

Logic: Implementation

Virendra Singh

Professor

Computer Architecture and Dependable Systems Lab

Department of Computer Science & Engineering, and

Department of Electrical Engineering

Indian Institute of Technology Bombay

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

E-mail: viren@{cse, ee}.iitb.ac.in

CS-230: Digital Logic Design & Computer Architecture

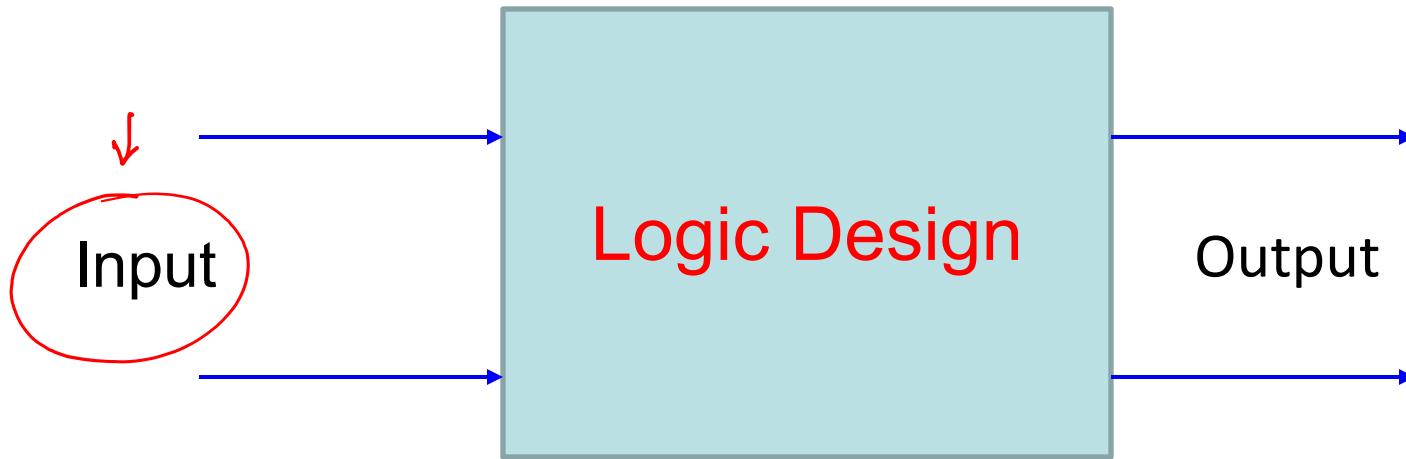


Lecture 4 (11 January 2022)

CADSL

Digital System

Truth Table :



$$\text{Output} = f(\text{Input})$$



canonical form Logic Expressions

Truth Table ✓

$n=3$

X Y Z	F
0 0 0	0 1
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

✓

$2^3 = 8$

Logic Expression ✓

$$F = \overline{X} \cdot \overline{Y} \cdot \underline{Z} + X \cdot \overline{Y} \cdot \overline{Z} + X \cdot \overline{Y} \cdot Z \\ + X \cdot Y \cdot \overline{Z} + X \cdot Y \cdot Z$$



2^{2n}

2^n

- Logic expressions, truth tables describe the same function!
- Truth tables are unique; expressions are not. This gives flexibility in implementing functions.

Optimization \Rightarrow minimise cost



How Many Logic Functions?

- Output column of truth table has length 2^n for n input variables.
- It can be arranged in 2^{2^n} ways for n variables.
- Example: $n = 1$, single variable.

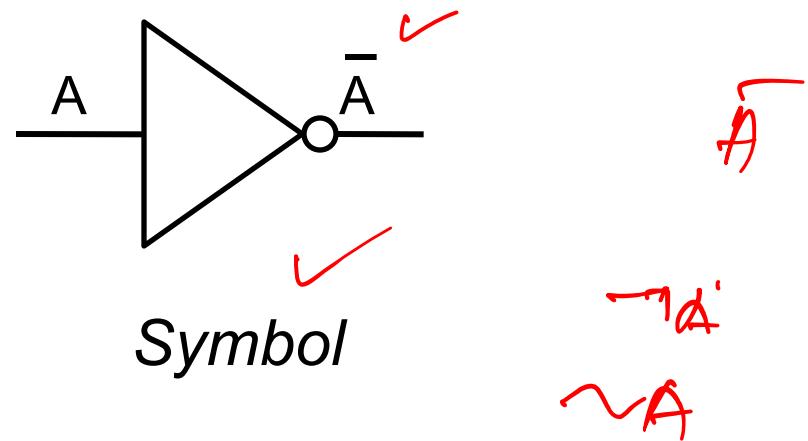
2 choices
 2^{2^n}
 $A \oplus F(A)$
 $2^2 = 4$

Input	Output functions			
A	F1(A)	F2(A)	F3(A)	F4(A)
0.	0]	0 :	1 :	1 .
1.	0]	1 :	0 :	1 .



NOT Gate

Truth Table	
A	\bar{A}
0	1
1	0

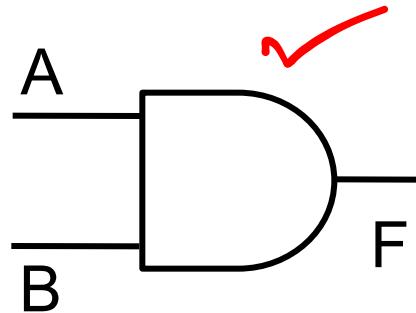


Logic Function



AND Gate

Symbol



Logic Function

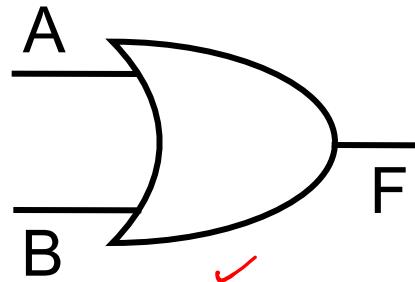
Truth Table

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1



OR Gate

Symbol



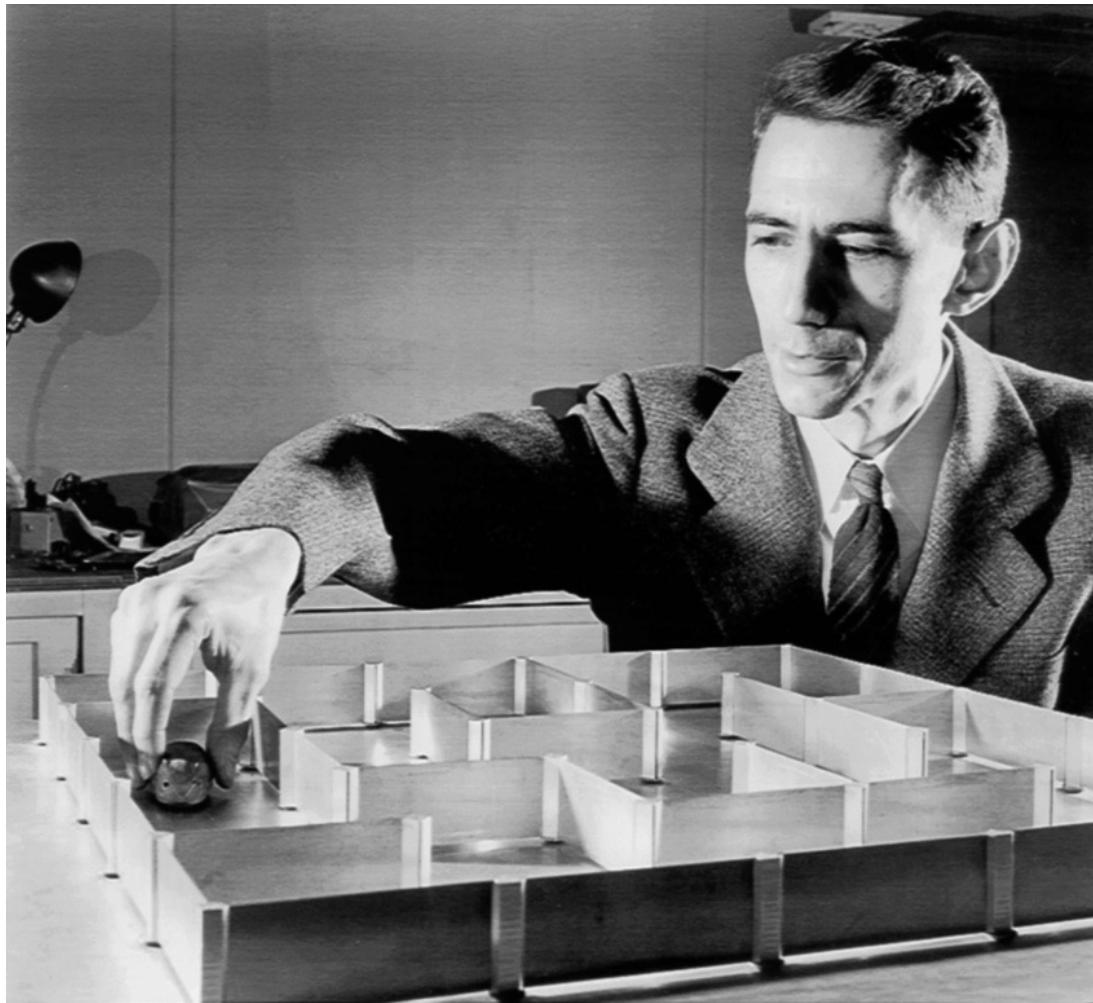
Logic Function

Truth Table

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1



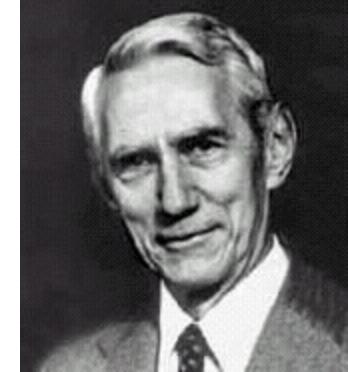
Claude E. Shannon (1916-2001)



http://www.kugelbahn.ch/sesam_e.htm



Shannon's Legacy



- A *Symbolic Analysis of Relay and Switching Circuits*,
Master's Thesis, MIT, 1940. Perhaps the most
influential master's thesis of the 20th century.
- *An Algebra for Theoretical Genetics*, PhD Thesis, MIT,
1940.
- Founded the field of Information Theory.
- C. E. Shannon and W. Weaver, *The Mathematical
Theory of Communication*, University of Illinois Press,
1949. A “must read.”



Logic Function Implementation

- Using Switches

- For inputs:

- logic 1 is switch closed
 - logic 0 is switch open

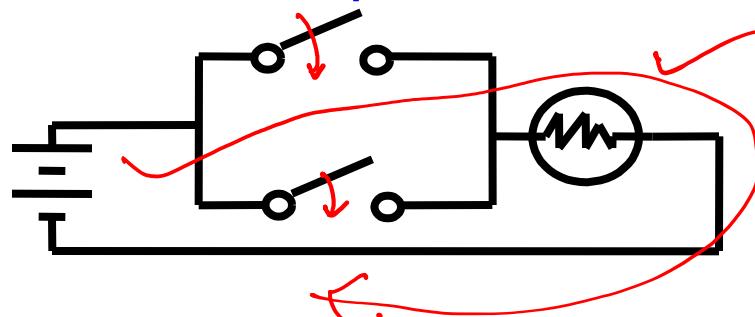
- For outputs:

- logic 1 is light on
 - logic 0 is light off.

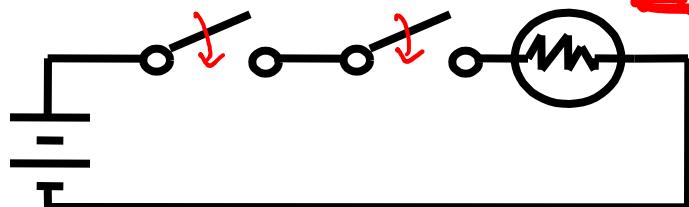
- NOT uses a switch such
that:

- logic 1 is switch open
 - logic 0 is switch closed

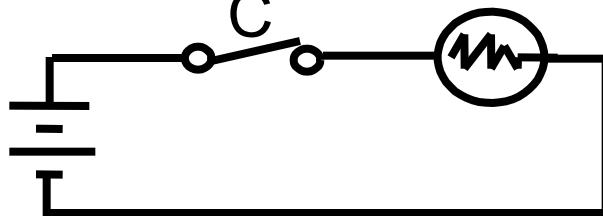
Switches in parallel => OR



Switches in series => AND



Normally-closed switch => NOT



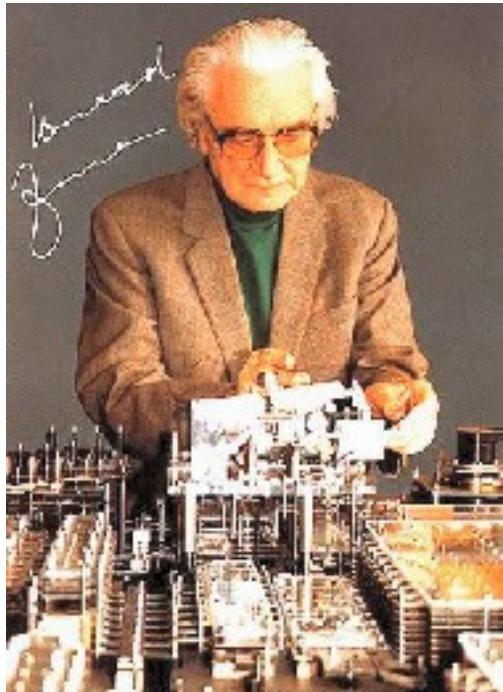
Switching Devices

- Electromechanical relays (1940s)
- Vacuum tubes (1950s)
- Bipolar transistors (1960 - 1980) ↗ BJT
- Field effect transistors (1980 -) ✓
- Integrated circuits (1970 -) ↘

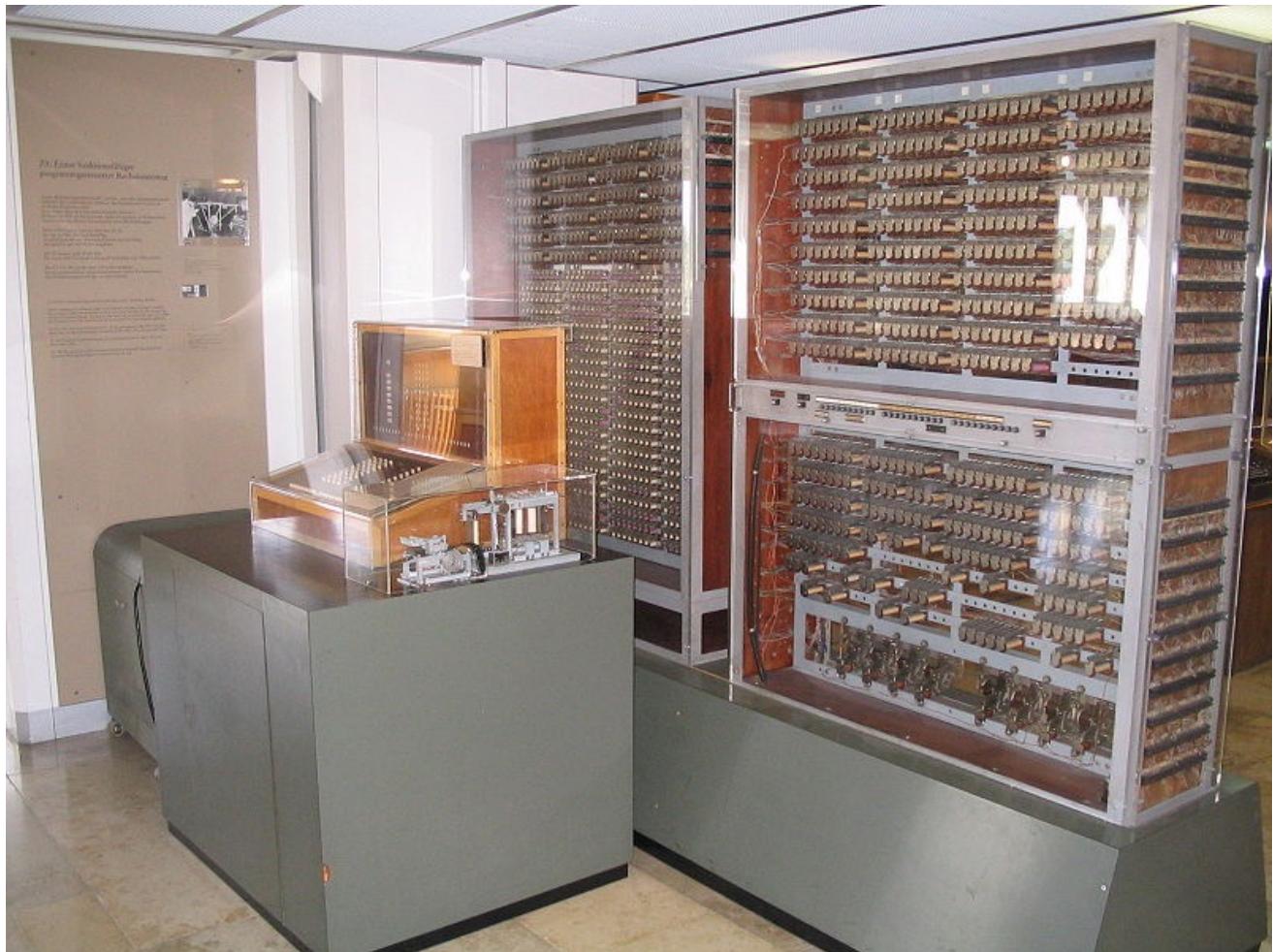


Relay Computers

Conrad Zuse (1910-1995)



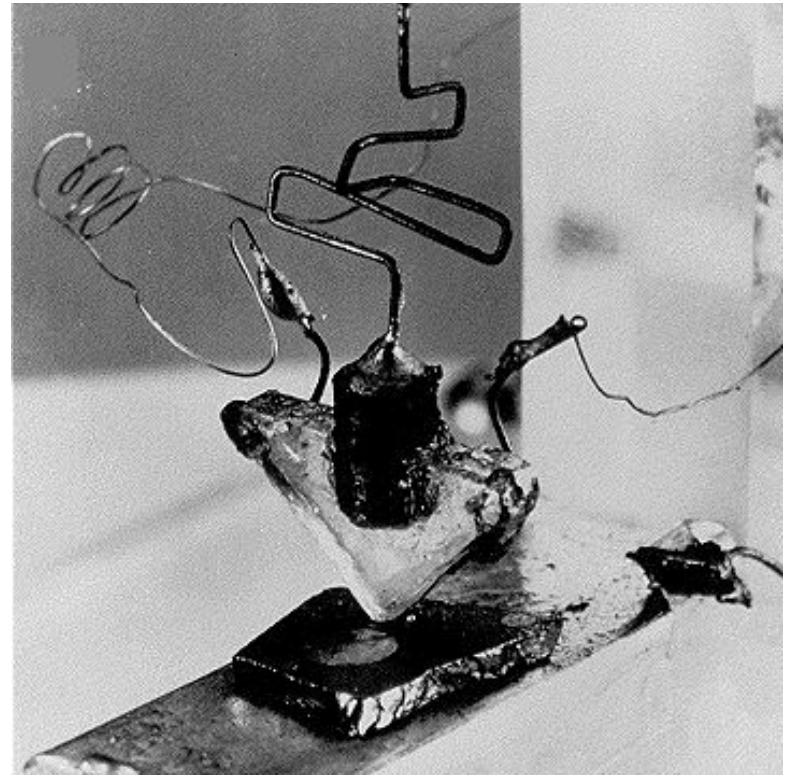
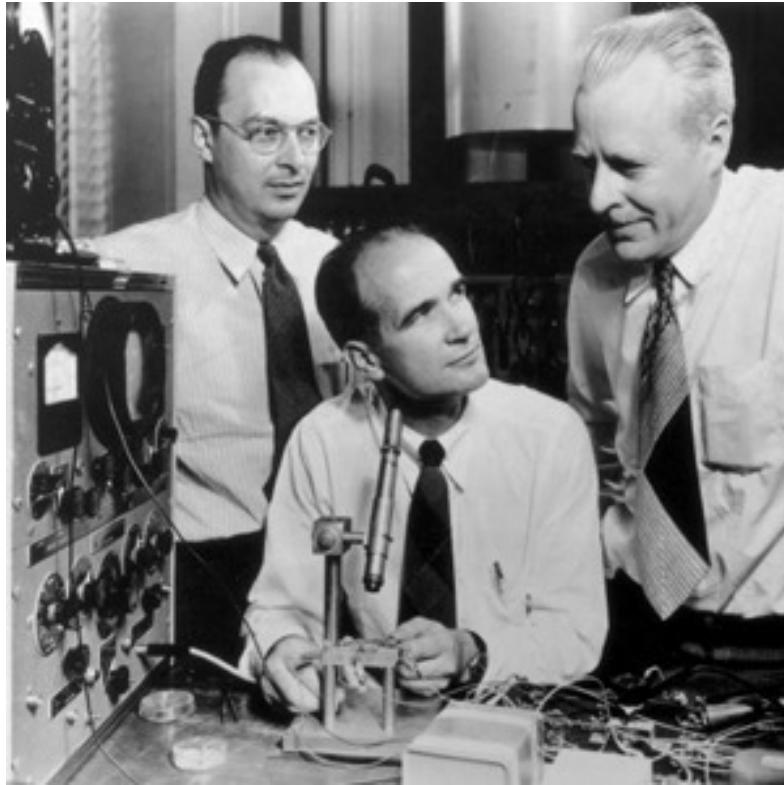
Z1 (1938)



Z3 (1941)

Transistor, 1948

The thinker, the tinkerer, the visionary and the transistor
John Bardeen, Walter Brattain, William Shockley
Nobel Prize, 1956

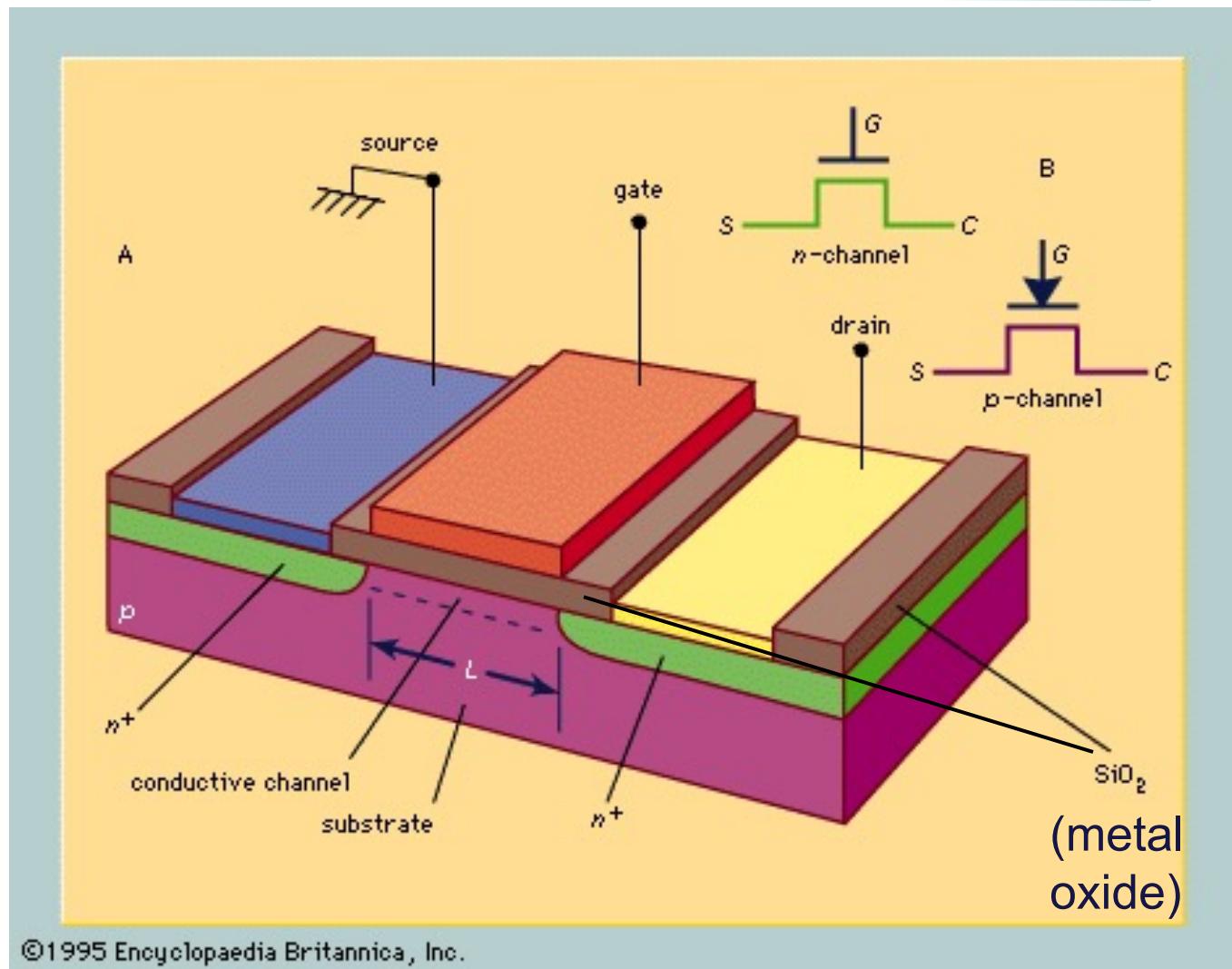


Bell Laboratories, Murray Hill, New Jersey



Field Effect Transistor (FET)

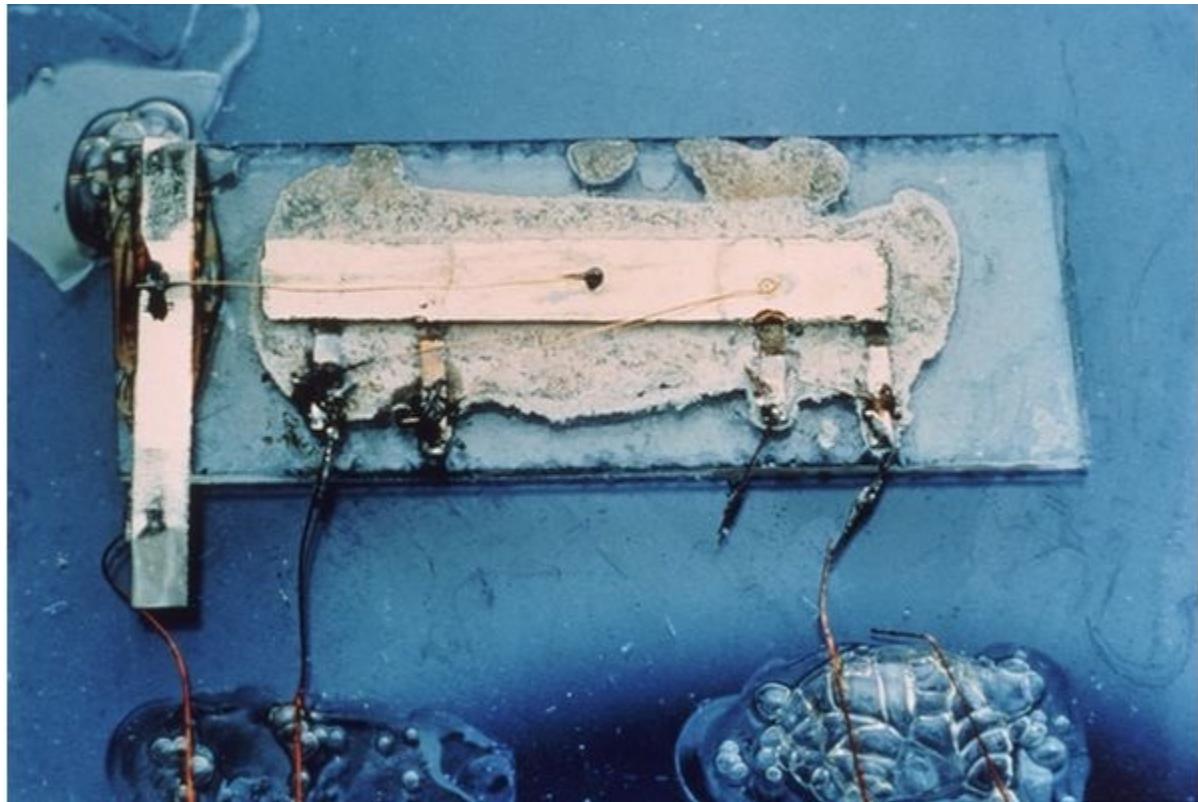
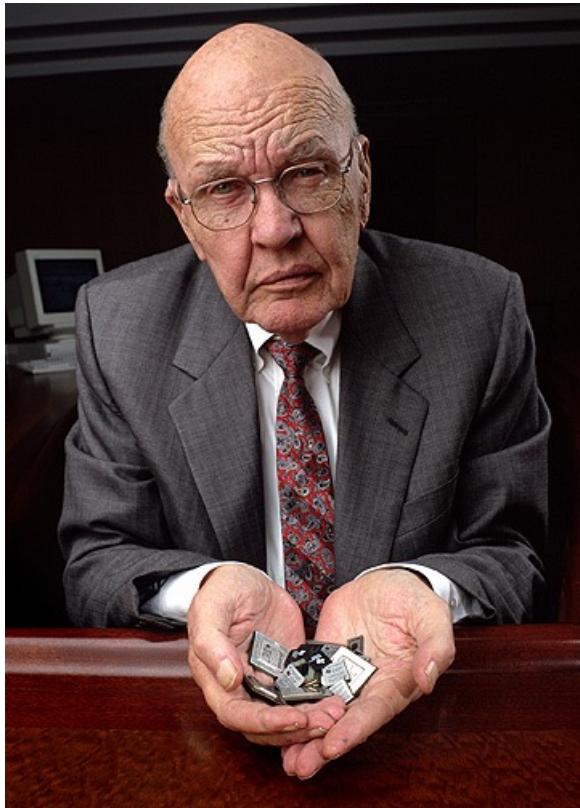
a.k.a.
metal oxide
semiconductor
(MOS) FET.



©1995 Encyclopaedia Britannica, Inc.



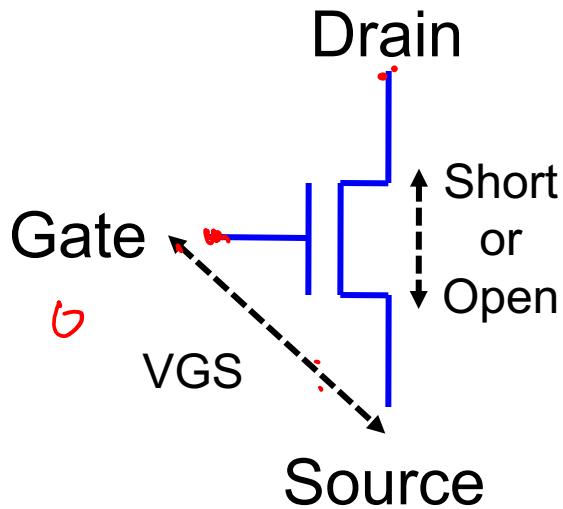
Integrated Circuit (1958)



Jack Kilby (1923-2005), Nobel Prize, 2000

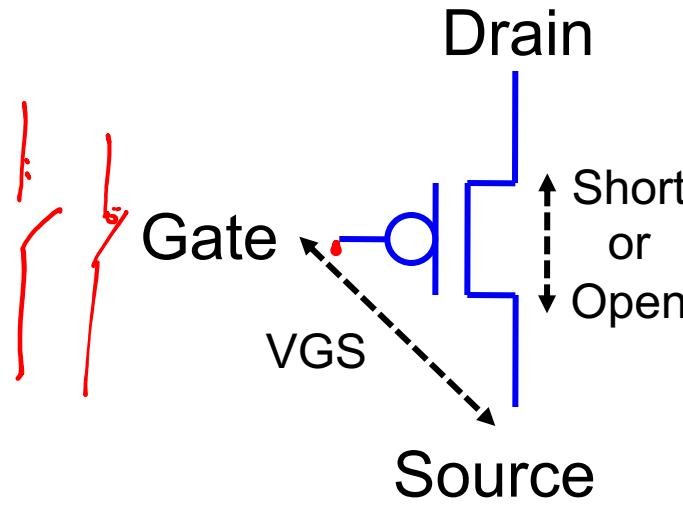


MOSFET (Metal Oxide Semiconductor Field Effect Transistor)



NMOSFET

$V_{GS} = 0$, open
 $V_{GS} = \text{high}$, short



PMOSFET

$V_{GS} = 0$, short
 $V_{GS} = \text{high}$, open

Reference:

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



NMOSFET NOT Gate (Early Design)

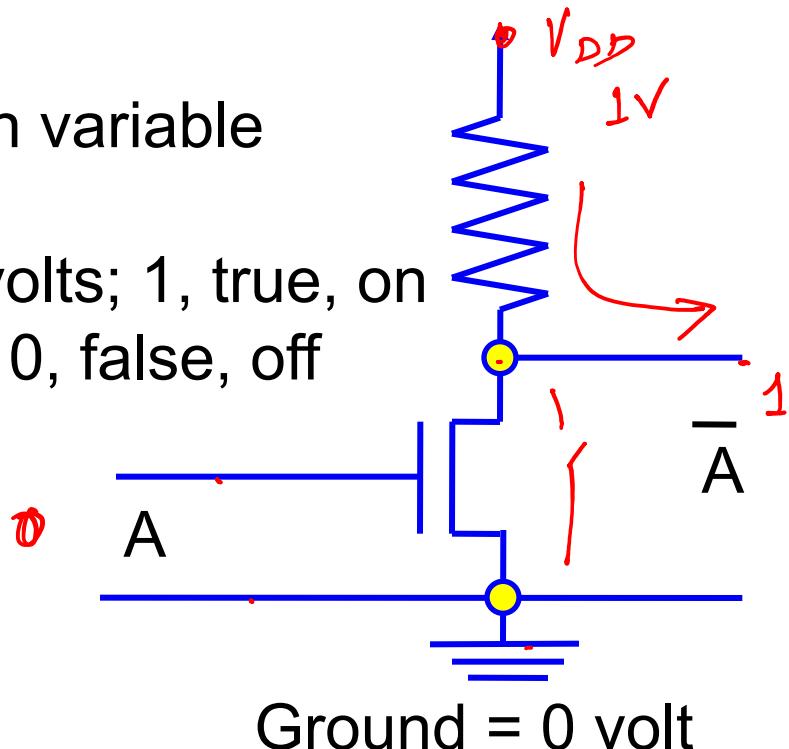
1.8

Power supply
VDD volts
w.r.t. ground

A: Boolean variable

A = VDD volts; 1, true, on

A = 0 volt; 0, false, off



Problem: When $A = 1$, current leakage causes power dissipation.

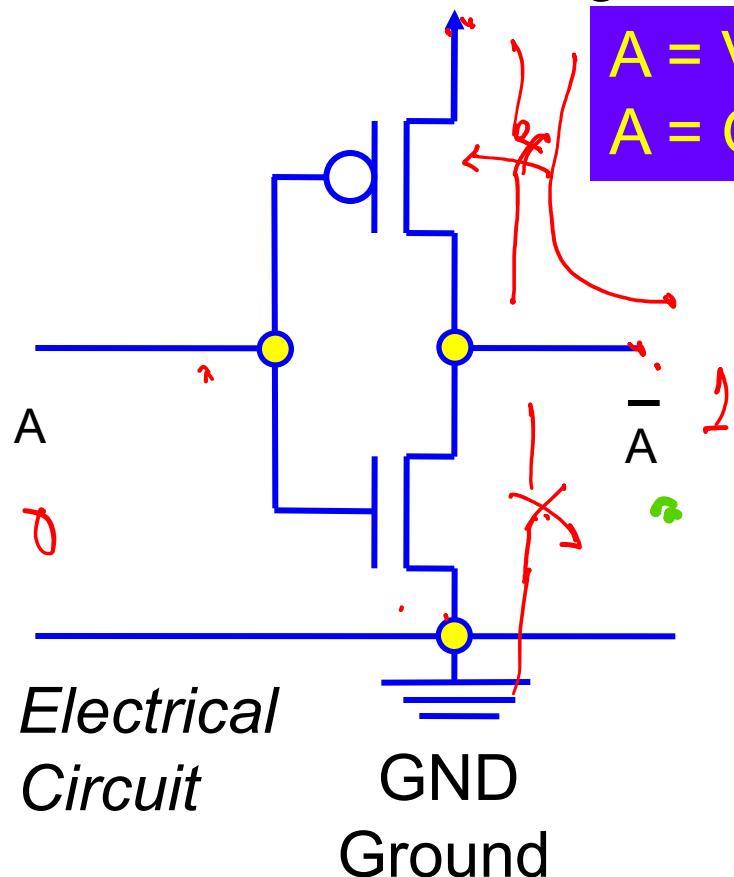
Solution: Complementary MOS design proposed by

F. M. Wanlass and C.-T. Sah,
“Nanowatt Logic Using Field-
Effect Metal-Oxide
Semiconductor Triodes,”
*International Solid State Circuits
Conference Digest of Technical
Papers*, Feb 20, 1963, pp. 32-33.

CMOS NOT Gate (Modern Design)

Power supply

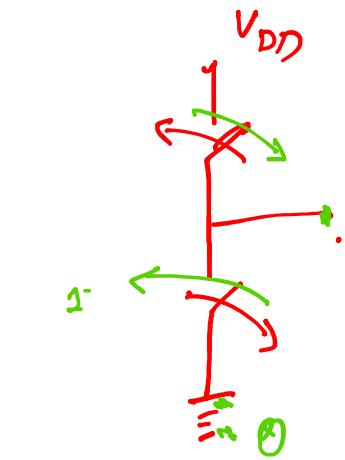
$V_{DD} = 1$ volt; voltage depends on technology.



$A = V_{DD} = 1$ volt is state “1”
 $A = GND = 0$ volt is state “0”

Truth Table	
A	\bar{A}
0	1
1	0

Boolean Function



Symbol

\overline{A}

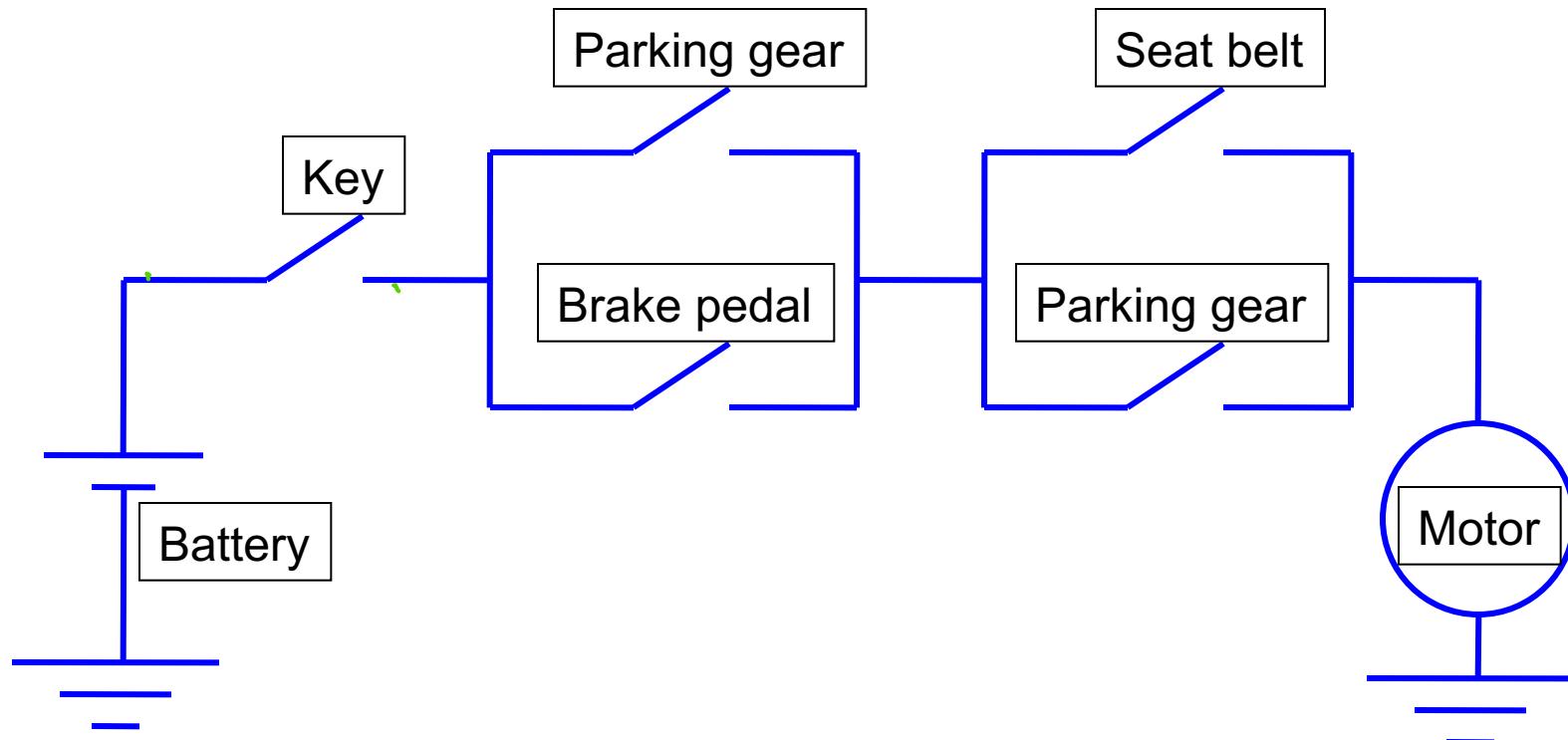


Example: Automobile Ignition

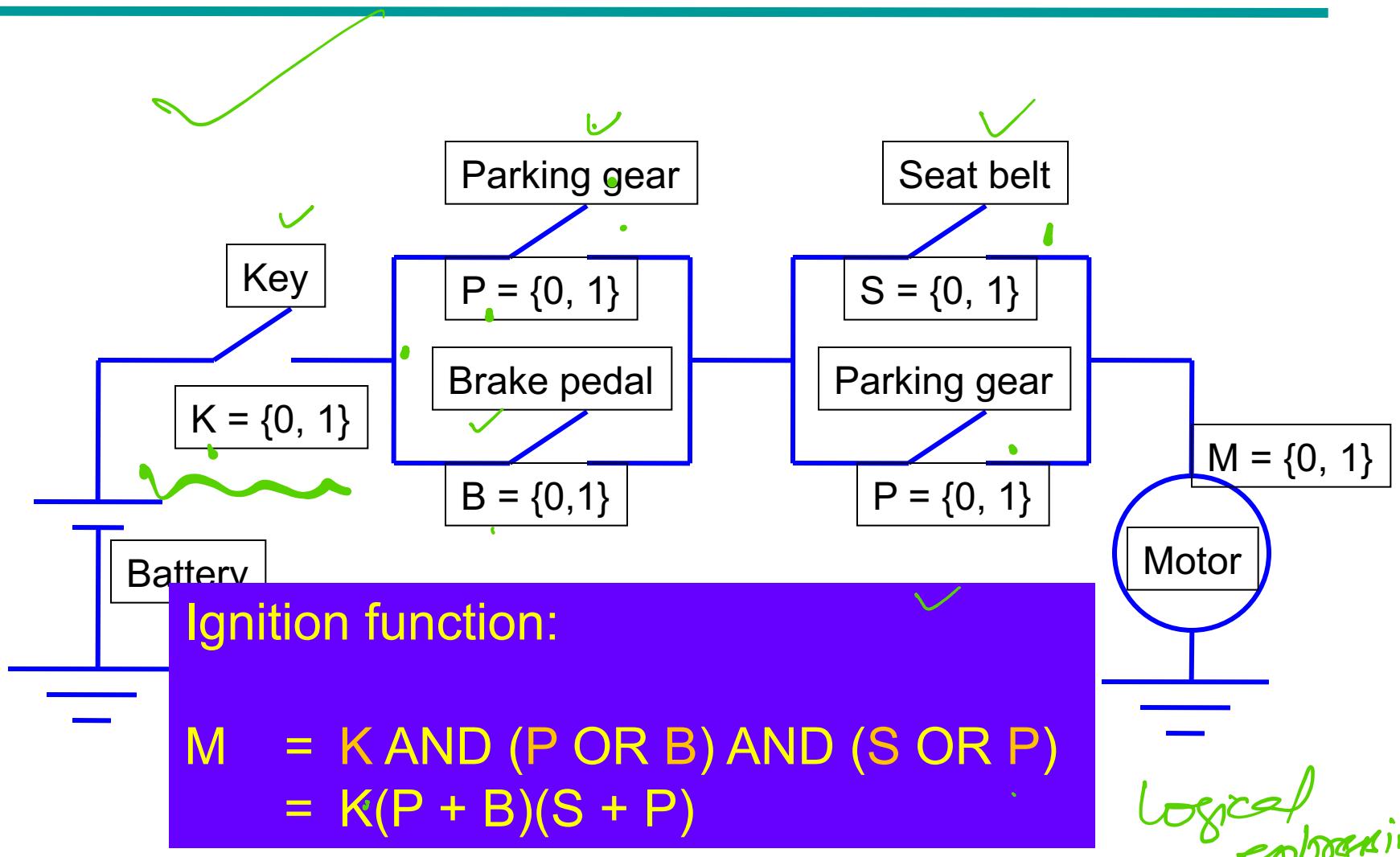
- Engine turns on when
- Ignition key is applied AND
 - Car is in parking gear OR
 - Brake pedal is on
- AND
 - Seat belt is fastened OR
 - Car is in parking gear



Switching logic

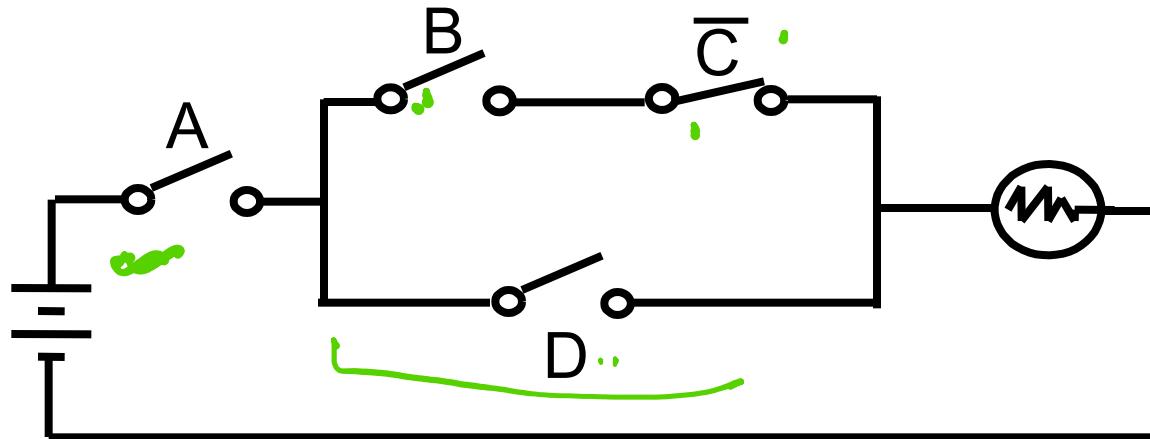


Define Variables



Logic Function Implementation

- Example: Logic Using Switches



- Light is on ($L = 1$) for \checkmark
$$L(A, B, C, D) = \underline{A} ((B C') + D) = \underline{\underline{A}} B C' + \underline{\underline{A}} D$$
 and off ($L = 0$), otherwise.
- Useful model for relay circuits and for CMOS gate circuits, the foundation of current digital logic technology

Logic Expression

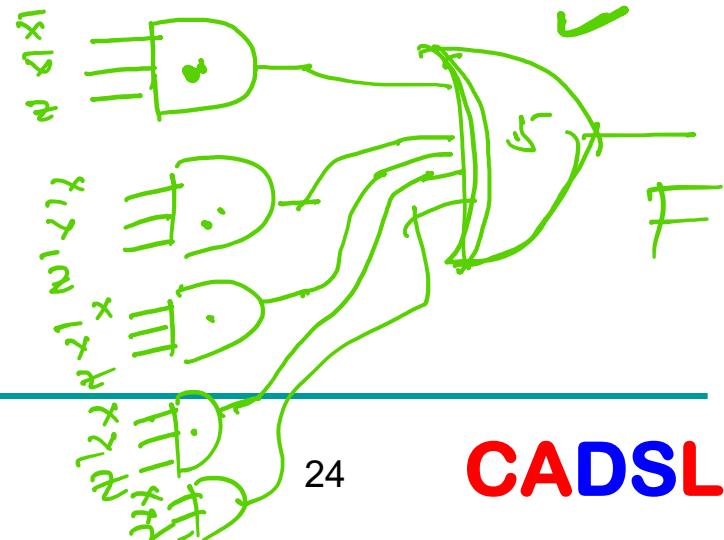
$$15+5+3 \\ 23$$

Truth Table

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

Logic Expression

$$\Rightarrow F = \underline{\bar{X} \cdot \bar{Y} \cdot Z} + \underline{X \cdot \bar{Y} \cdot \bar{Z}} + \underline{X \cdot \bar{Y} \cdot Z}$$
$$\quad \quad \quad \underline{X \cdot Y \cdot \bar{Z}} + \underline{X \cdot Y \cdot Z}$$



Logic Expressions

Truth Table

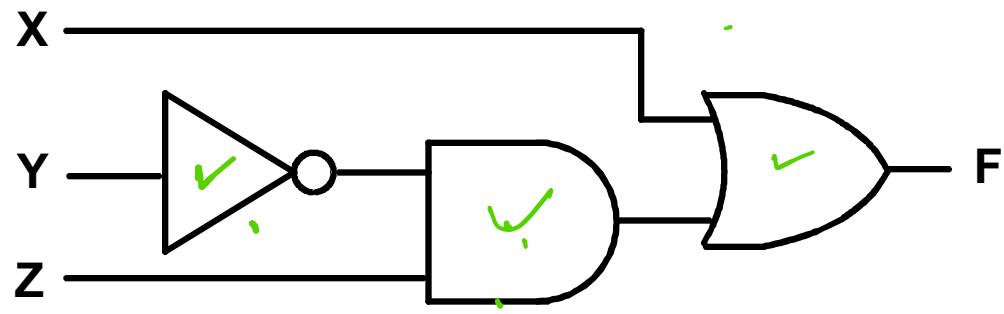
X Y Z	F
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	1
1 1 0	1
1 1 1	1

Equation

$$F = X + \overline{Y} \cdot Z$$

SOP

Logic Diagram



$$2+2\times 1=5$$



Digital Logic Design

- Express input output relationship using Truth table
- Generate the logical expression by disjunction (OR) terms (conjunction of variables – AND) where system evaluates to true
- Replace all operators by the logic gates
- Replace logic gates by its transistor level circuit

Switches



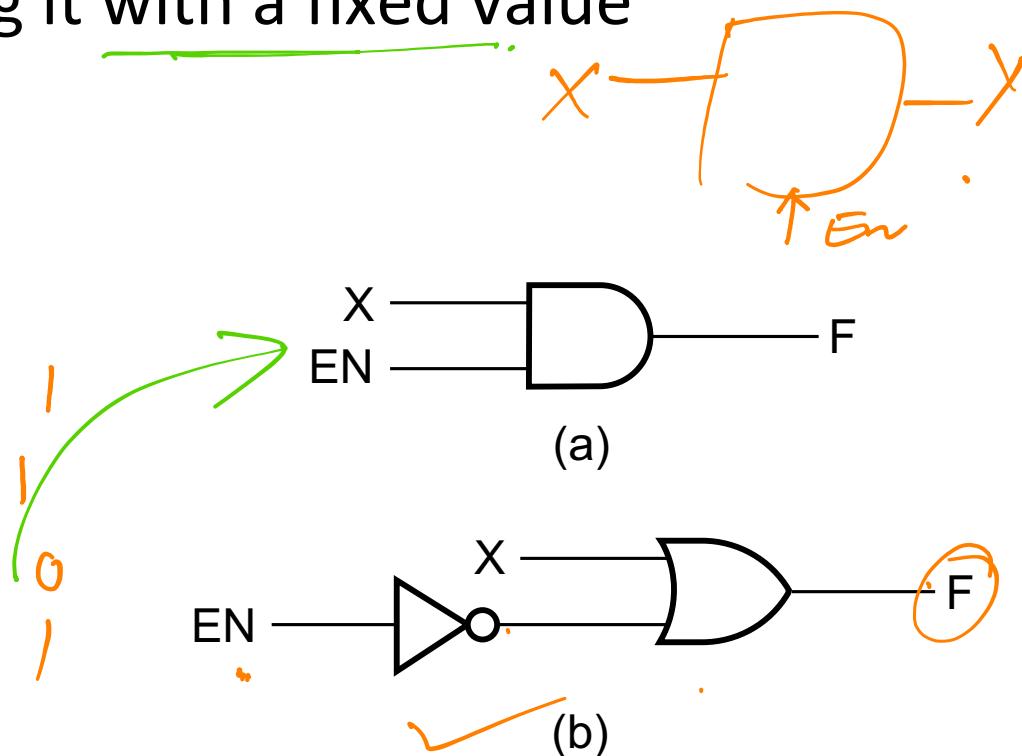
Common Functions



Enabling Function

- *Enabling* permits an input signal to pass through to an output
- *Disabling* blocks an input signal from passing through to an output, replacing it with a fixed value
- When disabled, 0 output
- When disabled, 1 output

EN	X	Y
1	0	0
1	1	0
0	1	0
0	1	1



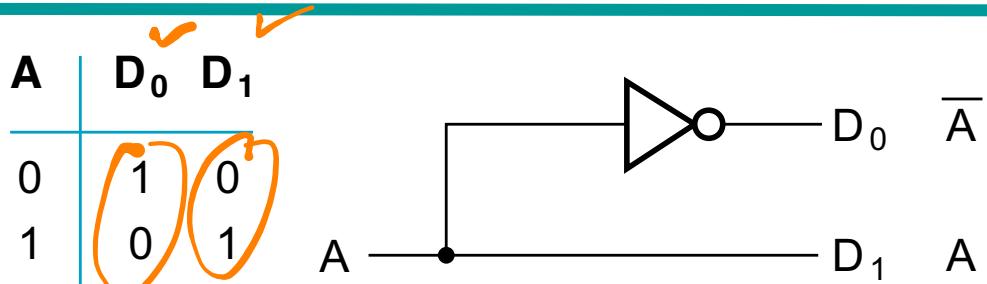
Decoding Function

- Decoding - the
 - Conversion of n -bit input to m -bit output
 - Given $n \leq m \leq 2^n$
- Circuits that perform decoding are called *decoders*
 - Called n -to- m line decoders, where $m \leq 2^n$, and
 - Generate 2^n (or fewer) 1's in output for the n input variables

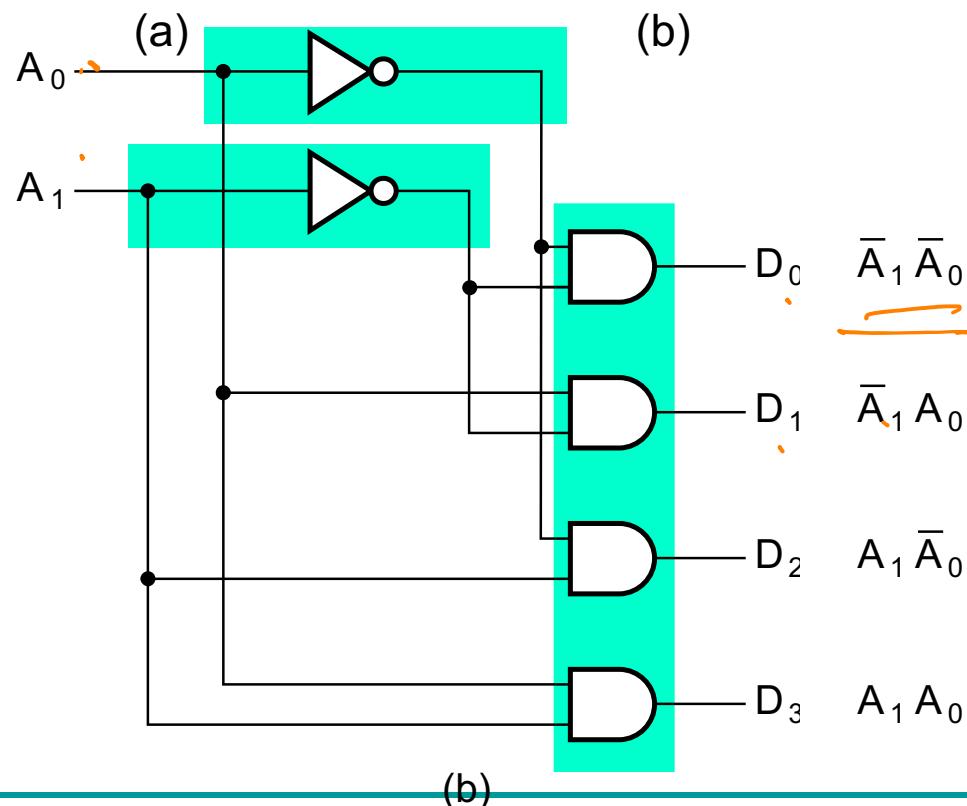
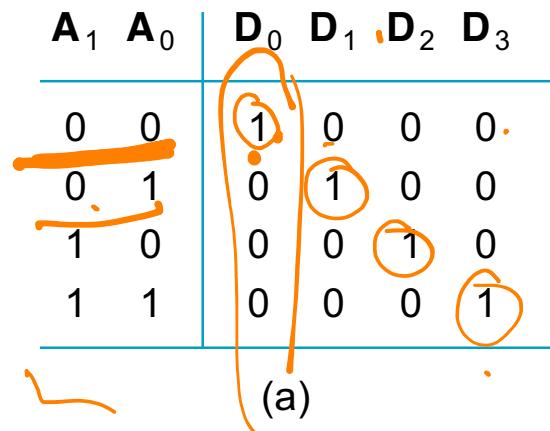


Decoder

- 1-to-2-Line Decoder



- 2-to-4-Line Decoder



Encoding Function

- Encoding - the opposite of decoding
 - Conversion of m -bit input to n -bit output
- Circuits that perform encoding are called *encoders*
 - An encoder has 2^n (or fewer) input lines and n output lines which generate the binary code corresponding to the input values
 - Typically, an encoder converts a code containing exactly one bit that is 1 to a binary code corresponding to the position in which the 1 appears.



Encoder

- A decimal-to-BCD encoder
 - Inputs: 10 bits corresponding to decimal digits 0 through 9, (D_0, \dots, D_9)
 - Outputs: 4 bits with BCD codes
 - Function: If input bit D_i is a 1, then the output (A_3, A_2, A_1, A_0) is the BCD code for i ,
 - The truth table could be formed, but alternatively, the equations for each of the four outputs can be obtained directly.
- 



Encoder

- Input D_i is a term in equation A_j if bit A_j is 1 in the binary value for i .

- Equations:

$$A_3 = D_8 + D_9$$

$$A_2 = D_4 + D_5 + D_6 + D_7$$

$$A_1 = D_2 + D_3 + D_6 + D_7$$

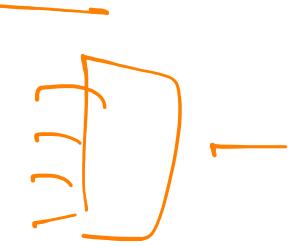
$$A_0 = D_1 + D_3 + D_5 + D_7 + D_9$$

D_0	D_1	D_2	D_3
0000			✓
0001			✓
0010			✓
0011			
0100			
0101			
0110			
0111			
1000			○
1001			○



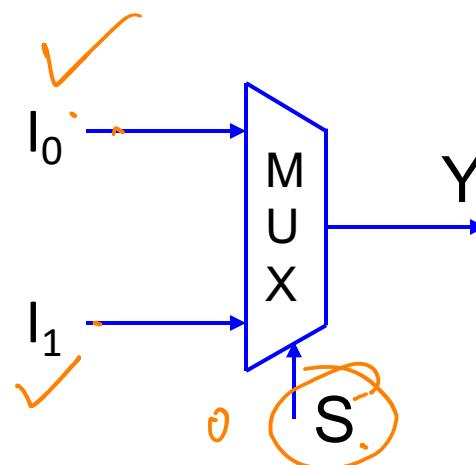
Selection Function

- Selecting of data or information is a critical function in digital systems and computers
- Circuits that perform selecting have:
 - A set of information inputs from which the selection is made
 - A single output ✓
 - A set of control lines for making the selection
- Logic circuits that perform selecting are called **multiplexers**



Multiplexers

- A **multiplexer** selects one input line and transfers it to output
 - n control inputs ($S_{n-1}, \dots S_0$) called *selection inputs*
 - $m \leq 2^n$ information inputs ($I_{2^n-1}, \dots I_0$)
 - output Y



2-to-1-Line Multiplexer

- Since $2 = 2^1$, $n = 1$
- The single selection variable S has two values:
 - $S = 0$ selects input I_0
 - $S = 1$ selects input I_1
- Truth Table

S	I_0	I_1	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

✓ $Y = I_0 \cdot \bar{S} + S \cdot I_1$

• Logic expression

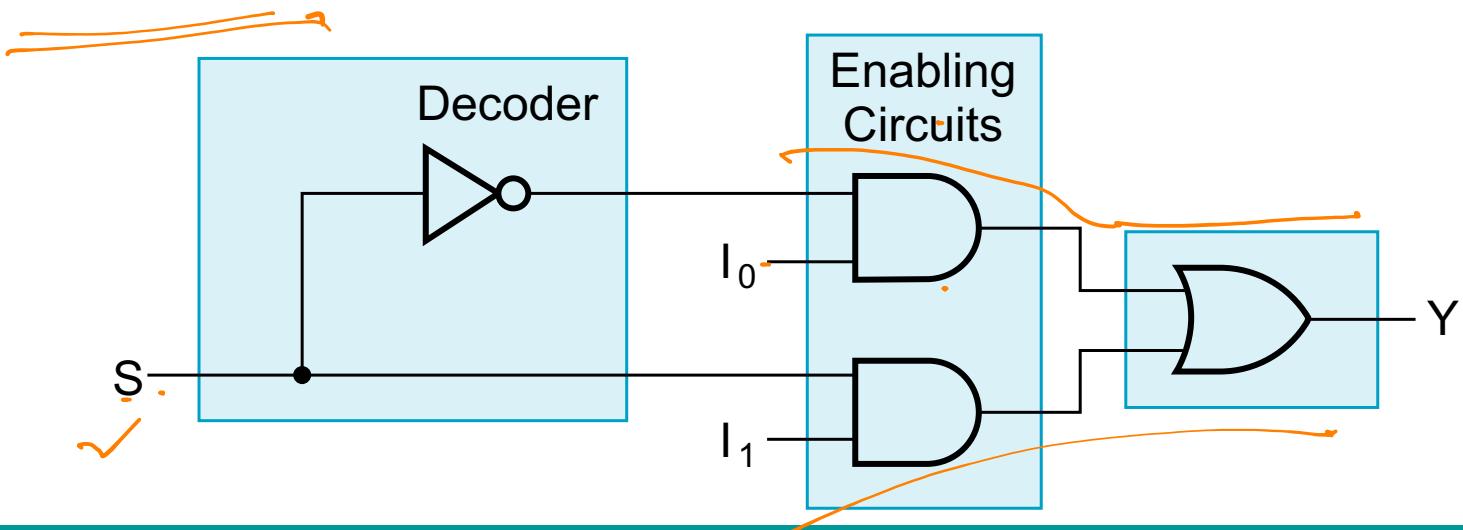
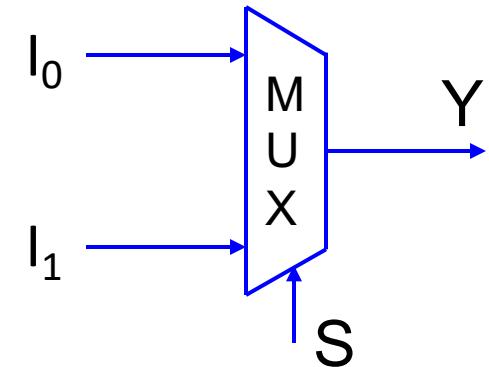
$$Y = \bar{S} \cdot I_0 \cdot \bar{I}_1 + \bar{S} \cdot I_0 \cdot I_1 + S \cdot \bar{I}_0 \cdot I_1 + S \cdot I_0 \cdot I_1$$



2-to-1-Line Multiplexer

- The single selection variable S has two values:
 - $S = 0$ selects input I_0
 - $S = 1$ selects input I_1
- The logic equation:

$$Y = I_0 \bar{S} + S \cdot I_1$$



Thank You

