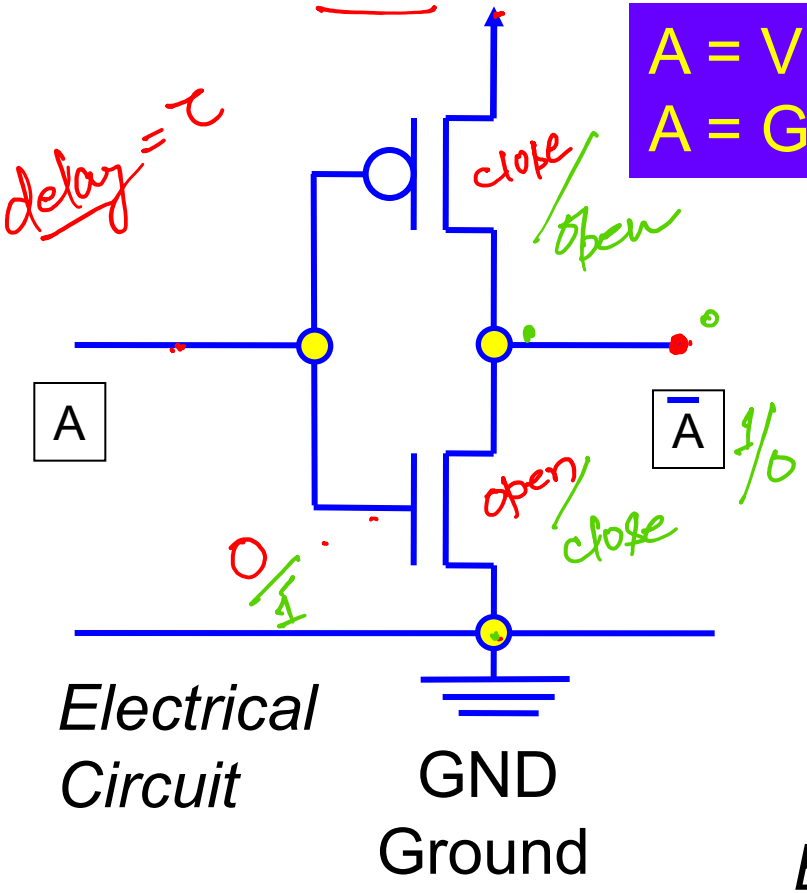

Reference:
R. C. Jaeger and T. N. Blalock, *Microelectronic Circuit Design, Third Edition*, McGraw Hill.

# CMOS NOT Gate (Modern Design)

Power supply

VDD = 1 volt; voltage depends on technology.

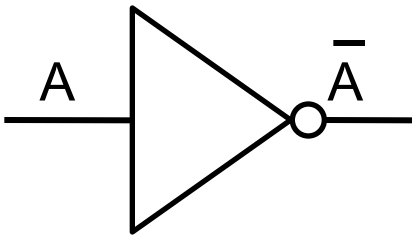
A = VDD = 1 volt is state "1"
A = GND = 0 volt is state "0"




*Electrical Circuit*

| Truth Table | |
|---|---|
| A | $\overline{A}$ |
| 0 | 1 |
| 1 | 0 |

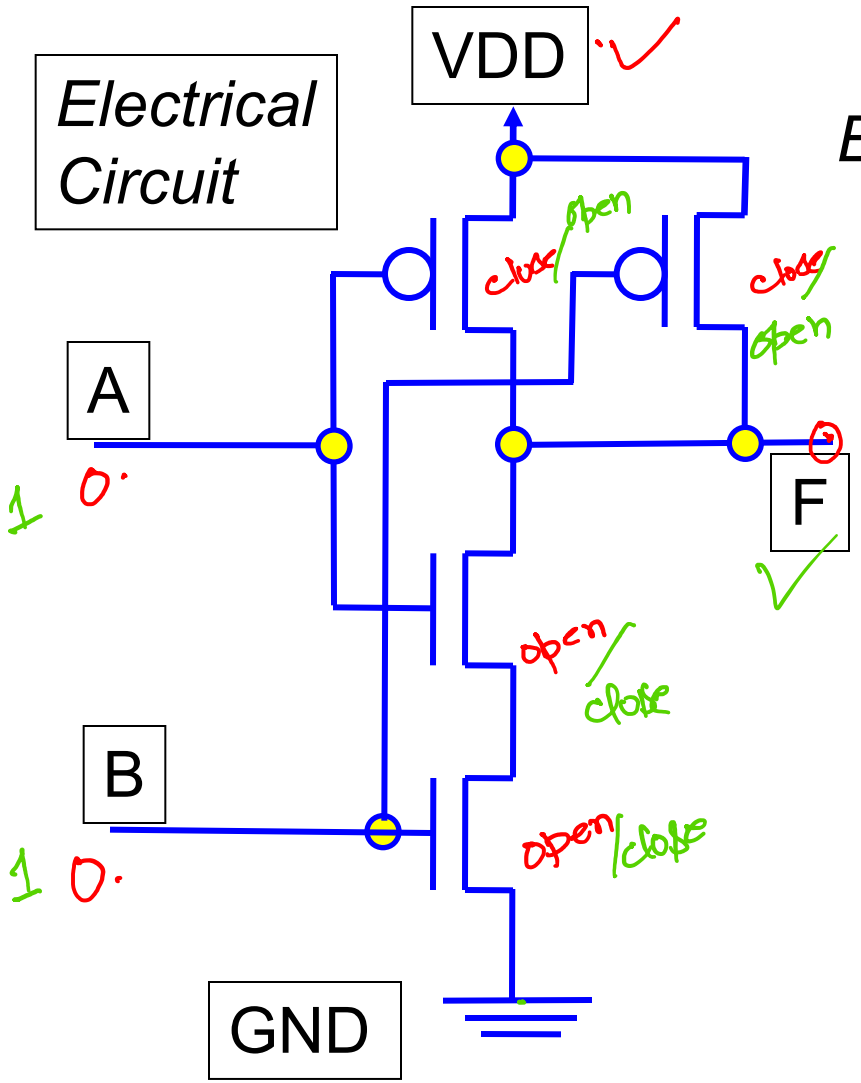
*Boolean Function*




*Symbol*

# CMOS Logic Gate: NAND

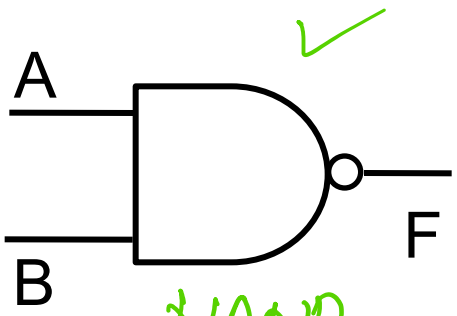
*Electrical Circuit*




*Boolean Function*

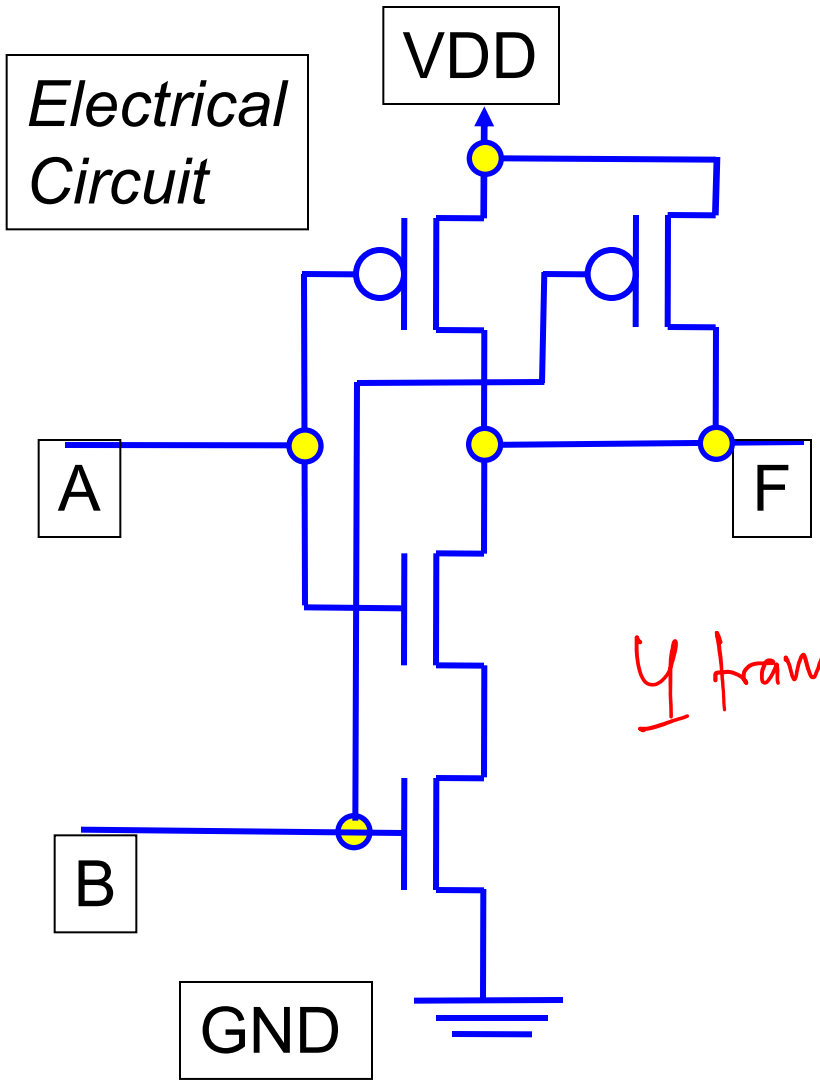
| Truth Table | | |
|---|---|---|
| A | B | F |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

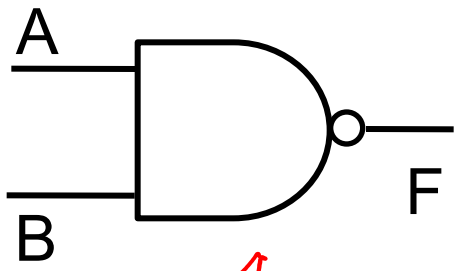
*Symbol*

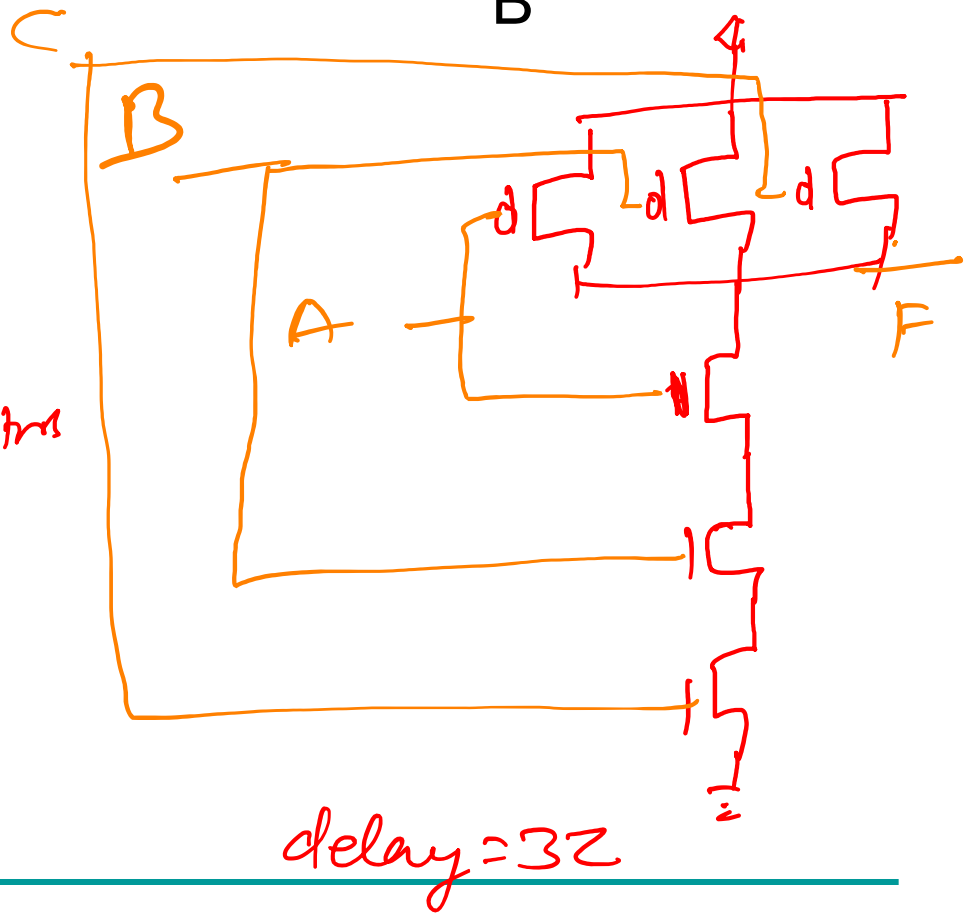



delay = 2τ.

# CMOS Logic Gate: NAND

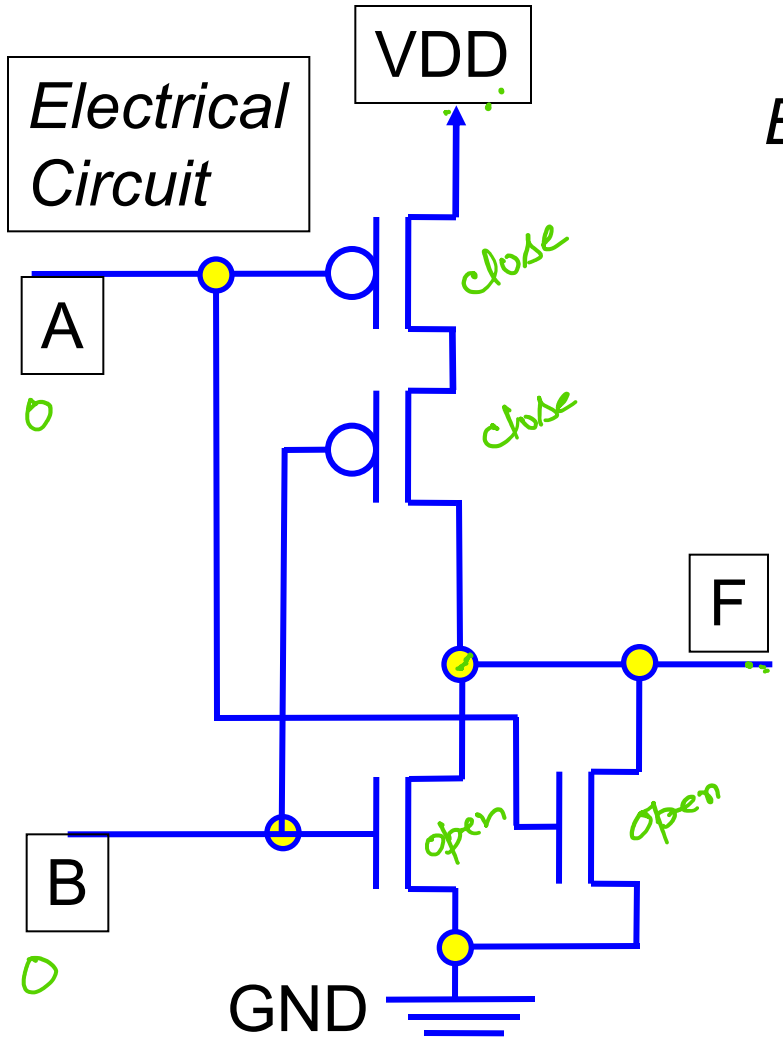



4 transistors

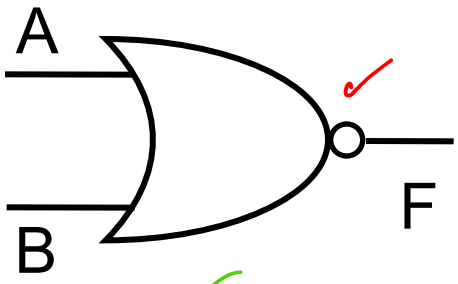
# CMOS Logic Gate: NOR



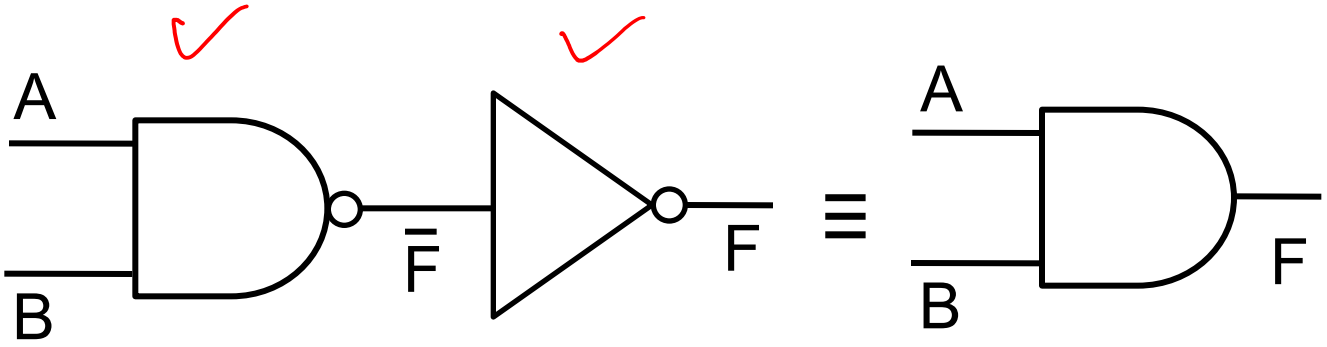

*Boolean Function*

| Truth Table | | |
|---|---|---|
| A | B | F |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

*Symbol*



# CMOS Logic Gate: AND

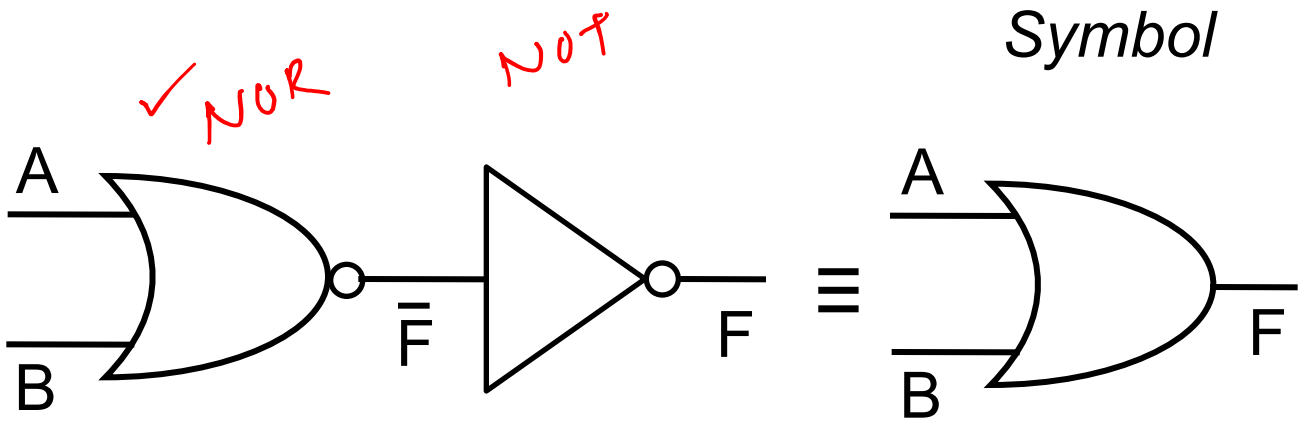



*Symbol*

*Boolean Function*

| Truth Table | | |
|---|---|---|
| A | B | F |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

# CMOS Logic Gate: OR




*Symbol*

*Boolean Function*

| Truth Table | | |
|---|---|---|
| A | B | F |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

NOR-NOR
NAND-AND

# CMOS Gates

| Logic function | Number of transistors | |
|---|---|---|
| | 1 or 2 inputs | N inputs |
| NOT | 2 | - |
| AND | 6 | 2N + 2 |
| OR | 6 | 2N + 2 |
| NAND | 4 | 2N |
| NOR | 4 | 2N |

# Logic Minimization

Performance

Inverter = 1

AND = 2N

OR = 2N

NAND = 2N

NOR = 2N

delay $\propto$ N

Total delay

# Thank You