

BAN HỌC TẬP CÔNG NGHỆ PHẦN MỀM

TRAINING GIỮA KỲ HỌC KỲ I NĂM HỌC 2022 – 2023



Sharing is learning



 **BAN HỌC TẬP**

Khoa Công nghệ Phần mềm

Trường Đại học Công nghệ Thông tin

Đại học Quốc gia thành phố Hồ Chí Minh

 **CONTACT**

bht.cnpm.uit@gmail.com

fb.com/bhtcnpm

fb.com/groups/bht.cnpm.uit

TRAINING

TỔ CHỨC - CẤU TRÚC MÁY TÍNH II

- ⌚ **Thời gian:** 10:00 thứ 4 ngày 07/12/2022
- 📍 **Địa điểm:** Giảng đường 3 – Tòa nhà A
- 👤 **Trainers:** Huỳnh Lê Đan Linh – KTMP2022.2
Nguyễn Hoài Như – KTPM2022.2



Sharing is learning

LƯU Ý

- I. Buổi offline sẽ tóm tắt một số vấn đề về lý thuyết và trọng tâm vào kiến thức để giải bài tập. Các điểm lý thuyết học thuộc sẽ chỉ nói sơ lược.
- II. Cuối mỗi phần sẽ có thời gian đặt câu hỏi trong phạm vi phần vừa ôn tập. Nếu chưa thể trả lời được, đáp án sẽ được tìm hiểu và public vào buổi online.
- III. Chúc các bạn có một buổi ôn tập hiệu quả.



TABLE OF CONTENTS

- I. TỔNG QUAN VỀ MÁY TÍNH**
- II. BIỂU DIỄN THÔNG TIN TRONG MÁY TÍNH**
- III. ĐẠI SỐ BOOLEAN**
- IV. MẠCH SỐ**
- V. MẠCH SỐ TRONG MÁY TÍNH**

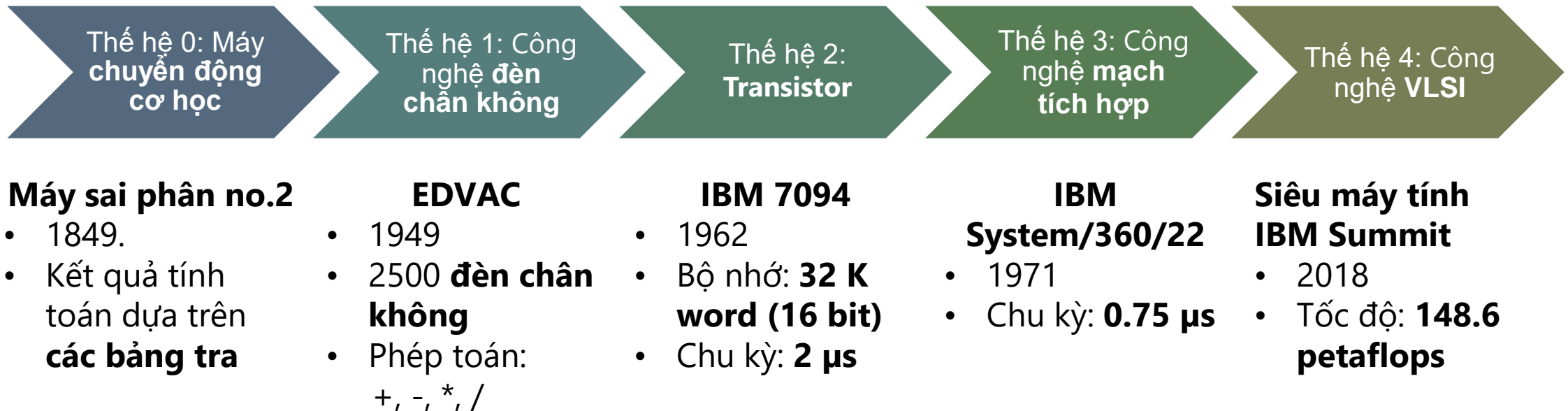


I. TỔNG QUAN VỀ MÁY TÍNH

A. LỊCH SỬ MÁY TÍNH

MacBook Pro 15' 2019

- Tốc độ: **2.6 Ghz**



I. TỔNG QUAN VỀ MÁY TÍNH

B. PHÂN LOẠI MÁY TÍNH

- Máy tính cá nhân (Personal computers)
 - Kích thước: nhỏ gọn
 - Tốc độ: lên đến 238,310 MIPS ở 3.0 GHz
 - Khả năng xử lý: Đa dụng cho các ứng dụng văn phòng, học tập, giải trí.
 - Ví dụ: Máy tính để bàn, Máy tính xách tay



I. TỔNG QUAN VỀ MÁY TÍNH

B. PHÂN LOẠI MÁY TÍNH

- Máy chủ (Servers)
 - Kích thước: lớn
 - Tốc độ: lên đến 148.6 petaflops
 - Khả năng xử lý: tính toán với tốc độ siêu nhanh, độ chính xác cực lớn.
 - Khả năng lưu trữ dữ liệu: cực lớn.



Sharing is learning

I. TỔNG QUAN VỀ MÁY TÍNH

B. PHÂN LOẠI MÁY TÍNH

- Máy chủ (Servers)
 - **Low-end servers:**
 - ✓ Ứng dụng **lưu trữ, doanh nghiệp nhỏ, dịch vụ web.**
 - ✓ Chi phí khoảng 1000\$.



I. TỔNG QUAN VỀ MÁY TÍNH

B. PHÂN LOẠI MÁY TÍNH

- Máy chủ (Servers)
 - **Supercomputers:**
 - ✓ Tính toán **kỹ thuật và khoa học phức tạp, hiệu năng cao nhất.**
 - ✓ **Hàng trăm - hàng ngàn** bộ xử lý, bộ nhớ kích cỡ **gigabytes - terabytes**
 - ✓ Khả năng lưu trữ dữ liệu **terabytes đến petabytes.**
 - ✓ Chi phí hàng triệu đến hàng trăm triệu đôla.



I. TỔNG QUAN VỀ MÁY TÍNH

B. PHÂN LOẠI MÁY TÍNH

- Máy chủ (Servers)
 - **Datacenter:**
 - ✓ Được sử dụng bởi những công ty như eBay, Google
 - ✓ **Hàng ngàn** bộ xử lý, bộ nhớ hàng **terabytes**
 - ✓ Khả năng lưu trữ hàng **petabytes**.
 - ✓ Datacenter thường được xem như là các cụm máy tính lớn.



I. TỔNG QUAN VỀ MÁY TÍNH

B. PHÂN LOẠI MÁY TÍNH

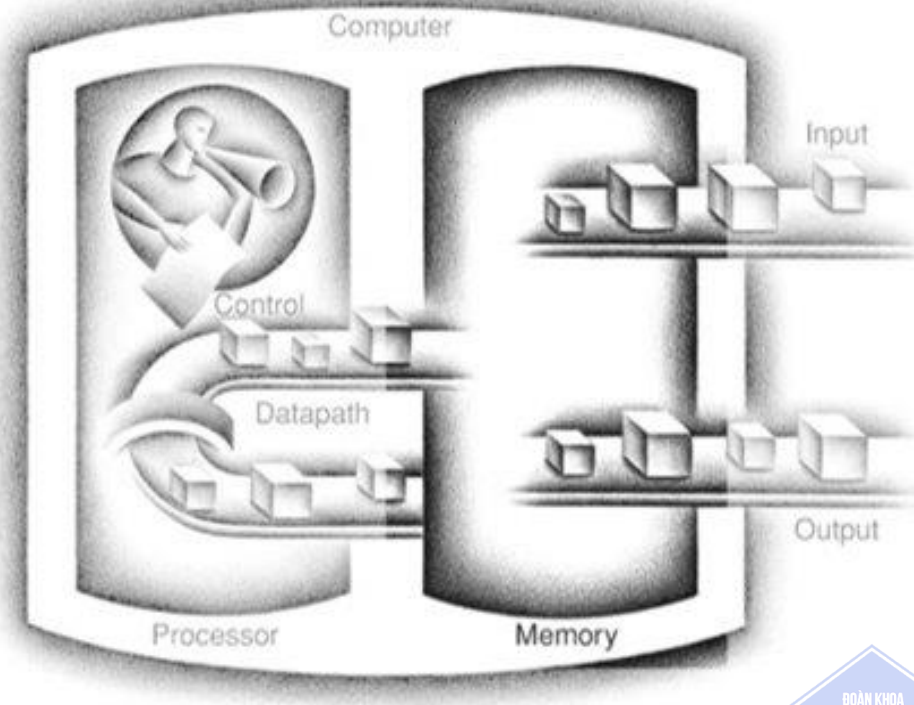
- Máy tính nhúng (Embedded computers)
 - Kích thước: nhỏ gọn, tích hợp bên trong một thiết bị
 - Tốc độ: Không cần cao (thường dưới 400 Mhz).
 - Khả năng xử lý: Được tối ưu cho một số chức năng cụ thể.
 - Ví dụ: Raspberry Pi



I. TỔNG QUAN VỀ MÁY TÍNH

C. THÀNH PHẦN TRONG MÁY TÍNH

- Bộ xử lý (Processor)
 - Khối đường dữ liệu (Datapath) - Tính toán
 - Khối điều khiển (Controller)
 - ✓ Điều khiển Datapath, Bộ nhớ và I/O
- Bộ nhớ (Memory)
- Các thiết bị nhập /xuất (I/O)



I. TỔNG QUAN VỀ MÁY TÍNH

C. THÀNH PHẦN TRONG MÁY TÍNH

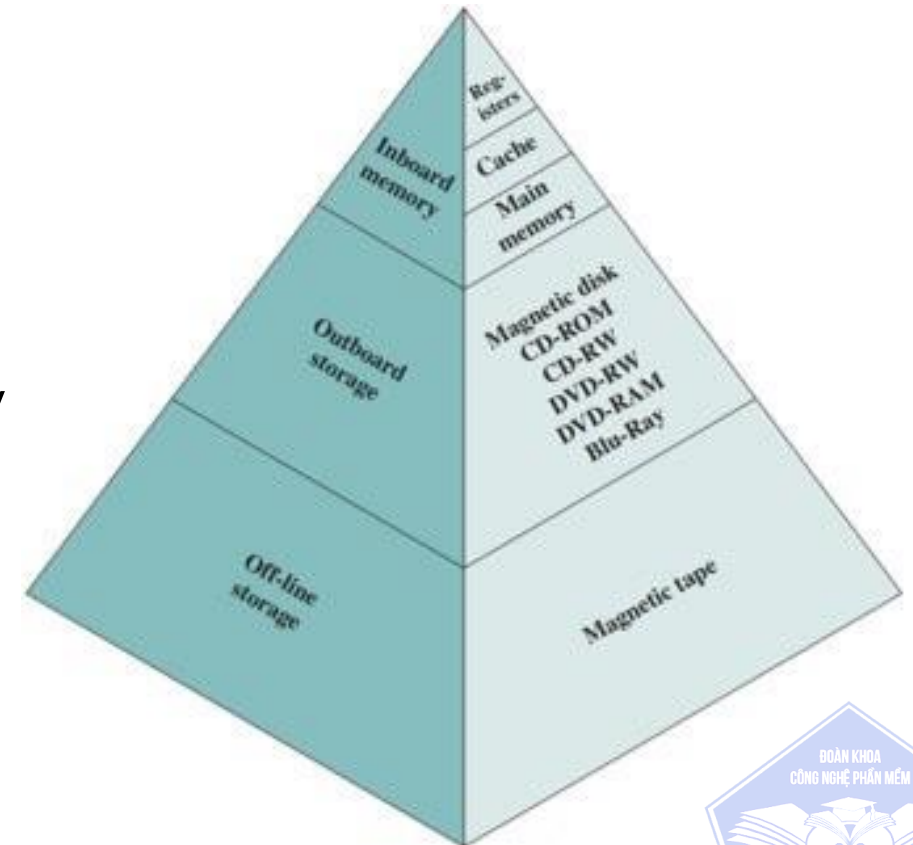
- Bộ nhớ (Memory) - Phân cấp bộ nhớ
 - Bộ nhớ càng nhanh thì càng đắt - Dung lượng nhỏ
 - Bộ nhớ càng rẻ thì càng chậm - Dung lượng lớn
 - Phân cấp bộ nhớ để có được bộ nhớ vừa nhanh và vừa có dung lượng lớn



I. TỔNG QUAN VỀ MÁY TÍNH

C. THÀNH PHẦN TRONG MÁY TÍNH

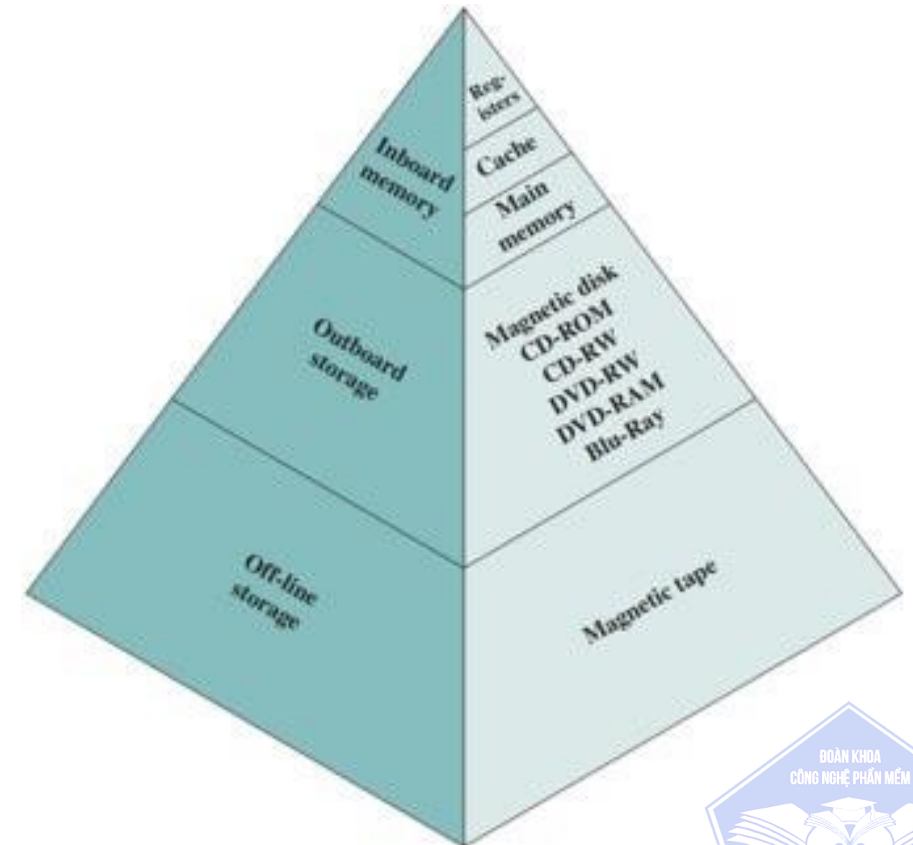
- Bộ nhớ (Memory) - Phân cấp bộ nhớ
 - Cấp 0: Tập các thanh ghi nằm trong bộ vi xử lý
 - Cấp 1: Primary cache (cache sơ cấp)
 - Cấp 2: Secondary cache (cache thứ cấp)
 - Cấp 3: Main Memory (Bộ nhớ chính)
 - Cấp 4: Secondary memory (Bộ nhớ thứ cấp – bộ nhớ ngoài)



I. TỔNG QUAN VỀ MÁY TÍNH

C. THÀNH PHẦN TRONG MÁY TÍNH

- Bộ nhớ (Memory) - Phân cấp bộ nhớ
 - **Càng về sau, dung lượng càng lớn, tốc độ càng chậm**



II. BIỂU DIỄN THÔNG TIN

A. THÔNG TIN. DỮ LIỆU. TÍN HIỆU

- Thông tin: dữ liệu lưu trữ, truyền đi, nhận được. Giải quyết tính không chắc chắn trong một ngữ cảnh cụ thể
- Dữ liệu: thể hiện của thông tin dưới dạng các tín hiệu vật lý. Thông tin chứa đựng ý nghĩa (tri thức) - dữ liệu là các dữ kiện không có cấu trúc và không có ý nghĩa rõ ràng nếu không được tổ chức và xử lý.
- Tín hiệu: bất kỳ đại lượng vật lý nào thay đổi theo thời gian, không gian - Đại lượng mang thông tin.

II. BIỂU DIỄN THÔNG TIN

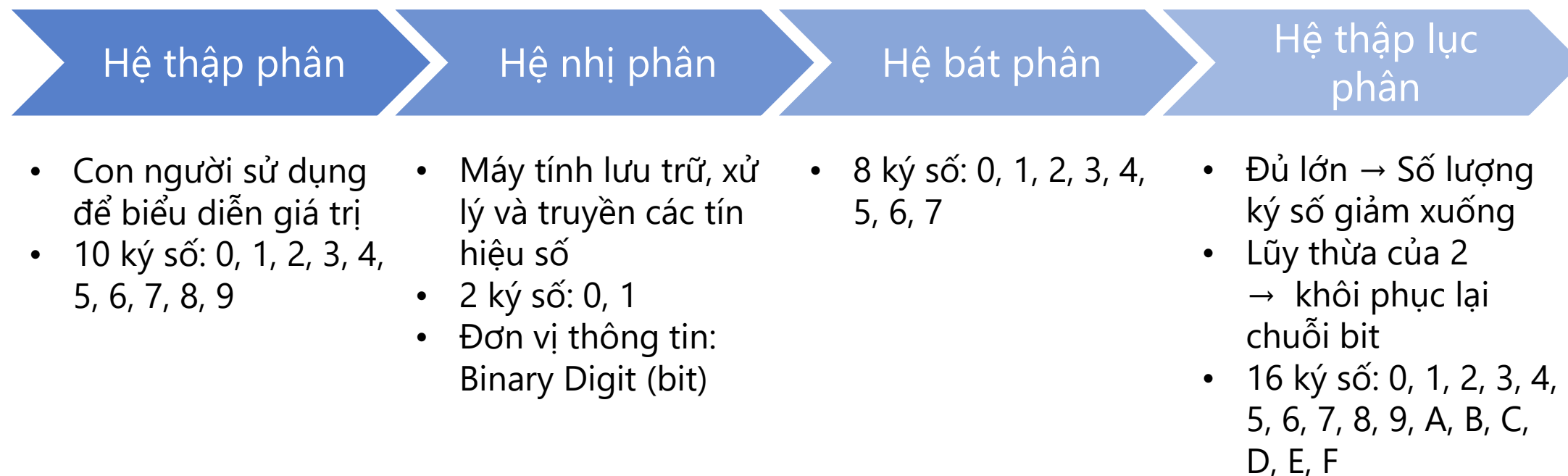
A. THÔNG TIN. DỮ LIỆU. TÍN HIỆU

- Máy tính xử lý thông tin như thế nào?
 - Máy tính không thể tự mình xử lý các tín hiệu trong thế giới thực mà chỉ có thể xử lý các tín hiệu số.
 - A/DC: chuyển đổi các tín hiệu tương tự trong thế giới thực thành các tín hiệu số.
 - Ngược lại, để phản hồi lại thế giới thực thì cần phải có D/AC



II. BIỂU DIỄN THÔNG TIN

B. CÁC HỆ SỐ BIỂU DIỄN THÔNG TIN



II. BIỂU DIỄN THÔNG TIN

B. CÁC HỆ SỐ BIỂU DIỄN THÔNG TIN

Cơ số 10	0	1	2	3	4	5	6	7
Cơ số 2	0000	0001	0010	0011	0100	0101	0110	0111
Cơ số 16	0	1	2	3	4	5	6	7

Cơ số 10	8	9	10	11	12	13	14	15
Cơ số 2	1000	1001	1010	1011	1100	1101	1110	1111
Cơ số 16	8	9	A	B	C	D	E	F


II. BIỂU DIỄN THÔNG TIN

C. CHUYỂN ĐỔI HỆ CƠ SỐ

- Thập phân sang nhị phân, bát phân, thập lục phân
 - Lần lượt chia cho cơ số mình muốn chuyển cho đến khi thương số bằng 0.
 - Kết quả là các số dư trong các phép chia viết ra theo thứ tự ngược lại.

$$923_{10} = 39B_{16}$$

923		16		
B		57		16
		9		3
				3
				0





II. BIỂU DIỄN THÔNG TIN

C. CHUYỂN ĐỔI HỆ CƠ SỐ

- Thập phân sang nhị phân
 - Phân tích số đã cho thành tổng của các lũy thừa 2, tìm lũy thừa 2 lớn nhất trước.
 - Số mũ của các lũy thừa 2 chính là vị trí mà bit có trọng số tương ứng là 1, còn các lũy thừa nào không có trong tổng thì trọng số tương ứng là 0.

$$23 = 2^4 + 2^2 + 2^1 + 2^0$$

2^4	2^3	2^2	2^1	2^0
1	0	1	1	1

$$15 = 2^3 + 2^2 + 2^1 + 2^0$$

2^4	2^3	2^2	2^1	2^0
0	1	1	1	1

II. BIỂU DIỄN THÔNG TIN

C. CHUYỂN ĐỔI HỆ CƠ SỐ

- Nhị phân, bát phân, thập lục phân sang thập phân
 - Đánh vị trí các ký số (từ phải sang), thứ tự bắt đầu từ 0
 - Trọng số: Cơ số (của hệ) ^ vị trí
 - Giá trị mỗi kí số: Kí số . Trọng số
 - Cộng tất cả các giá trị của các kí số

$$10101_2 = 2^4 + 2^2 + 2^1 = 21_{10}$$

4	3	2	1	0
1	0	1	0	1

$$14F_{16} = 1.16^2 + 4.16^1 + 15.16^0 = 335_{10}$$

2	1	0
1	4	F



II. BIỂU DIỄN THÔNG TIN

C. CHUYỂN ĐỔI HỆ CƠ SỐ

- Nhị phân, bát phân, thập lục phân sang thập phân
 - Tuy nhiên, khi chuyển từ hệ bát phân, thập lục phân sang thập phân, người ta thường dùng hệ nhị phân làm trung gian



II. BIỂU DIỄN THÔNG TIN

C. CHUYỂN ĐỔI HỆ CƠ SỐ

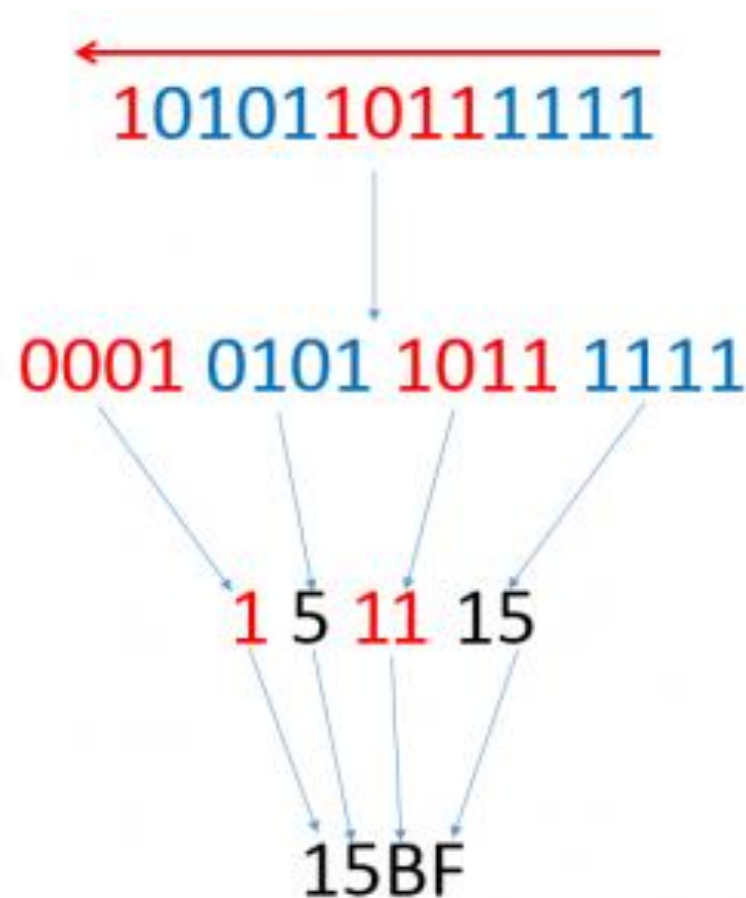
- Bát phân, thập lục phân sang nhị phân và ngược lại
 - Mỗi chữ số trong hệ **bát phân** được biểu diễn thành **bộ 3 bit** trong hệ nhị phân
 - Mỗi chữ số trong hệ **thập lục phân** được biểu diễn thành **bộ 4 bit** trong hệ nhị phân



II. BIỂU DIỄN THÔNG TIN

C. CHUYỂN ĐỔI HỆ CƠ SỐ

- Bát phân, thập lục phân sang nhị phân và ngược lại
 - Nhóm các ký số của số nhị phân thành các bộ tương ứng, nếu thiếu thì thêm 0 vào phía trước bộ
 - Mỗi bộ số biểu diễn cho một ký số trong hệ cơ số cần chuyển đổi



II. BIỂU DIỄN THÔNG TIN

C. CHUYỂN ĐỔI HỆ CƠ SỐ - BÀI TẬP

THẬP PHẦN	47				248	
NHỊ PHẦN				110110010		
THẬP LỤC PHẦN		25F				127
BÁT PHẦN			64			

II. BIỂU DIỄN THÔNG TIN

C. CHUYỂN ĐỔI HỆ CƠ SỐ - BÀI TẬP

THẬP PHẦN	47	607	52	434	248	295
NHỊ PHẦN	101111	1001011111	110100	110110010	11111000	100100111
THẬP LỤC PHẦN	2F	25F	34	1B2	F8	127
BÁT PHẦN	57	1137	64	662	370	447

II. BIỂU DIỄN THÔNG TIN

D. TÍNH TOÁN TRONG HỆ NHỊ PHÂN

CỘNG	TRỪ
$0 + 0 = 0$	$0 - 0 = 0$
$0 + 1 = 1$	$0 - 1 = 1$ (mượn 1)
$1 + 0 = 1$	$1 - 0 = 1$
$1 + 1 = 0$ (nhớ 1)	$1 - 1 = 0$

$$\begin{array}{r} 14 \\ + 7 \\ \hline 21 \end{array}$$

$$\begin{array}{r} 19 \\ - 6 \\ \hline 13 \end{array}$$

$$\begin{array}{r} 01110 \\ + 00111 \\ \hline 10101 \end{array}$$

$$\begin{array}{r} 10011 \\ - 00110 \\ \hline 01101 \end{array}$$



II. BIỂU DIỄN THÔNG TIN

D. TÍNH TOÁN TRONG HỆ NHỊ PHÂN

CỘNG	TRỪ
$0 + 0 = 0$	$0 - 0 = 0$
$0 + 1 = 1$	$0 - 1 = 1$ (mượn 1)
$1 + 0 = 1$	$1 - 0 = 1$
$1 + 1 = 0$ (nhớ 1)	$1 - 1 = 0$

$$\begin{array}{r} 6 \\ - 19 \\ \hline - 13 \end{array}$$

$$\begin{array}{r} 00110 \\ - 10011 \\ \hline 10011 \\ \text{Sai} \end{array}$$



II. BIỂU DIỄN THÔNG TIN

D. TÍNH TOÁN TRONG HỆ NHỊ PHÂN

- Thêm 1 bit làm dấu (Dấu và độ lớn):

- 0 là dấu +, 1 là dấu -

- Bù 1 – Bù 2 để biểu diễn số âm

- Nhị phân

- Bù 1: Nghịch đảo bit (không đảo bit dấu)

- Bù 2: Cộng thêm 1

- Biểu diễn bù 2 của một số dương là chính nó

Thập phân	+19
Nhị phân	00010011
BD bù 2	00010011

Thập phân	-13
Nhị phân	10001101
BD bù 1	11110010
BD bù 2	11110011



II. BIỂU DIỄN THÔNG TIN

D. TÍNH TOÁN TRONG HỆ NHỊ PHÂN

- Tính toán trên hệ nhị phân dựa vào bù 2
 - $A - B = A + (\text{bù 2 của } B)$
 - Sau khi thực hiện phép tính nếu chưa ra kết quả như ý, thực hiện bù 1 -> bù 2 kết quả
 - Không thực hiện đảo bit bên trái ngoài cùng
=> bit dấu

Thập phân	-19
Nhị phân	10010011
BD Bù 1	11101100
BD Bù 2	11101101

$$\begin{array}{r} 6 \\ - 19 \\ \hline - 13 \end{array}$$

$$\begin{array}{r} 00000110 \\ + 11101101 \\ \hline 11110011 \\ \text{Bù 1 } 10001100 \\ \text{Bù 2 } 10001101 \end{array}$$



II. BIỂU DIỄN THÔNG TIN

D. TÍNH TOÁN TRONG HỆ NHỊ PHÂN

- Tính toán trên hệ nhị phân dựa vào bù 2

➤ $A - B = A + (\text{bù 2 của } B)$

➤ Sau khi thực hiện phép tính nếu chưa ra kết

quả như ý, thực hiện bù 1 -> bù 2 kết quả

➤ Không thực hiện đảo bit bên trái

ngoài cùng => bit dấu

$$\begin{array}{r} - 7 \\ - 15 \\ \hline - 22 \end{array}$$

$$\begin{array}{r} 11111001 \\ + 11110001 \\ \hline 11101010 \end{array}$$



II. BIỂU DIỄN THÔNG TIN

D. TÍNH TOÁN TRONG HỆ NHỊ PHÂN - BÀI TẬP

$$23_{10} + 34_{10}$$

$$22_8 - 15_8$$

$$12_{10} - 20_{10}$$

$$-7_{10} - 24_{10}$$

II. BIỂU DIỄN THÔNG TIN

D. TÍNH TOÁN TRONG HỆ NHỊ PHÂN - BÀI TẬP

$$23_{10} + 34_{10}$$

$$\begin{array}{r} 010111 \\ + 100010 \\ \hline \end{array}$$

$$22_8 - 15_8$$

$$\begin{array}{r} 10010 \\ - 01101 \\ \hline \end{array}$$

$$12_{10} - 20_{10}$$

$$\begin{array}{r} 001100 \\ + 101100 \\ \hline \end{array}$$

$$- 7_{10} - 24_{10}$$

$$\begin{array}{r} 111001 \\ + 101000 \\ \hline \end{array}$$

II. BIỂU DIỄN THÔNG TIN

D. TÍNH TOÁN TRONG HỆ NHỊ PHÂN - BÀI TẬP

$$23_{10} + 34_{10}$$

$$\begin{array}{r} 010111 \\ + 100010 \\ \hline 111001 \\ \Rightarrow 57_{10} \end{array}$$

$$22_8 - 15_8$$

$$\begin{array}{r} 10010 \\ - 01101 \\ \hline 00101 \\ \Rightarrow 5_8 \\ 5_{10} \end{array}$$

$$12_{10} - 20_{10}$$

$$\begin{array}{r} 001100 \\ + 101100 \\ \hline 111000 \\ \Rightarrow 100111 \\ 101000 \\ -8_{10} \end{array}$$

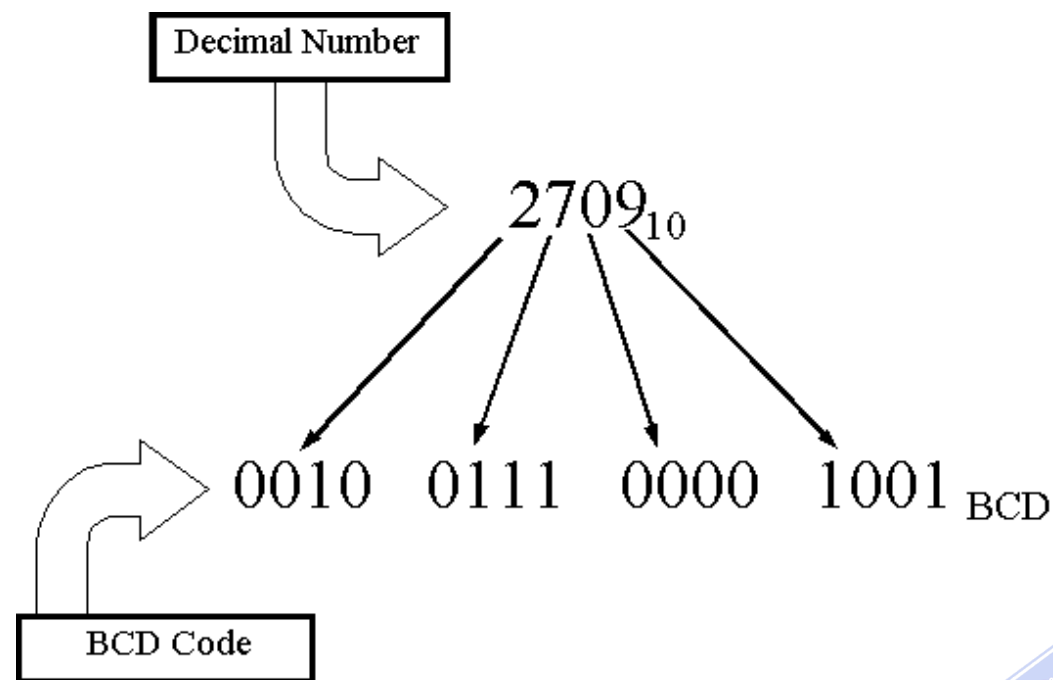
$$-7_{10} - 24_{10}$$

$$\begin{array}{r} 111001 \\ + 101000 \\ \hline 100001 \\ \text{Bù 1} \quad 111110 \\ \text{Bù 2} \quad 111111 \\ -31_{10} \end{array}$$

II. BIỂU DIỄN THÔNG TIN

E. MÃ BCD

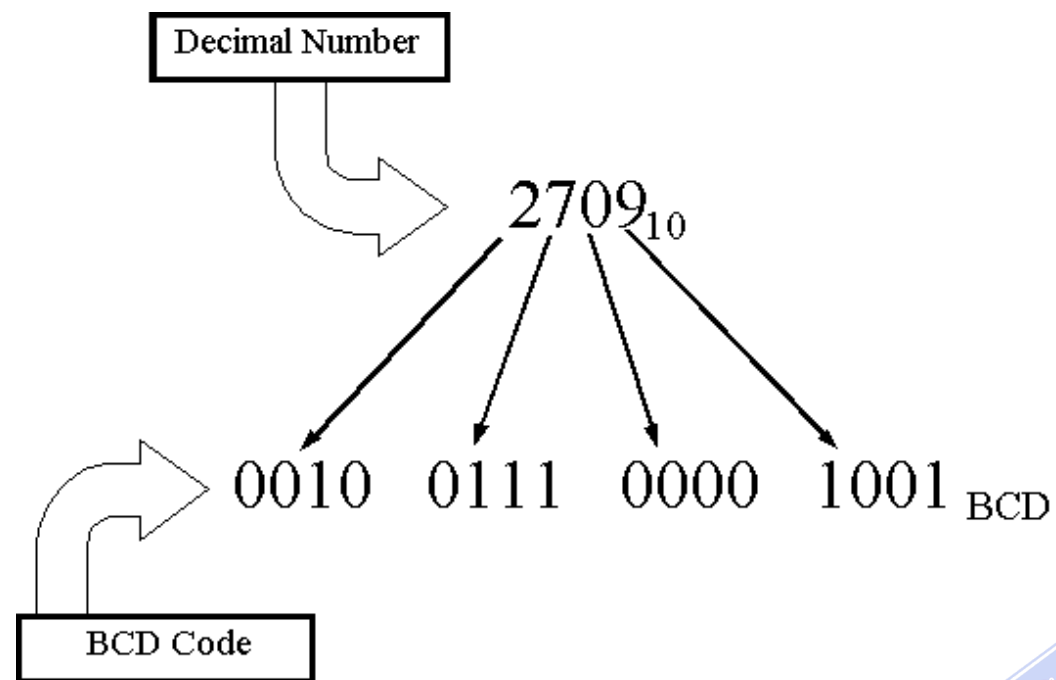
- Phương pháp biểu diễn
 - Phù hợp với phần cứng máy tính
 - Dễ hiểu cho con người
 - Ứng dụng: Đèn đếm số 7 đoạn
- Cách biểu diễn mã BCD: Sử dụng **mỗi 4 bit** để mã hóa duy nhất **1 ký số** thập phân.



II. BIỂU DIỄN THÔNG TIN

E. MÃ BCD - VÍ DỤ

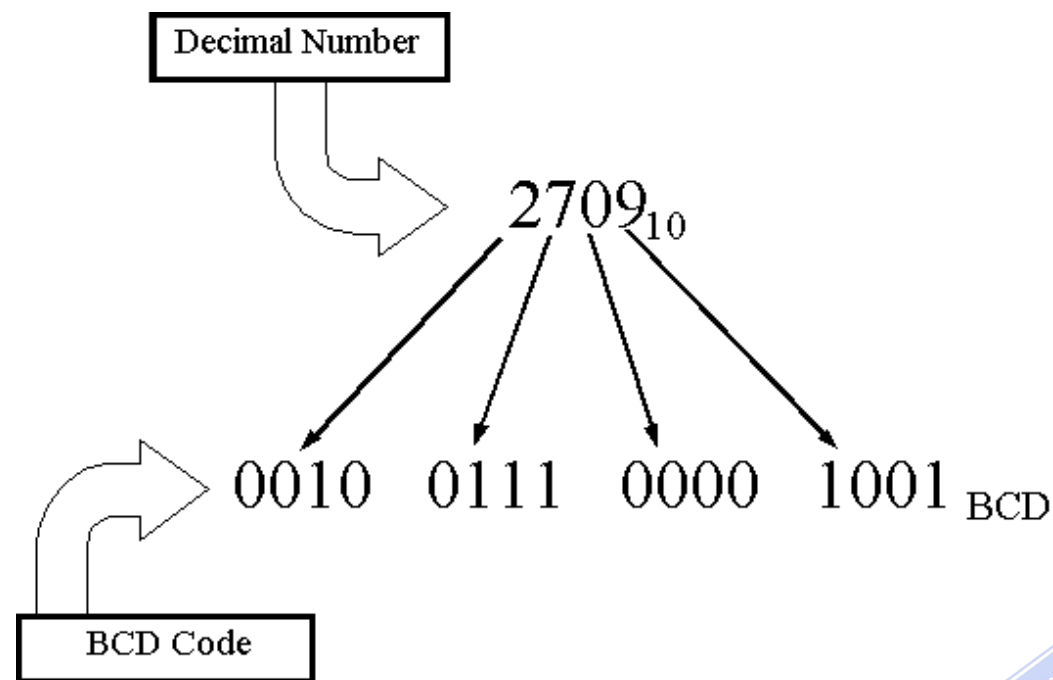
- $00100110100110000111_{\text{BCD}}$
- $100101010010011000010000_{\text{BCD}}$



II. BIỂU DIỄN THÔNG TIN

E. MÃ BCD - VÍ DỤ

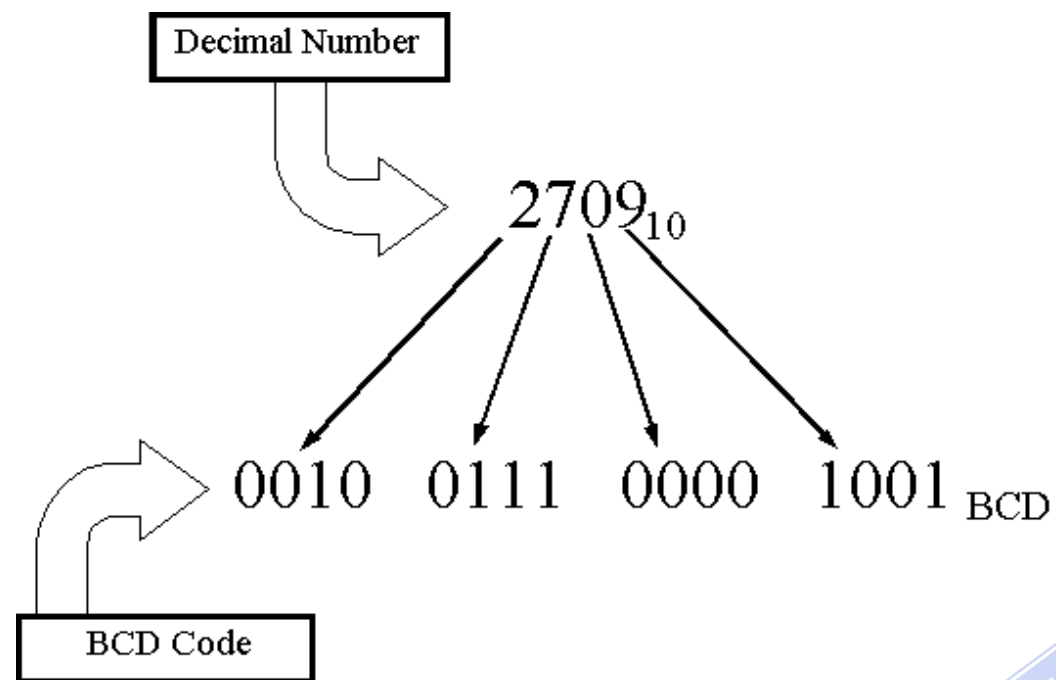
- $00100110100110000111_{\text{BCD}}$
 - $0010\ 0110\ 1001\ 1000\ 0111_{\text{BCD}}$
- $100101010010011000010000_{\text{BCD}}$
 - $1001\ 0101\ 0010\ 0110\ 0001\ 0000_{\text{BCD}}$



II. BIỂU DIỄN THÔNG TIN

E. MÃ BCD - VÍ DỤ

- $00100110100110000111_{\text{BCD}}$
 - $0010\ 0110\ 1001\ 1000\ 0111_{\text{BCD}}$
 - 26987_{10}
- $100101010010011000010000_{\text{BCD}}$
 - $1001\ 0101\ 0010\ 0110\ 0001\ 0000_{\text{BCD}}$
 - 952610_{10}



II. BIỂU DIỄN THÔNG TIN

F. MÃ ASCII

- Phương pháp sử dụng 7 bit để biểu diễn mỗi ký tự.

$b_4b_3b_2b_1$	$b_7b_6b_5$							
	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	“	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	‘	7	G	W	g	w
1000	BS	CAN	(8	H	X	h	x
1001	HT	EM)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	-	o	DEL

III. ĐẠI SỐ BOOLEAN

A. GIỚI THIỆU

- Đại số Boolean là một cấu trúc đại số liên quan đến việc thao tác với các biến luận lý nhị phân
- Các nguyên tắc cơ bản:
 - Sử dụng hệ cơ số nhị phân
 - Các phép toán luận lý (cộng, nhân, bù, ...)
 - Độ ưu tiên của các phép toán
 - Tính đóng (closure)



III. ĐẠI SỐ BOOLEAN

B. CÁC PHÉP TOÁN LUẬN LÝ - PHÉP CỘNG

- Phép toán: Dấu +, | hay OR
- Biểu thức : $A + B = C$ Hay $A \text{ OR } B = C$
- Nguyên tắc:
 - Kết quả trả về 0 (FALSE) khi và chỉ khi tất cả giá trị đầu vào là 0 (FALSE).
 - Kết quả là 1 (TRUE) khi có bất kì một giá trị nhập vào có giá trị là 1 (TRUE).

Input			Output
A	+	B	= C
0		0	0
0		1	1
1		0	1
1		1	1

Ví dụ:

A	1 0 0 1 1 0 1 0
B	1 1 0 0 1 0 0 1
A + B hay A OR B	1 1 0 1 1 0 1 1



Sharing is learning

III. ĐẠI SỐ BOOLEAN

B. CÁC PHÉP TOÁN LUẬN LÝ - PHÉP NHÂN

- Phép toán: Dấu \cdot , & hay AND
- Biểu thức : $A \cdot B = C$ Hay $A \text{ AND } B = C$
- Nguyên tắc:
 - Kết quả trả về 1 (TRUE) khi và chỉ khi tất cả giá trị đầu vào là 1 (TRUE).
 - Kết quả là 0 (FALSE) khi có bất kì một giá trị nhập vào có giá trị là 0 (FALSE).

Input		Output
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

Ví dụ:

A	1 0 0 1 1 0 1 0
B	1 1 0 0 1 0 0 1
A . B hay A AND B	1 0 0 0 1 0 0 0



Sharing is learning

III. ĐẠI SỐ BOOLEAN

B. CÁC PHÉP TOÁN LUẬN LÝ - PHÉP BÙ

- Phép toán: Dấu $'$, $\bar{}$, \sim hay NOT
- Biểu thức : \bar{A} Hay NOT A
- Nguyên tắc:

Input	Output
A	\bar{A}
0	1
1	0

Ví dụ:

A	1 0 0 1 1 0 1 0
\bar{A} hay NOT A	0 1 1 0 0 1 0 1

8

- Kết quả trả về 1 (TRUE) nếu giá trị đầu vào là 0 (FALSE).
- Kết quả là 0 (FALSE) nếu giá trị nhập vào là 1 (TRUE).



Sharing is learning

III. ĐẠI SỐ BOOLEAN

B. CÁC PHÉP TOÁN LUẬN LÝ - **ĐỘ ƯU TIÊN**

- Biểu thức được tính **từ trái sang phải**.
- Biểu thức trong **ngoặc đơn** -> **NOT** -> **AND** -> **OR**
- VD: $C \cdot (A' + B \cdot C)' + (A + B) \cdot C'$



III. ĐẠI SỐ BOOLEAN

C. CÁC TIÊN ĐỀ BOOLEAN

Một tập B khác rỗng cùng với các thao tác (phép toán) AND (\cdot), OR ($+$) và NOT ($\bar{}$) được gọi là một đại số Boolean nếu các tiên đề sau đây được thỏa mãn với mọi $x, y, z \in B$:

- **Tiên đề 1:** Cấu trúc đóng với các phép toán (\cdot) và ($+$). Nếu $x, y \in B$ thì: $(x + y) \in B$ và $x \cdot y \in B$
- **Tiên đề 2:** Tồn tại 2 phần tử trung hòa khác nhau thuộc B, ký hiệu là 0 và 1 sao cho:
 - $x \cdot 1 = 1 \cdot x = x$
 - $x + 0 = 0 + x = x$



III. ĐẠI SỐ BOOLEAN

C. CÁC TIÊN ĐỀ BOOLEAN

- **Tiên đề 3:** Tính giao hoán

➤ $x \cdot y = y \cdot x$

➤ $x + y = y + x$

- **Tiên đề 4:** Tính phân phối

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

➤ $x + y \cdot z = (x + y)(x + z)$

- **Tiên đề 5:** Tồn tại phần tử bù. Với mọi $x \in B$, tồn tại duy nhất phần tử bù $x' \in B$ sao cho:

➤ $x \cdot x' = x' \cdot x = 0$

➤ $x + x' = x' + x = 1$

- **Tiên đề 6:** Tồn tại ít nhất 2 phần tử $x, y \in B$ sao cho $x \neq y$

III. ĐẠI SỐ BOOLEAN

D. HÀM BOOLEAN

- Kết hợp các **biến, hằng số, toán tử, dấu ngoặc** tạo thành một **Biểu thức Boolean**.
 - Ví dụ: $x + yz$
- Kết hợp theo thứ tự: **1 tên hàm, 1 dấu bằng và 1 biểu thức Boolean** được một **Hàm Boolean** (Hàm Boolean Dạng chuẩn)
 - Ví dụ: $f(x, y, z) = x + yz$



III. ĐẠI SỐ BOOLEAN

D. HÀM BOOLEAN

- Bảng chân trị (bảng sự thật)
 - Thể hiện mối quan hệ giữa giá trị của một hàm Boolean và các biến của hàm đó

- 2^n hàng (n là số biến)

- n+1 cột

$$f(x, y, z) = x + yz$$

Liệt kê tất cả các tổ hợp có thể

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

*Giá trị của hàm
tương ứng với mỗi
tổ hợp các biến*



Sharing is learning

III. ĐẠI SỐ BOOLEAN

D. HÀM BOOLEAN

- Dạng chính tắc của hàm Boolean
 - Literal: là 1 biến hay phủ định của biến đó (A hay A')
 - Term của n literal là sự kết hợp của các literal mà mỗi literal chỉ xuất hiện một lần duy nhất.

Ví dụ: term của 3 literal A' , B , C là $A'.B.C$

- Biểu diễn dạng chính tắc của hàm Boolean bằng một trong hai cách:
 - **Tổng của các minterm** khiến hàm Boolean có giá trị **1 (1-minterm)**
 - **Tích của các maxterm** khiến hàm Boolean có giá trị **0 (0-maxterm)**



III. ĐẠI SỐ BOOLEAN

D. HÀM BOOLEAN

- Dạng chính tắc của hàm Boolean - Ví dụ biểu diễn theo **minterm**

Biến			Minterm		Maxterm	
x	y	z	Biểu thức	Ký hiệu	Biểu thức	Ký hiệu
0	0	0	$\bar{x}\bar{y}\bar{z}$	m_0	$x+y+z$	M_0
0	0	1	$\bar{x}\bar{y}z$	m_1	$x+y+\bar{z}$	M_1
0	1	0	$\bar{x}y\bar{z}$	m_2	$x+\bar{y}+z$	M_2
0	1	1	$\bar{x}yz$	m_3	$x+\bar{y}+\bar{z}$	M_3
1	0	0	$x\bar{y}\bar{z}$	m_4	$\bar{x}+y+z$	M_4
1	0	1	$x\bar{y}z$	m_5	$\bar{x}+y+\bar{z}$	M_5
1	1	0	$xy\bar{z}$	m_6	$\bar{x}+\bar{y}+z$	M_6
1	1	1	xyz	m_7	$\bar{x}+\bar{y}+\bar{z}$	M_7

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

$$f(x, y, z) = \bar{x}\bar{y}z + \bar{x}yz + x\bar{y}\bar{z} + xy\bar{z} + xyz$$



III. ĐẠI SỐ BOOLEAN

D. HÀM BOOLEAN

- Dạng chính tắc của hàm Boolean - Ví dụ biểu diễn theo **maxterm**

Biến			Minterm		Maxterm	
x	y	z	Biểu thức	Ký hiệu	Biểu thức	Ký hiệu
0	0	0	$\bar{x} \bar{y} \bar{z}$	m_0	$x + y + z$	M_0
0	0	1	$\bar{x} \bar{y} z$	m_1	$x + y + \bar{z}$	M_1
0	1	0	$\bar{x} y \bar{z}$	m_2	$x + \bar{y} + z$	M_2
0	1	1	$\bar{x} y z$	m_3	$x + \bar{y} + \bar{z}$	M_3
1	0	0	$x \bar{y} \bar{z}$	m_4	$\bar{x} + y + z$	M_4
1	0	1	$x \bar{y} z$	m_5	$\bar{x} + y + \bar{z}$	M_5
1	1	0	$x y \bar{z}$	m_6	$\bar{x} + \bar{y} + z$	M_6
1	1	1	$x y z$	m_7	$\bar{x} + \bar{y} + \bar{z}$	M_7

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

$$f(x, y, z) = (x + y + z) (x + \bar{y} + z) (\bar{x} + y + \bar{z})$$

III. ĐẠI SỐ BOOLEAN

E. TÍNH CHẤT

- Nguyên lí đối ngẫu: Đại số Boolean mang tính đối ngẫu

- Đổi phép toán (+) thành (•)
- Đổi phần tử đồng nhất 0 thành 1
- Ví dụ các đối ngẫu:

	Cột 1	Cột 2	Column 3
Row 1	$1 + 1 = 1$	$1 + 0 = 0 + 1 = 1$	$0 + 0 = 0$
Row 2	$0 \cdot 0 = 0$	$0 \cdot 1 = 1 \cdot 0 = 0$	$1 \cdot 1 = 1$

- Nếu một biểu thức Boolean là đúng thì **đối ngẫu của nó cũng đúng**



III. ĐẠI SỐ BOOLEAN

E. TÍNH CHẤT - ĐỊNH LÝ ĐẠI SỐ BOOLEAN

- **Định lý 1:** Tính lũy đẳng

➤ $x + x = x$

➤ $x \cdot x = x$

- **Định lý 2:** Tính nuốt

➤ $x + 1 = 1$

➤ $x \cdot 0 = 0$

- **Định lý 3:** Tính hấp

thụ

➤ $x + x \cdot y = x$

➤ $x(x + y) = x$

- **Định lý 4:** Tính phủ

định của phủ định

➤ $(x')' = x$

- **Định lý 5:** Tính kết hợp

➤ $x + (y + z) = (x + y) + z$

➤ $x(y \cdot z) = (x \cdot y)z$

- **Định lý 6: De-Morgan**

➤ $(x + y)' = x' \cdot y'$

➤ $(x \cdot y)' = x' + y'$



III. ĐẠI SỐ BOOLEAN

F. TỐI ƯU HÓA LUẬN LÝ - THÀNH PHẦN THỪA CỦA BIỂU THỨC

- Một biểu thức gọi là dư thừa nếu nó có chứa
 - Literal lặp: $x.x$ hay $x + x$
 - Biến và bù của biến: $x.x'$ hay $x + x'$
 - Hằng: 0 hay 1
- Các thành phần dư thừa có thể loại bỏ khỏi biểu thức
- Thành phần dư thừa không cần hiện thực trong phần cứng



III. ĐẠI SỐ BOOLEAN

F. TỐI ƯU HÓA LUẬN LÝ

- Tối ưu hóa luận lý: Làm **giảm số lượng tổng/tích** hoặc **số lượng biến** hoặc **phần bù** của nó trong mỗi tổng/tích sử dụng các tiên đề, định lý Boolean



III. ĐẠI SỐ BOOLEAN

F. TỐI ƯU HÓA LUẬN LÝ - MỘT SỐ TIÊN ĐỀ, ĐỊNH LÝ THƯỜNG DÙNG

- **Tiên đề 2:** Phần tử trung hòa
 - $x \cdot 1 = 1 \cdot x = x$
 - $x + 0 = 0 + x = x$
- **Định lý 1:** Tính lũy đẳng
 - $x + x = x$
 - $x \cdot x = x$
- **Định lý 3:** Tính hấp thụ
 - $x + x \cdot y = x$
 - $x(x + y) = x$
- **Tiên đề 5:** Tồn tại phần tử bù.
 - $x \cdot x' = x' \cdot x = 0$
 - $x + x' = x' + x = 1$
- **Định lý 2:** Tính nuốt
 - $x + 1 = 1$
 - $x \cdot 0 = 0$
- **Định lý 6:** De-Morgan
 - $(x + y)' = x' \cdot y'$
 - $(x \cdot y)' = x' + y'$



III. ĐẠI SỐ BOOLEAN

G. PHƯƠNG PHÁP KARNAUGH (K-MAP)

Phương pháp tối ưu luận lý bằng hình học trực quan dựa trên các tính chất của đại số

Boolean:

- $\mathbf{xy + xy' = x(y + y') = x \cdot 1 = x}$
 - **Tổng của hai tích** khác nhau **đúng 1 bit** thì kết quả sẽ **rút gọn được bit khác nhau**
- $\mathbf{(x + y)(x + y') = x + y.y' = x + 0 = x}$
 - **Tích của hai tổng** khác nhau **đúng 1 bit** thì kết quả sẽ **rút gọn được bit khác nhau**



III. ĐẠI SỐ BOOLEAN

G. PHƯƠNG PHÁP KARNAUGH (K-MAP)

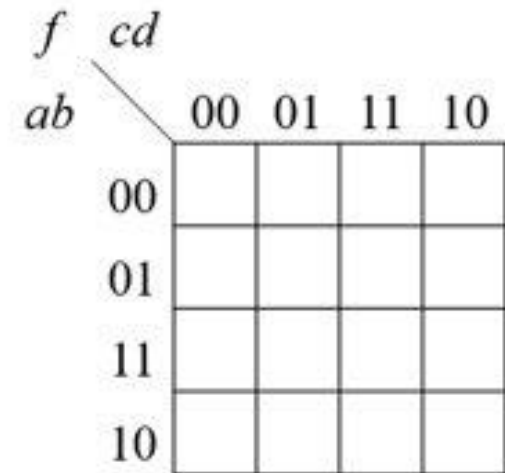
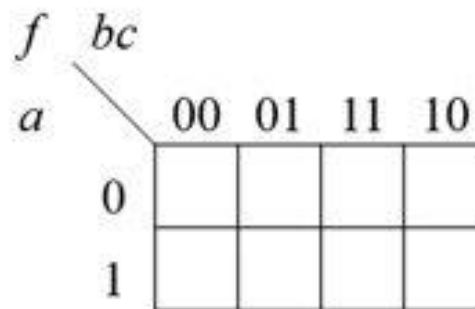
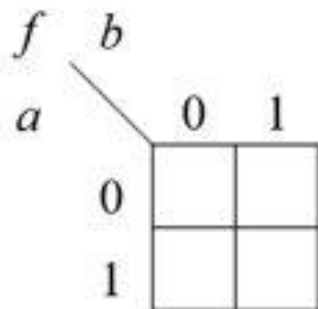
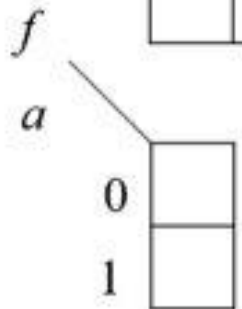
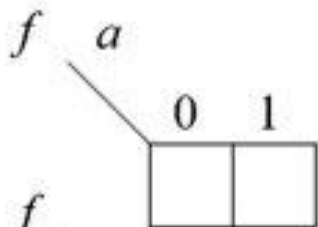
K-map là mảng 2 chiều các ô

- Số lượng ô = 2^n (n là số biến)
- Số lượng ô trên mỗi chiều = 2^i (i là số biến được gán trên mỗi chiều)
- Mỗi ô được gán 1 tổ hợp theo mã Gray: 2 chuỗi bit liên tiếp khác nhau 1 bit



III. ĐẠI SỐ BOOLEAN

G. PHƯƠNG PHÁP KARNAUGH (K-MAP)



III. ĐẠI SỐ BOOLEAN

G. PHƯƠNG PHÁP KARNAUGH

- Lập K-map (một trong hai cách):
 - Viết 1 tại các điểm biểu diễn các minterm vào ô tương ứng của K-map
 - Viết 0 tại các điểm biểu diễn các maxterm vào ô tương ứng của K-map

x	y	z	f
0	0	0	m_0/M_0
0	0	1	m_1/M_1
0	1	0	m_2/M_2
0	1	1	m_3/M_3
1	0	0	m_4/M_4
1	0	1	m_5/M_5
1	1	0	m_6/M_6
1	1	1	m_7/M_7

f	yz	00	01	11	10
x	0	m_0	m_1	m_3	m_2
1		m_4	m_5	m_7	m_6

f	yz	00	01	11	10
x	0	M_0	M_1	M_3	M_2
1		M_4	M_5	M_7	M_6

III. ĐẠI SỐ BOOLEAN

G. PHƯƠNG PHÁP KARNAUGH

- Lập K-map (một trong hai cách):
 - Viết 1 tại các điểm biểu diễn các minterm vào ô tương ứng của K-map
 - Viết 0 tại các điểm biểu diễn các maxterm vào ô tương ứng của K-map

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

f	yz	00	01	11	10
x	0		1	1	
	1	1		1	

f	yz	00	01	11	10
x	0	0			0
	1		0		0

III. ĐẠI SỐ BOOLEAN

G. PHƯƠNG PHÁP KARNAUGH (K-MAP) - VIẾT BIỂU THỨC

- Gom các nhóm **2^k ô liền kề** với $k \geq 0$
 - k là số biến được tối ưu trong mỗi nhóm
 - Gom các 1-minterm -> Tổng các tích có giá trị 1
 - Gom các 0-maxterm -> Tích các tổng có giá trị 0
- **Số lần gom phải ít nhất** => số tích/tổng của biểu thức cuối cùng là ít nhất
- Mỗi nhóm phải có **ít nhất 1 ô không thuộc** các nhóm khác
 - Tránh trường hợp dư thừa các tích/tổng mà các nhóm khác đã bao phủ



III. ĐẠI SỐ BOOLEAN

G. PHƯƠNG PHÁP KARNAUGH (K-MAP) - VIẾT BIỂU THỨC

$$F(x, y, z) = \sum m(1, 3, 4, 7)$$

$$F(a, b, c) = \prod M(1, 4, 5, 6)$$

F	yz				
		00	01	11	10
x	0		1	1	
	1	1		1	

$$F(x, y, z) = \bar{x}z + yz + x\bar{y}\bar{z}$$

F	bc				
		00	01	11	10
a	0		0		
	1	0	0		0

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



Sharing is learning

III. ĐẠI SỐ BOOLEAN

G. PHƯƠNG PHÁP KARNAUGH (K-MAP) - VIẾT BIỂU THỨC

$$f(x, y, z) = x + \bar{x}yz$$

f $x \backslash yz$					
		00	01	11	10
0				1	
1	1	1	1	1	1

$$F(x, y, z) = x + yz$$

$$f(x, y, z) = \bar{x}(x + y + \bar{z})$$

f $x \backslash yz$					
		00	01	11	10
0			0		
1	0	0	0	0	0

$$F(x, y, z) = \bar{x}(y + \bar{z})$$



III. ĐẠI SỐ BOOLEAN

G. PHƯƠNG PHÁP KARNAUGH (K-MAP) - VIẾT BIỂU THỨC

VD: Rút gọn các hàm sau bằng bìa Karnaugh

F1 (W, X, Y, Z) = $\Sigma(3, 4, 11, 12)$ theo dạng P.O.S (tích các tổng)

		F1 WX			
		00	01	11	10
YZ	00	0 0	4	12	8 0
	01	1 0	5	13 0	9 0
	11	3	7 0	15 0	11
	10	2 0	6 0	14 0	10 0

$(X + Y)$

$(\bar{X} + \bar{Z})$

$(\bar{Y} + Z)$

$$F1 = (X + Y)(\bar{X} + \bar{Z})(\bar{Y} + Z)$$

Hoặc $F1 = (X + Z)(Y + \bar{Z})(\bar{X} + \bar{Y})$

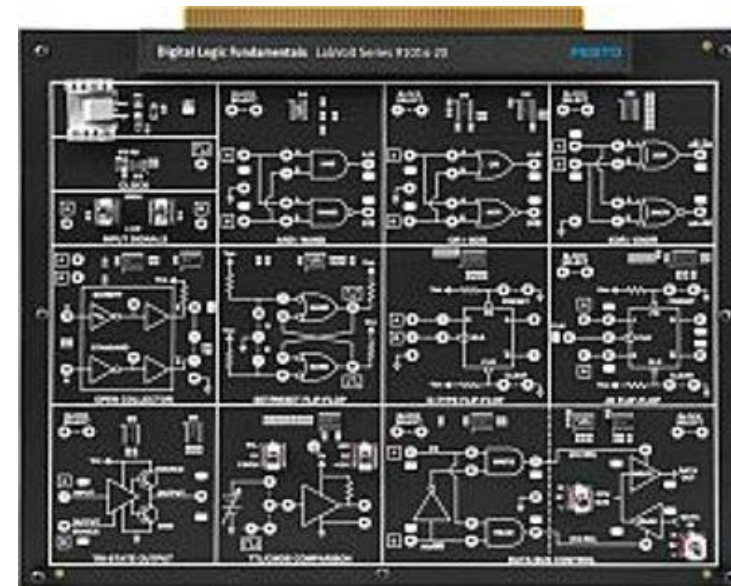
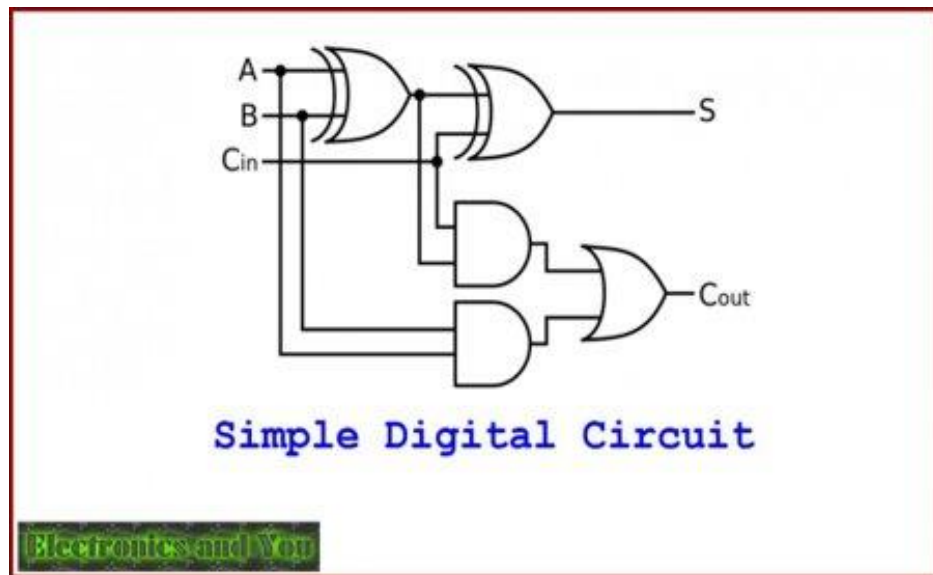


Sharing is learning

IV. MẠCH SỐ

A. KHÁI NIỆM

- Mạch số là mạch thu nhận, truyền, lưu trữ và xử lý tín hiệu số



Sharing is learning

IV. MẠCH SỐ

A. KHÁI NIỆM



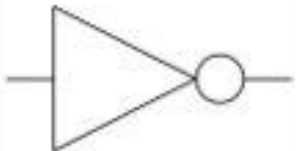
- Thiết kế mạch số
 - Xác định **chức năng** của mạch số
 - ✓ Tìm mối quan hệ giữa các ngõ vào và các ngõ ra
 - ✓ Dựa vào hàm Boolean và bảng chân trị
 - Xác định **cấu trúc** của mạch số:
 - ✓ Tìm và kết nối các thiết bị thực hiện các toán tử luận lý: AND, OR, NOT



IV. MẠCH SỐ

B. CÁC CỔNG LUẬN LÝ - **AND, OR, NOT**

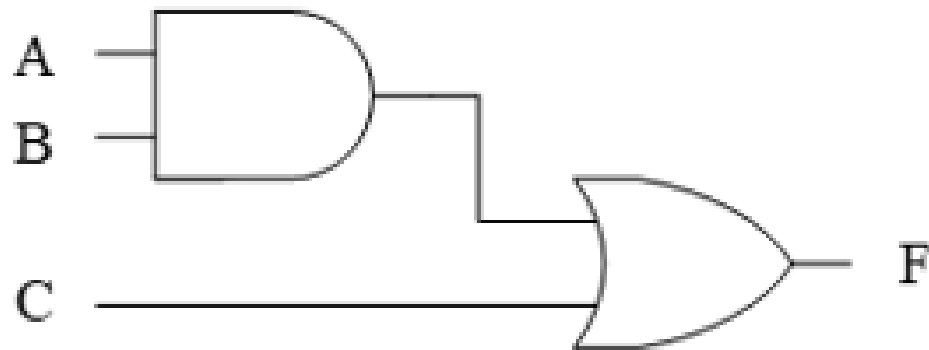
- Cổng AND: sự thực hiện vật lí của phép nhân luận lí (AND)
 - Tất cả input = 1 thì output = 1
- Cổng OR: sự thực hiện vật lí của phép cộng luận lí (OR)
 - Tất cả tín hiệu vào = 0 thì tín hiệu ra = 0
- Cổng NOT: sự thực hiện vật lí của phép bù
 - Đảo lại tín hiệu ra

Tên	Ký hiệu	Hàm Boolean	Bảng chân trị		
AND		$F = AB$	A	B	F
			0	0	0
			0	1	0
			1	0	0
			1	1	1
OR		$F = A + B$	A	B	F
			0	0	0
			0	1	1
			1	0	1
			1	1	1
NOT		$F = \bar{A}$	A	F	
			0	1	
			1	0	

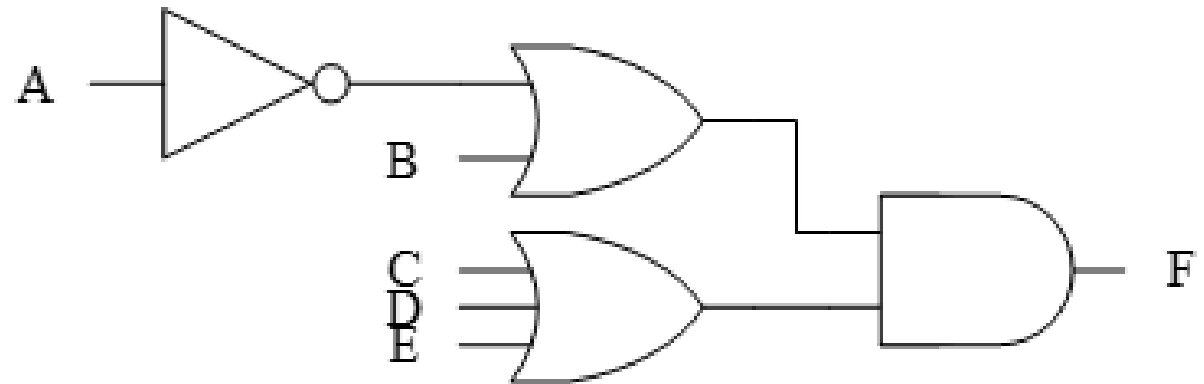
IV. MẠCH SỐ

B. CÁC CỔNG LUẬN LÝ - **AND, OR, NOT**

$$F = AB + C$$



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



IV. MẠCH SỐ

B. CÁC CÔNG LUẬN LÝ - **AND, OR, NOT**


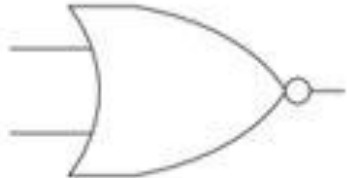
- Ví dụ: Hiện thực các hàm luận lý sau
- $F(A, B, C) = (A' + B)(C + D)B$
- $F(A, B, C) = (A' + BC)(C + ABD) + D$



IV. MẠCH SỐ

B. CÁC CỔNG LUẬN LÝ - **NAND, NOR**

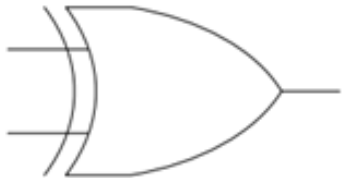

- Cổng NAND: một phần bù của cổng AND.
 - Cổng ra của NAND là 0 khi và chỉ khi tất cả cổng vào là 1.
- Cổng NOR: một phần bù của cổng OR.
 - Cổng ra của cổng NOR là 1 khi và chỉ khi tất cả các cổng vào là 0

Tên	Ký hiệu	Hàm Boolean	Bảng chân trị		
NAND		$F = \overline{AB}$	A	B	F
			0	0	1
			0	1	1
			1	0	1
			1	1	0
NOR		$F = \overline{A + B}$	A	B	F
			0	0	1
			0	1	0
			1	0	0
			1	1	0

IV. MẠCH SỐ

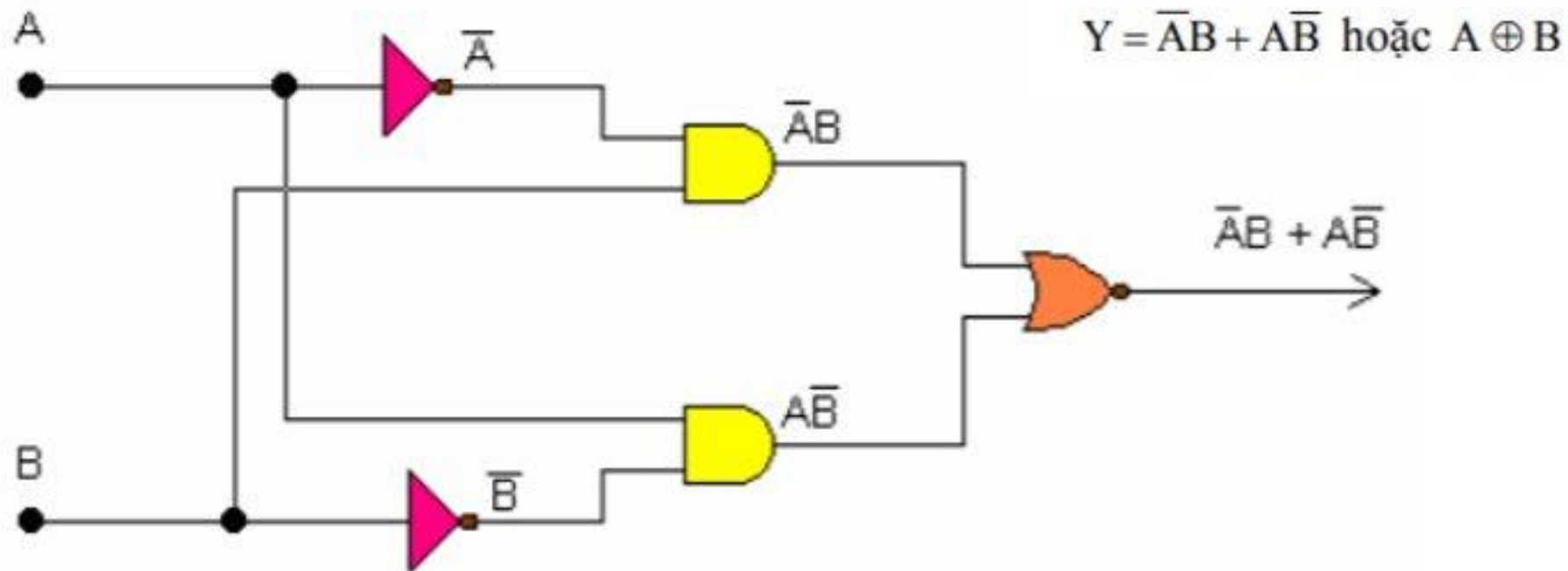
B. CÁC CỔNG LUẬN LÝ - **XOR, XNOR**

- Cổng ra của XOR bằng 1 khi **số đầu vào có giá trị 1 là số lẻ**
- Cổng ra của XNOR bằng 1 khi **số đầu vào có giá trị 0 là số chẵn**
- Hay nói: **XNOR là phần bù của XOR**

Tên	Ký hiệu	Hàm Boolean	Bảng chân trị		
XOR		$F = A \oplus B$	A	B	F
			0	0	0
			0	1	1
			1	0	1
			1	1	0
XNOR (NXOR)		$F = \overline{A \oplus B}$	A	B	F
			0	0	1
			0	1	0
			1	0	0
			1	1	1

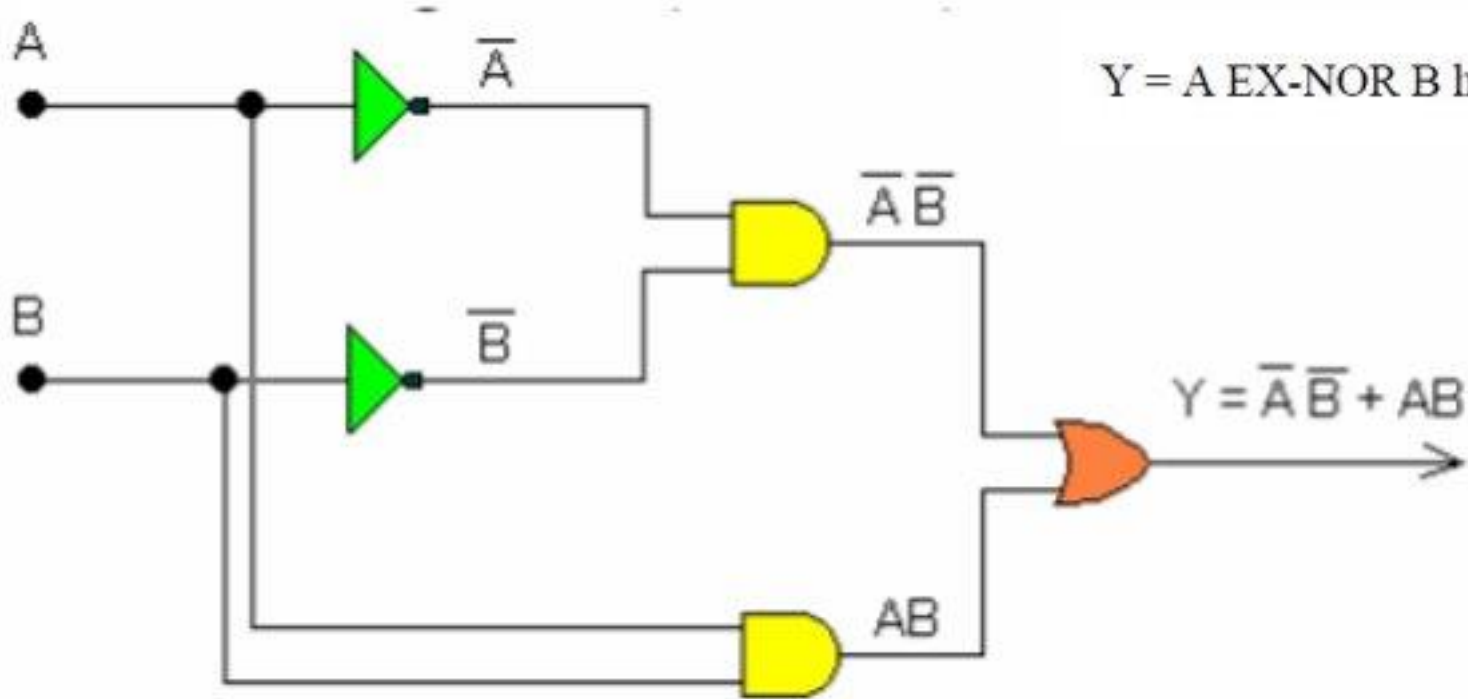
IV. MẠCH SỐ

B. CÁC CỔNG LUẬN LÝ - **XOR, XNOR**



IV. MẠCH SỐ

B. CÁC CỔNG LUẬN LÝ - **XOR, XNOR**



$$Y = A \text{ EX-NOR } B \text{ hay } Y = \bar{A}\bar{B} + \bar{A}\bar{B} = \bar{A} \oplus \bar{B}$$



IV. MẠCH SỐ

B. CÁC CỔNG LUẬN LÝ - **XOR, XNOR**

- Ví dụ: Hiện thực các hàm luận lý sau
- $F(A, B, C) = [(AB)' + C](A + B)'$
- $F(A, B, C) = A' + (B \text{ XOR } C)[C + (A \text{ XNOR } D)]$



IV. MẠCH SỐ

C. MẠCH TỔ HỢP

- Thiết bị tổ hợp là thiết bị có tính chất sau:
 - Có một hay nhiều ngõ vào
 - Có một hay nhiều ngõ ra
 - Có đặc tả chức năng mô tả chi tiết giá trị mỗi ngõ ra cho mọi tổ hợp giá trị của tất cả ngõ vào
 - Có đặc tả định thời mô tả thời gian lan truyền (thời gian tối thiểu mà ngõ ra sẽ ổn định và hợp lệ khi 1 ngõ vào thay đổi)
- Ví dụ: AND, OR, NOT, XOR, NAND, NOR, XNOR



IV. MẠCH SỐ

C. MẠCH TỔ HỢP

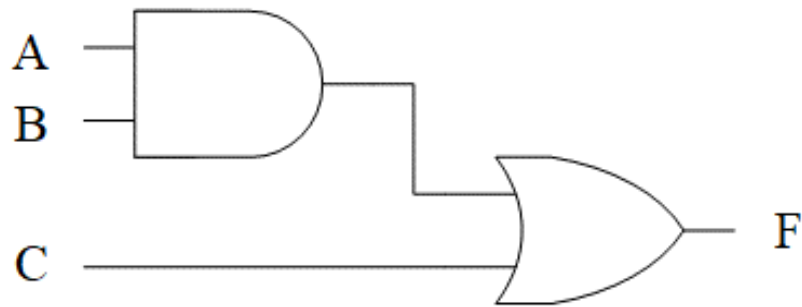
- Mạch tổ hợp là mạch chỉ chứa các thiết bị tổ hợp được kết nối với nhau và **không tồn tại hồi tiếp**
- Một thiết bị tổ hợp cũng được xem là một mạch tổ hợp
- Hồi tiếp: Ngõ ra được dùng như ngõ vào để **tính toán lại** ngõ ra



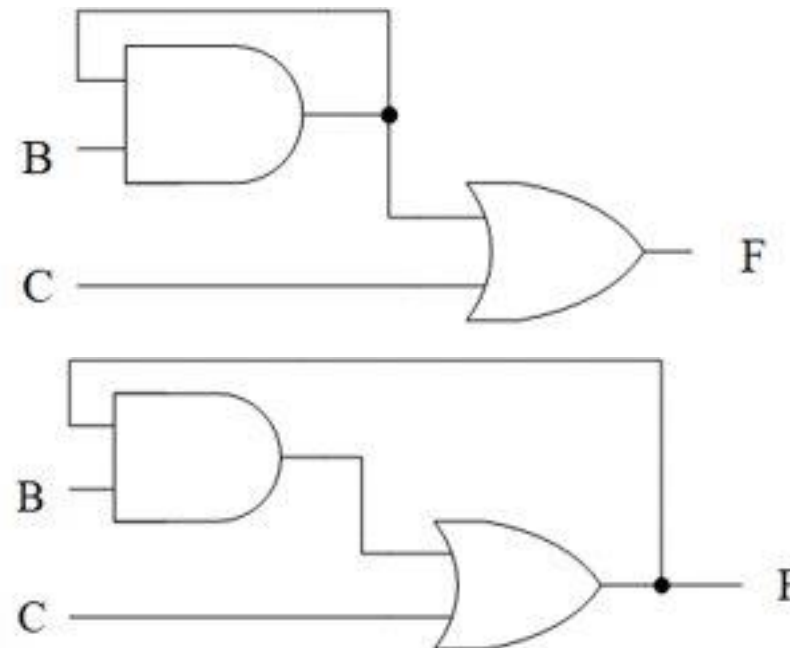
IV. MẠCH SỐ

C. MẠCH TỔ HỢP

Mạch tổ hợp



Mạch có hồi tiếp

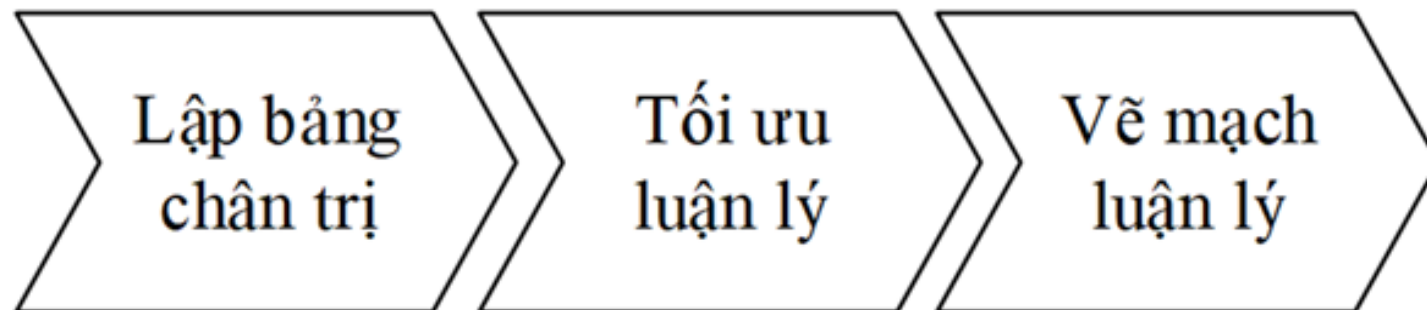


Sharing is learning

IV. MẠCH SỐ

C. MẠCH TỔ HỢP - THIẾT KẾ MẠCH

- Mục tiêu: Có được bản vẽ sơ đồ mạch tối ưu nhất về diện tích
 - Hàm Boolean ánh xạ ngõ ra phải có biểu thức tối ưu nhất => tối ưu luận lý dựa theo bảng chân trị

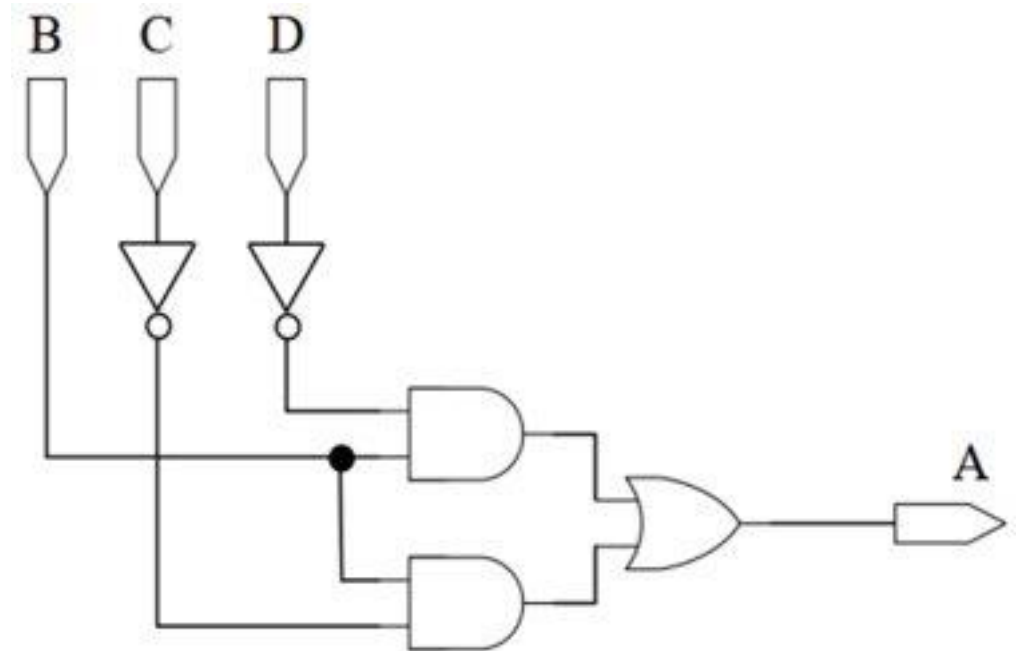
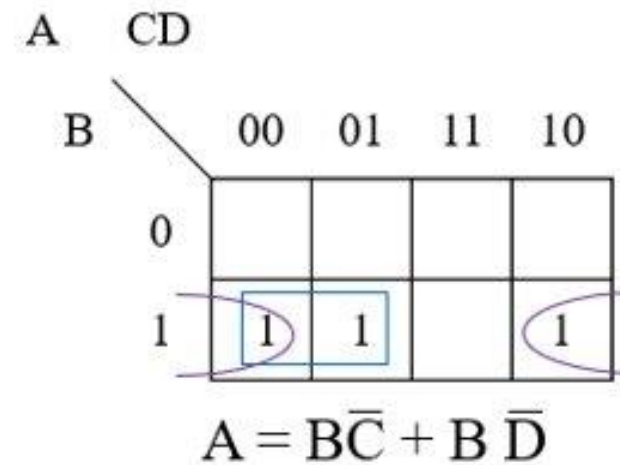


IV. MẠCH SỐ

C. MẠCH TỔ HỢP - THIẾT KẾ MẠCH

- Bài toán: Thiết kế mạch báo động (A = 1) cho lái xe với các tình huống: Bugi bật (B = 1) và cửa mở (C = 0), hoặc chưa cài dây an toàn (D = 0) và bugi bật (B = 1)

B	C	D	A
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	



IV. MẠCH SỐ

D. MẠCH TUẦN TỰ

- Vấn đề của mạch tổ hợp:
 - Số lượng tín hiệu cần xử lý tăng lên
 - ✓ Tăng số lượng ngõ vào để nhận tín hiệu
 - ✓ Mạch trở nên phức tạp và khó để hiện thực
- Mạch tuần tự: Nhập nhiều tín hiệu cho 1 ngõ vào bằng cách “chia ca”



IV. MẠCH SỐ

D. MẠCH TUẦN TỰ

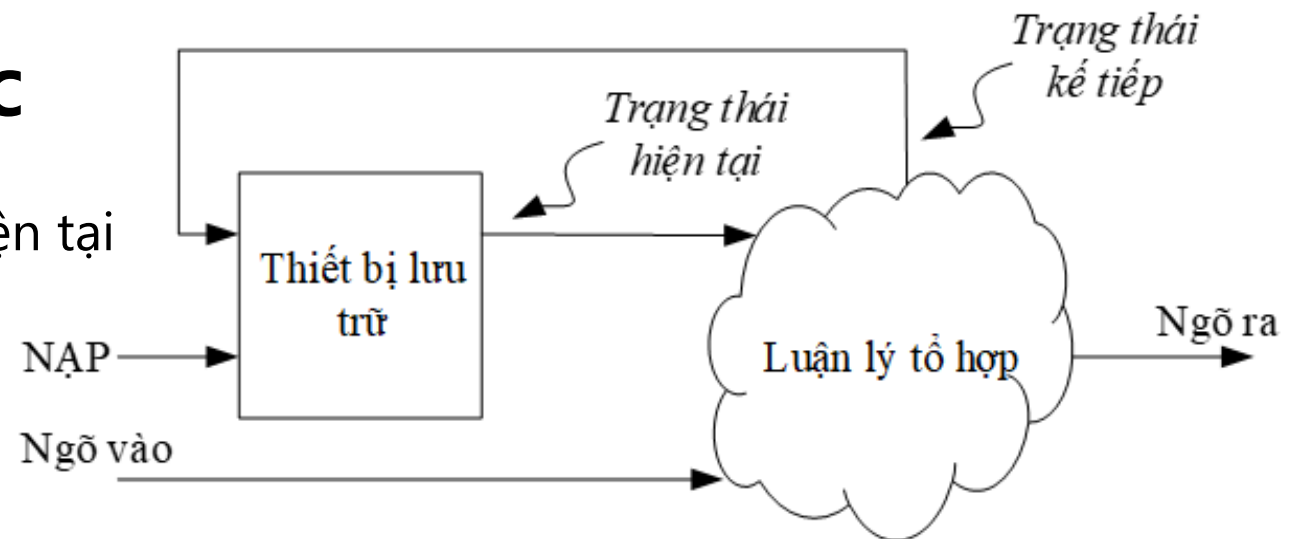
- Mạch tuần tự là mạch có trạng thái ngã ra không những **phụ thuộc vào tổ hợp các ngã vào** mà còn **phụ thuộc trạng thái ngã ra trước đó**. Ta nói mạch tuần tự có **tính nhớ**



IV. MẠCH SỐ

D. MẠCH TUẦN TỰ - CẤU TRÚC

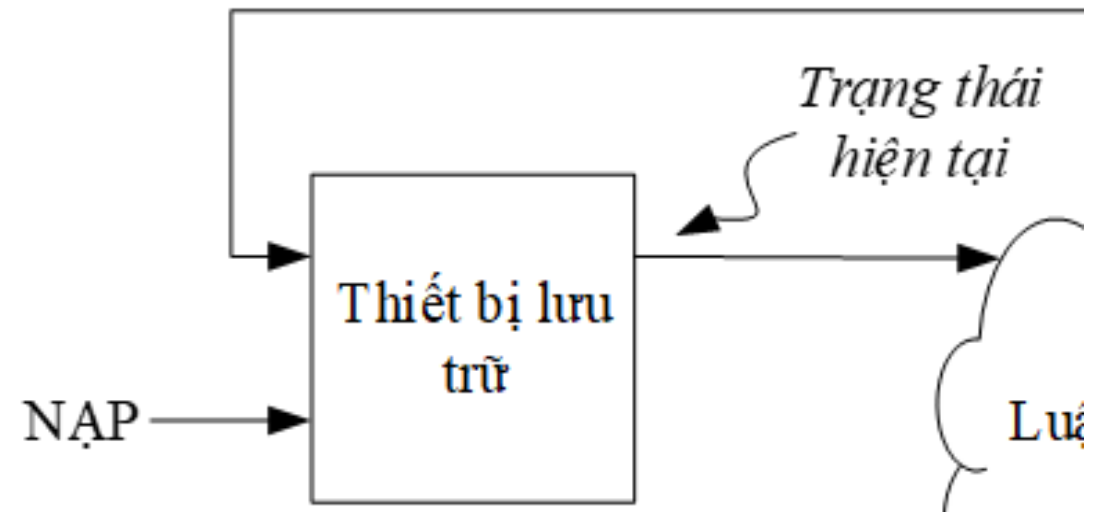
- Thiết bị lưu trữ: Lưu trữ trạng thái hiện tại
- Ngõ vào: Các ngõ vào
- Ngõ ra: Các ngõ ra
- NẠP: Ngõ vào điều khiển việc cập nhật giá trị
- Luận lý tổ hợp: Xử lý để xác định trạng thái kế tiếp và ngõ ra
- *Trạng thái* là tổ hợp các giá trị được lưu trong các thiết bị lưu trữ



IV. MẠCH SỐ

E. THIẾT BỊ LƯU TRỮ - CẤU TRÚC

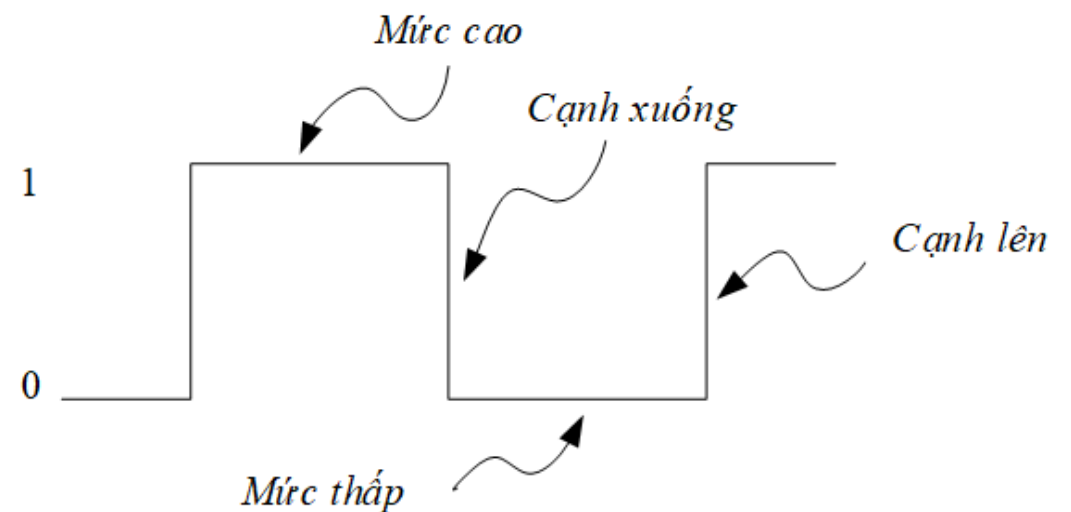
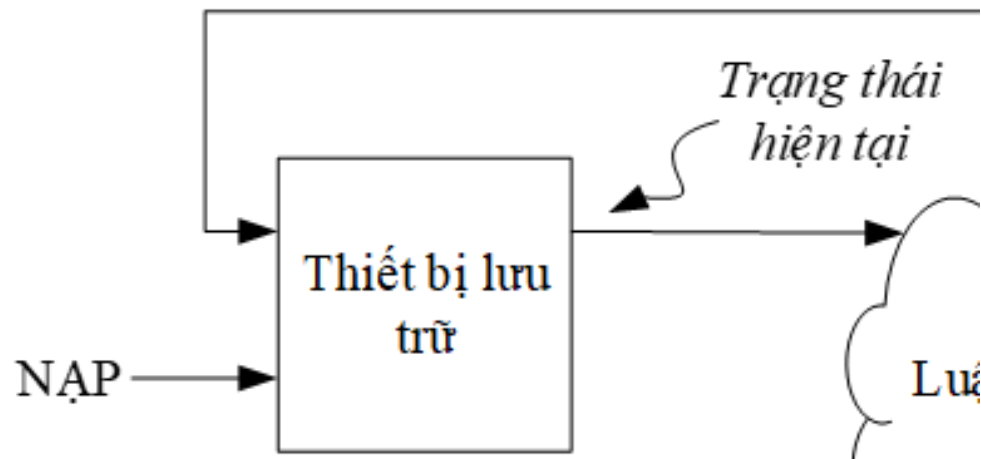
- Có ít nhất 1 ngõ ra mang giá trị mà chúng lưu trữ (Q)
- Có ít nhất 1 ngõ vào để thu nhận dữ liệu hoặc điều khiển
- Có 1 ngõ vào NẠP để điều khiển việc cập nhật giá trị (E, CLK)



IV. MẠCH SỐ

E. THIẾT BỊ LƯU TRỮ - CẤU TRÚC

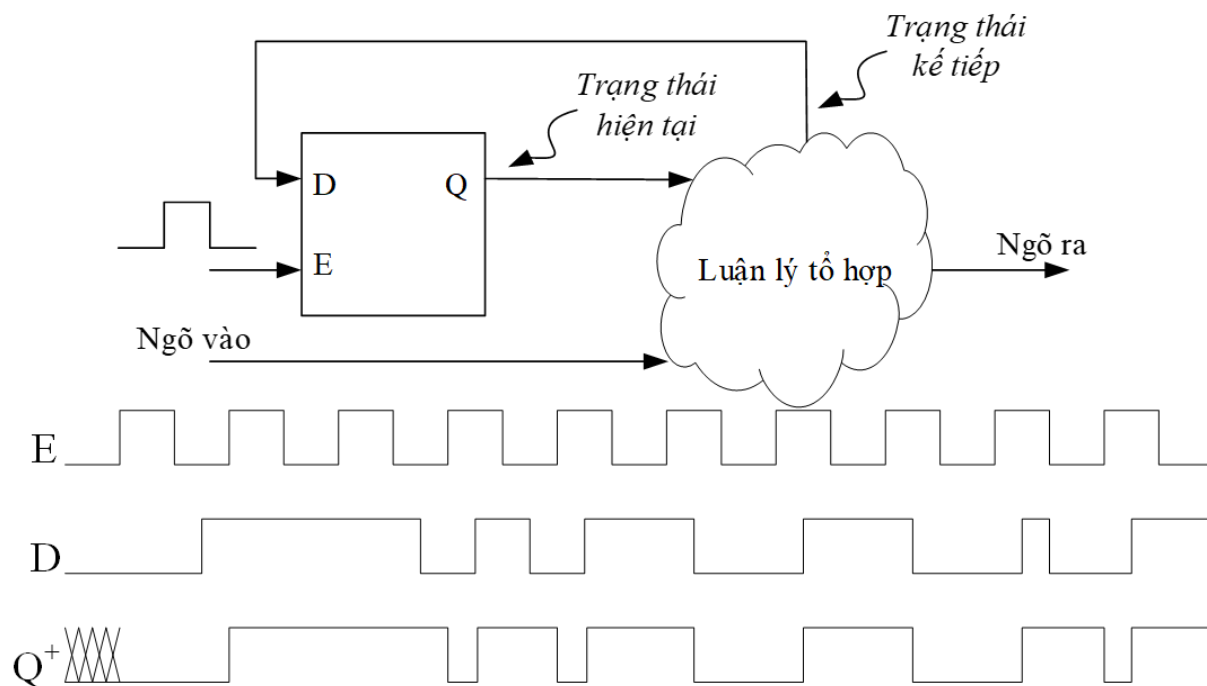
- Có 1 ngõ vào NẠP để điều khiển việc cập nhật giá trị (E, CLK)
- Đọc và ghi theo tín hiệu điều khiển



IV. MẠCH SỐ

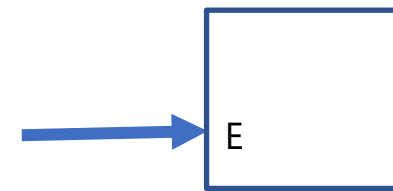
E. THIẾT BỊ LƯU TRỮ - MỘT SỐ THIẾT BỊ LƯU TRỮ

- Latch D: Tích cực theo mức, có khả năng lưu trữ 1 bit thông tin

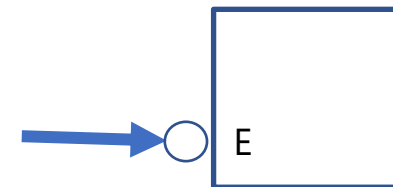


E	Q ⁺
0	Q
1	D

E	D	Q	Q ⁺
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



Tích cực mức cao



Tích cực mức thấp



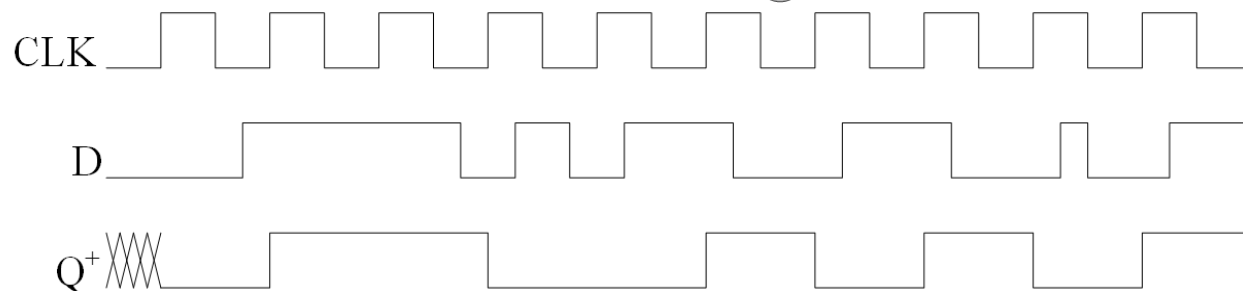
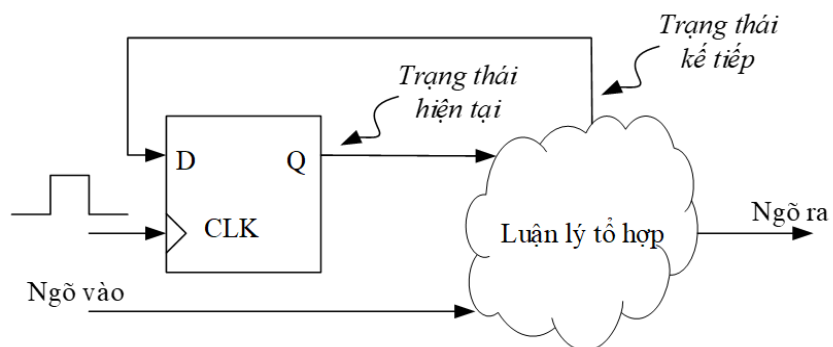
Sharing is learning

IV. MẠCH SỐ

E. THIẾT BỊ LƯU TRỮ - MỘT SỐ THIẾT BỊ LƯU TRỮ

- D Flipflop : Tích cực theo cạnh, có khả năng lưu trữ 1 bit thông tin

CLK	Q ⁺	CLK	D	Q	Q ⁺
-	Q	-	0	0	0
↑	D	-	0	1	1
		-	1	0	0
		-	1	1	1
		↑	0	0	0
		↑	0	1	0
		↑	1	0	1
		↑	1	1	1



→ CLK
Tích cực cạnh lên

○→ CLK
Tích cực cạnh xuống



Sharing is learning

IV. MẠCH SỐ

E. THIẾT BỊ LƯU TRỮ - MỘT SỐ THIẾT BỊ LƯU TRỮ

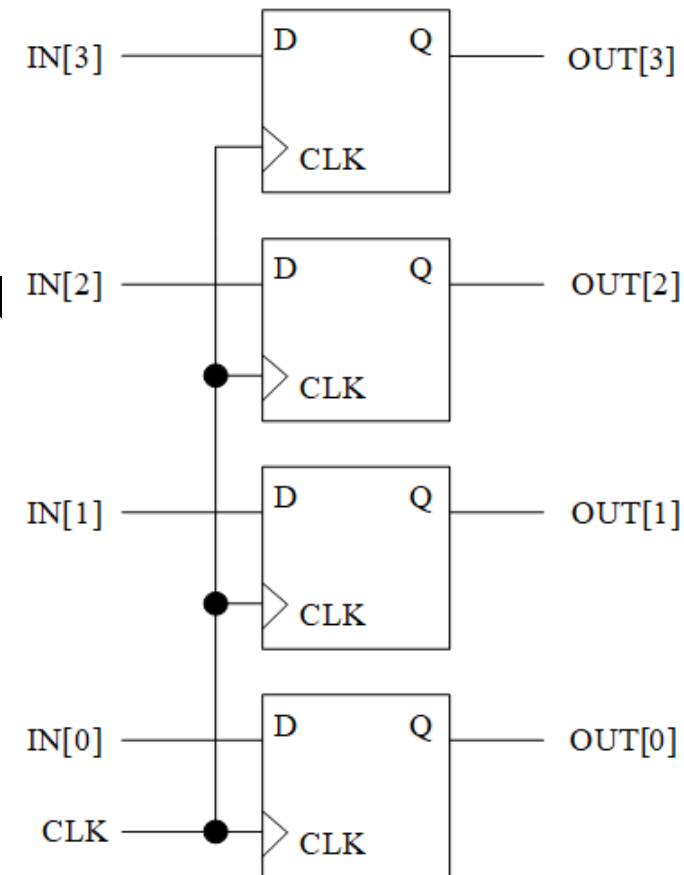
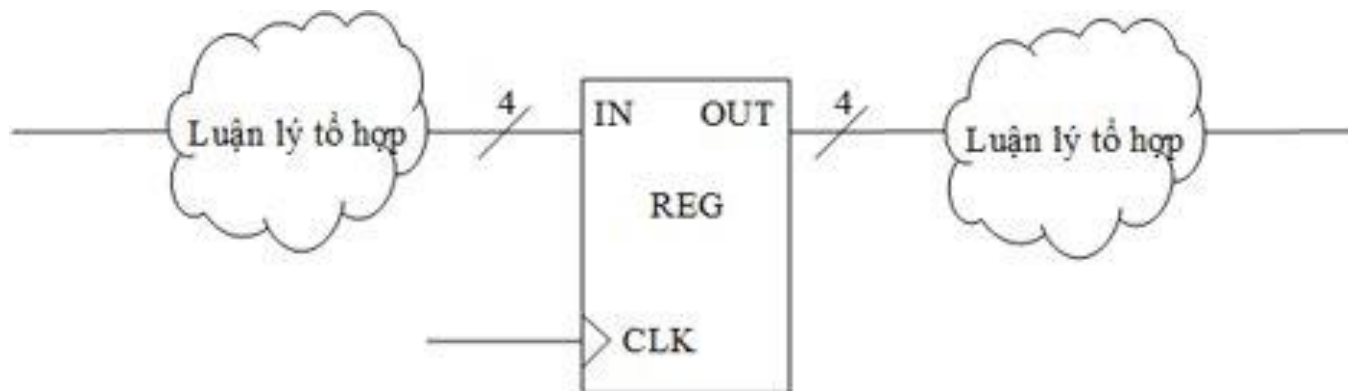
- Một Flipflop thường có:
 - Một hoặc hai ngõ vào dữ liệu, một ngõ vào xung CK và có thể có các ngõ vào với các chức năng khác.
 - Hai ngõ ra, thường được ký hiệu là Q (ngõ ra chính) và Q' (ngõ ra phụ)
 - Người ta gọi tên các Flipflop khác nhau bằng cách dựa vào tên các ngõ vào dữ liệu của chúng.
- Người ta thường dùng trạng thái của ngõ ra chính để chỉ trạng thái của FF. Nếu hai ngõ ra trạng thái giống nhau ta nói Flipflop ở trạng thái cấm.
- Flipflop chịu tác động của xung CLK còn Latch thì không.



IV. MẠCH SỐ

E. THIẾT BỊ LƯU TRỮ - MỘT SỐ THIẾT BỊ LƯU TRỮ

- Register (thanh ghi): các flipflop nối chung ngõ vào CLK
 - Có thể bổ sung khối luận lý tổ hợp để xử lý



V. MẠCH SỐ TRONG MÁY TÍNH

A. ALU (Arithmetic & Logic Unit)

- Một mạch tổ hợp có chức năng **thực hiện các phép toán số học và luận lý**
- Dựa theo **tín hiệu điều khiển Opcode**
=> thực hiện thao tác tương ứng theo như bảng hoạt động

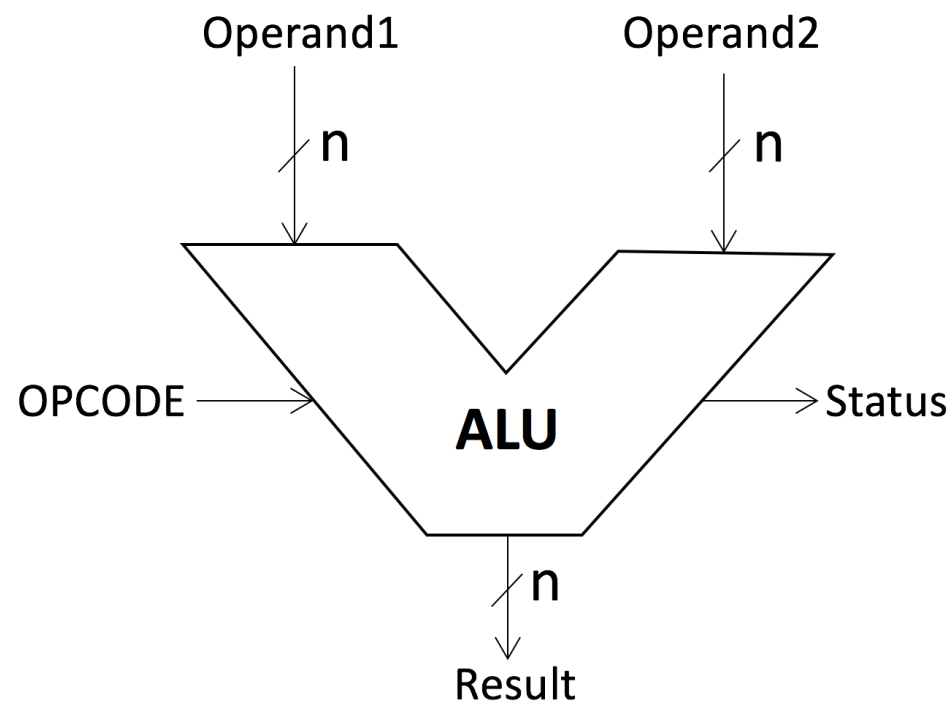
Opcode[2:0]]	Phép toán
000	$A + B$
001	$A + 1$
010	$A - B$
011	$A - 1$
100	$A \& B$
101	$A B$
110	A'
111	$A \text{ XOR } B$



V. MẠCH SỐ TRONG MÁY TÍNH

A. ALU (Arithmetic & Logic Unit)

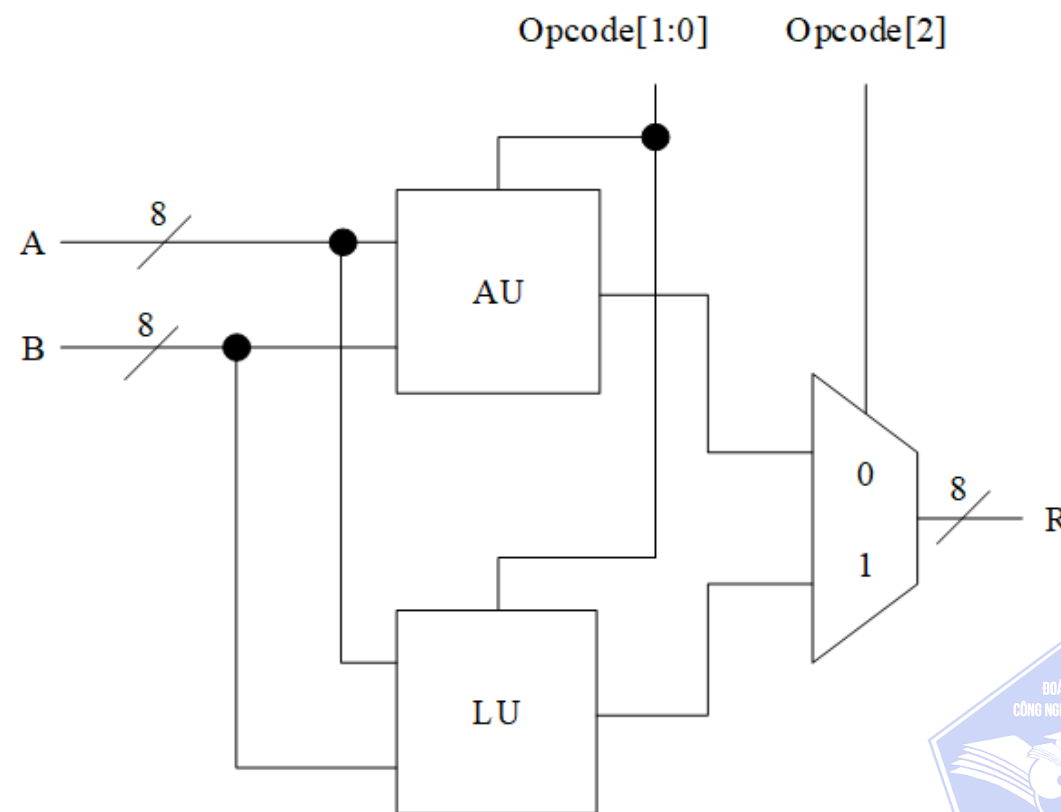
- Một mạch tổ hợp có chức năng **thực hiện các phép toán số học và luận lý**
- Dựa theo **tín hiệu điều khiển Opcode**
=> thực hiện thao tác tương ứng theo như bảng hoạt động



V. MẠCH SỐ TRONG MÁY TÍNH

A. ALU (Arithmetic & Logic Unit)

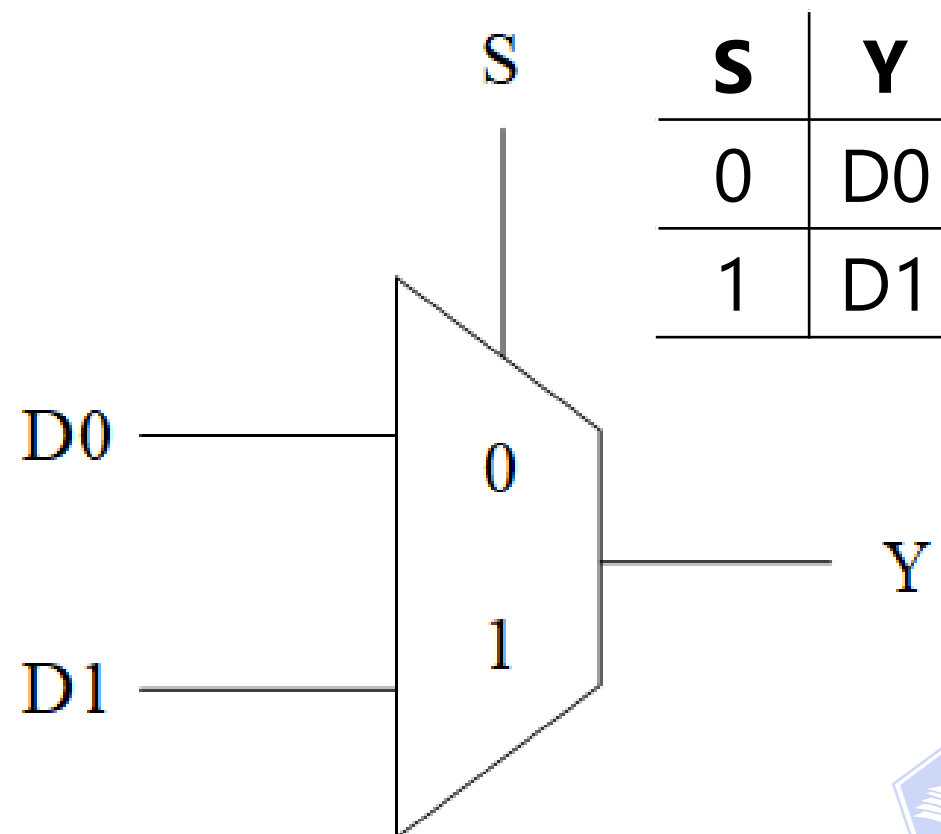
- 4 thao tác đầu là số học và 4 thao tác sau là luận lý => có thể chia ALU thành 2 mạch con là AU và LU
- Lựa chọn giữa kết quả số học (AU) hay luận lý (LU) sẽ được quyết định bởi bộ chọn



V. MẠCH SỐ TRONG MÁY TÍNH

B. BỘ CHỌN (MUX)

- Một **mạch tổ hợp** có chức năng **lựa chọn**
- Nhiều ngõ vào -> Dựa theo ngõ vào điều khiển -> Một ngõ ra



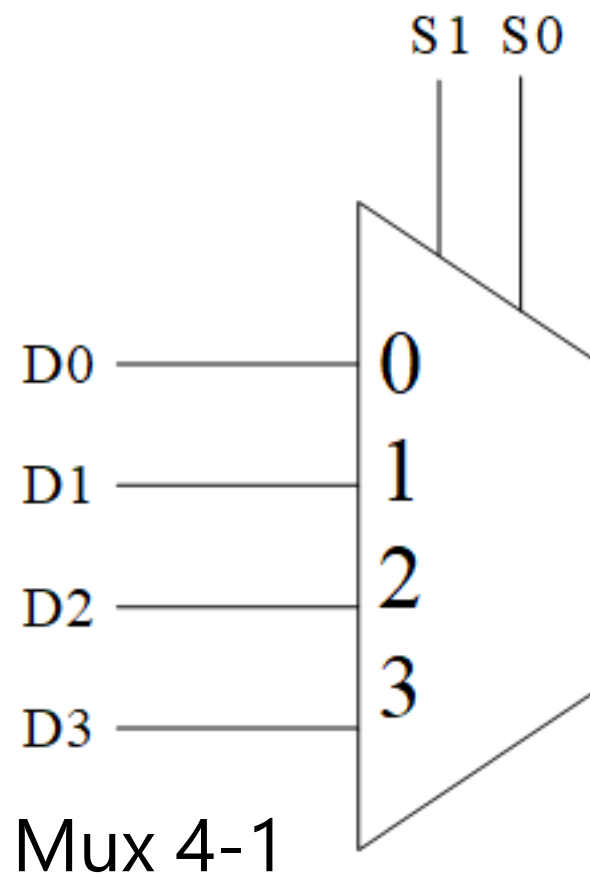
Mux 2-1



V. MẠCH SỐ TRONG MÁY TÍNH

B. BỘ CHỌN (MUX)

- Một **mạch tổ hợp** có chức năng **lựa chọn**
- Nhiều ngõ vào -> Dựa theo ngõ vào điều khiển -> Một ngõ ra



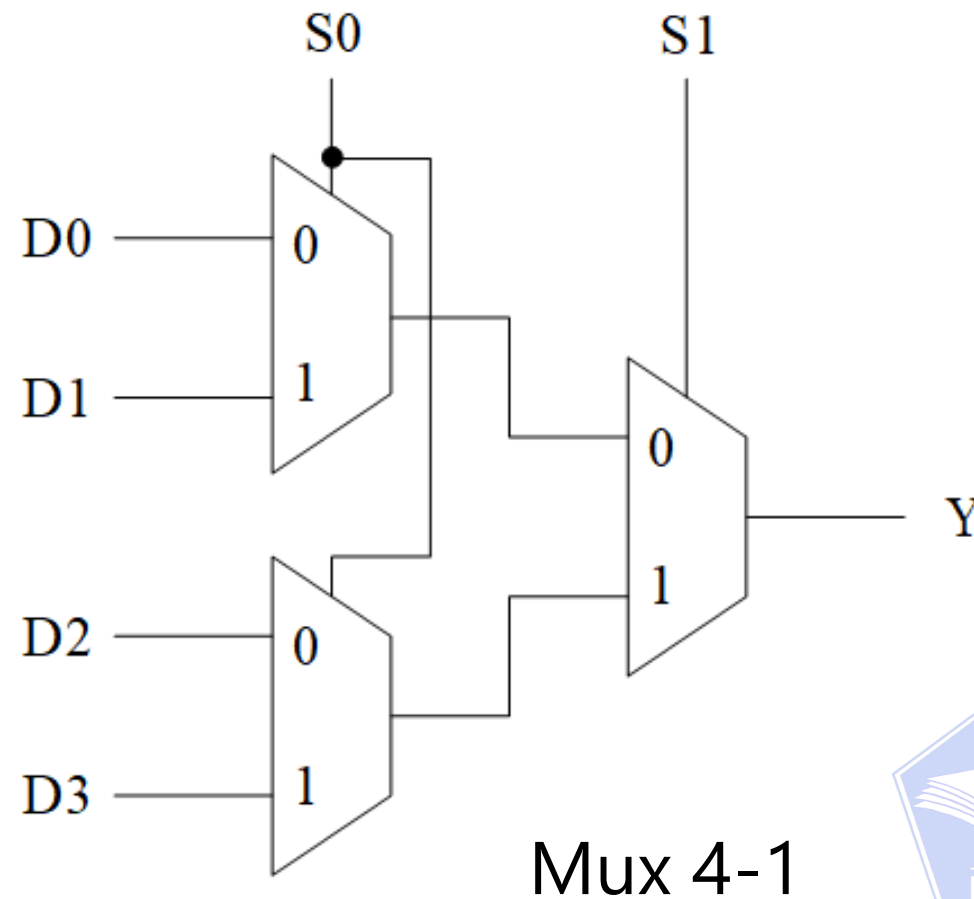
S1	S0	Y
0	0	D0
0	1	D1
1	0	D2
1	1	D3



V. MẠCH SỐ TRONG MÁY TÍNH

B. BỘ CHỌN (MUX)

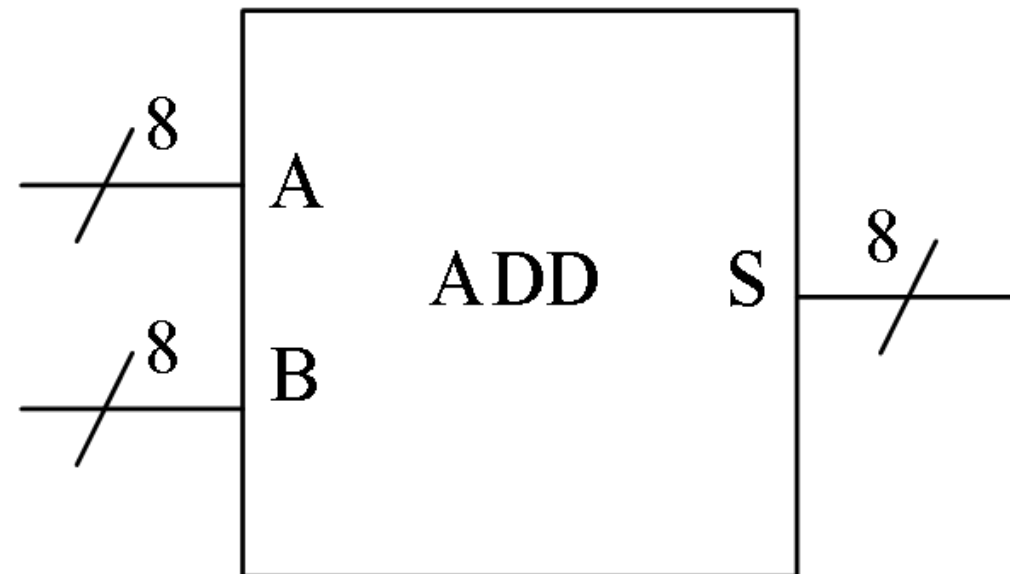
- Để có khả năng chọn 1 trong 4 thì chúng ta có thể kết nối các mux2 lại để có được mux4
- Tương tự, ta có thể thiết kế mux16, mux8...từ mux4, mux2



V. MẠCH SỐ TRONG MÁY TÍNH

C. BỘ CỘNG (ADDER)

- Một mạch tổ hợp có chức năng thực hiện **phép toán cộng số học**.
- Về chi tiết, bộ cộng sẽ thực hiện **cộng từng bit** với nhau

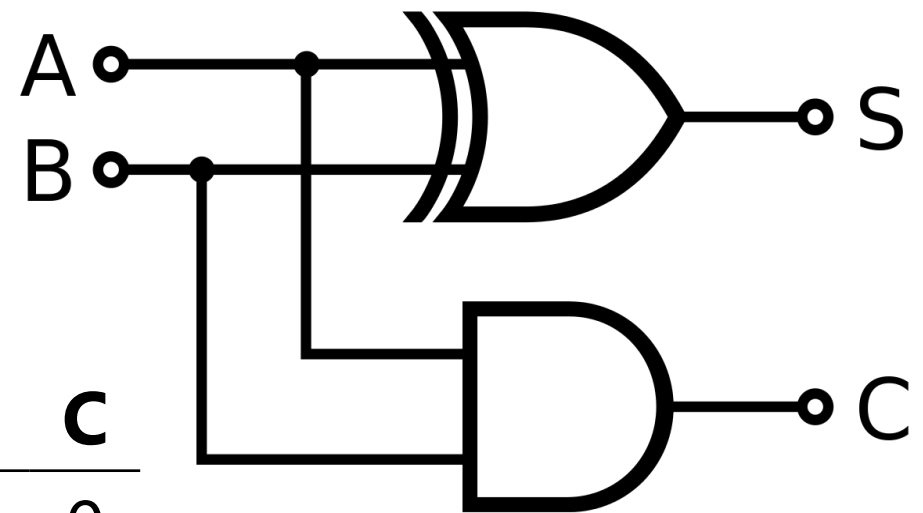


V. MẠCH SỐ TRONG MÁY TÍNH

C. BỘ CỘNG (ADDER)

- Half Adder (cộng bán phần)
 - Input: Bit A, bit B
 - Output: Tổng Sum, bit nhớ Carry
 - Không thể ứng dụng làm bộ cộng nhiều bit vì **đầu vào không có bit nhớ**

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

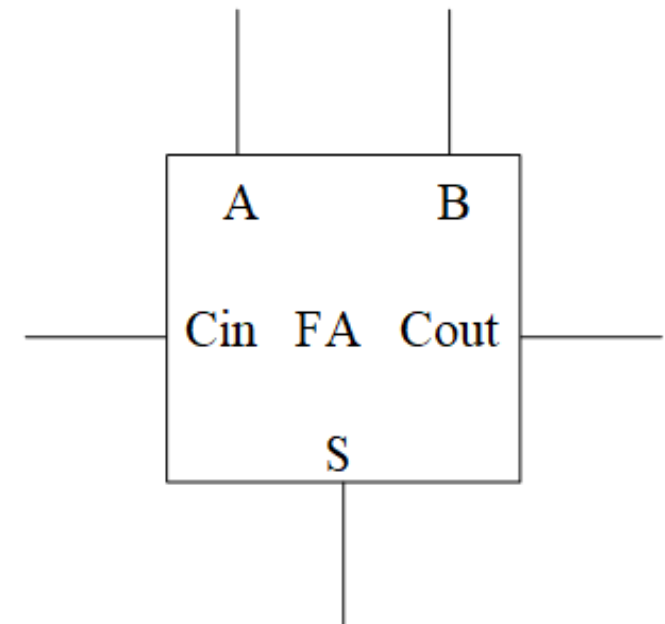


V. MẠCH SỐ TRONG MÁY TÍNH

C. BỘ CỘNG (ADDER)

- Full Adder (cộng toàn phần)
 - Input: Bit A, bit B, bit nhớ C_{in}
 - Output: Tổng Sum, bit nhớ C_{out}
 - Ứng dụng làm bộ cộng nhiều bit

C_{in}	A	B	S	C_{out}
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

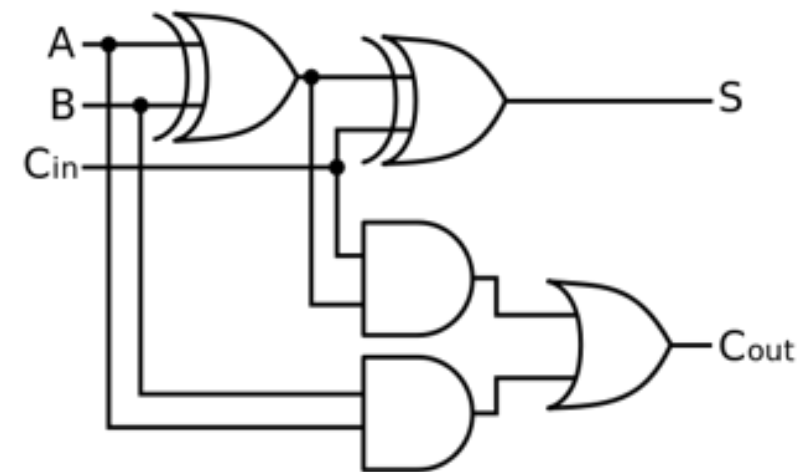


V. MẠCH SỐ TRONG MÁY TÍNH

C. BỘ CỘNG (ADDER)

- Full Addder (cộng toàn phần)
 - Input: Bit A, bit B, bit nhớ C_{in}
 - Output: Tổng Sum, bit nhớ C_{out}
 - Ứng dụng làm bộ cộng nhiều bit

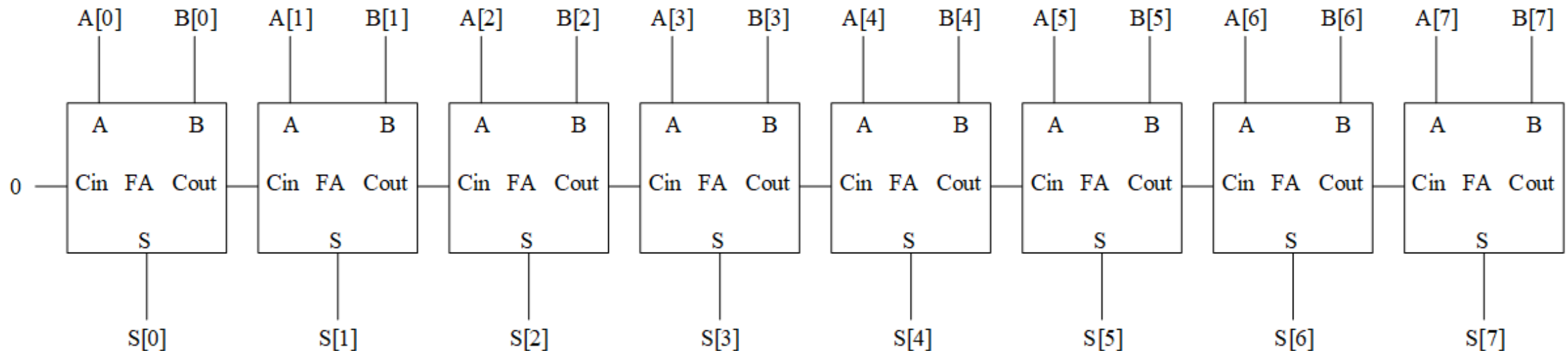
C_{in}	A	B	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



V. MẠCH SỐ TRONG MÁY TÍNH

C. BỘ CỘNG (ADDER)

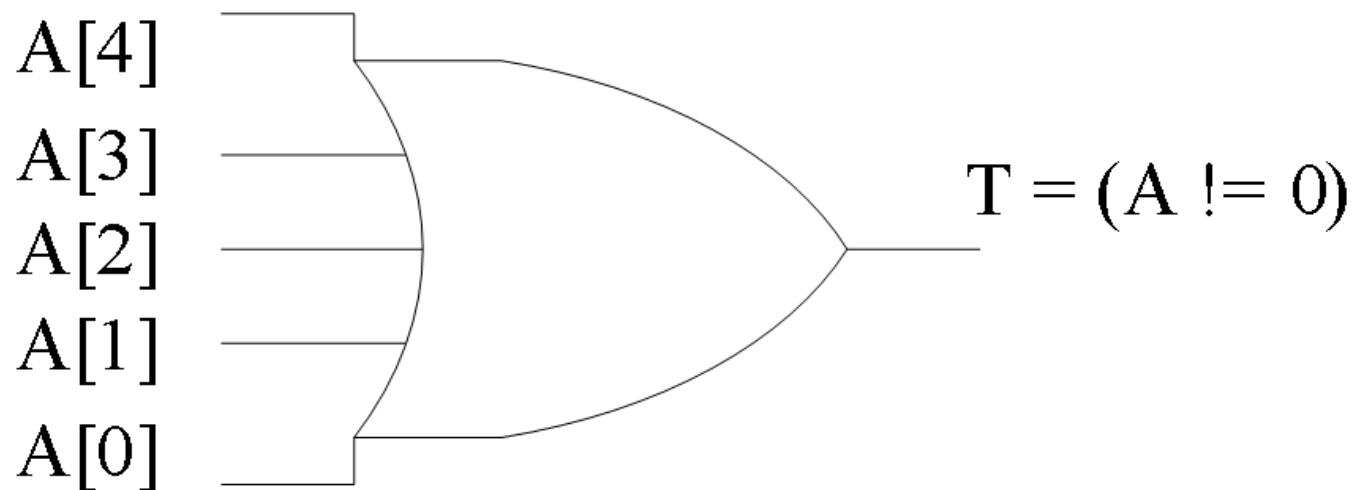
- Bộ cộng 2 số 8 bit
 - Bộ 8 bộ cộng toàn phần



V. MẠCH SỐ TRONG MÁY TÍNH

D. BỘ SO SÁNH

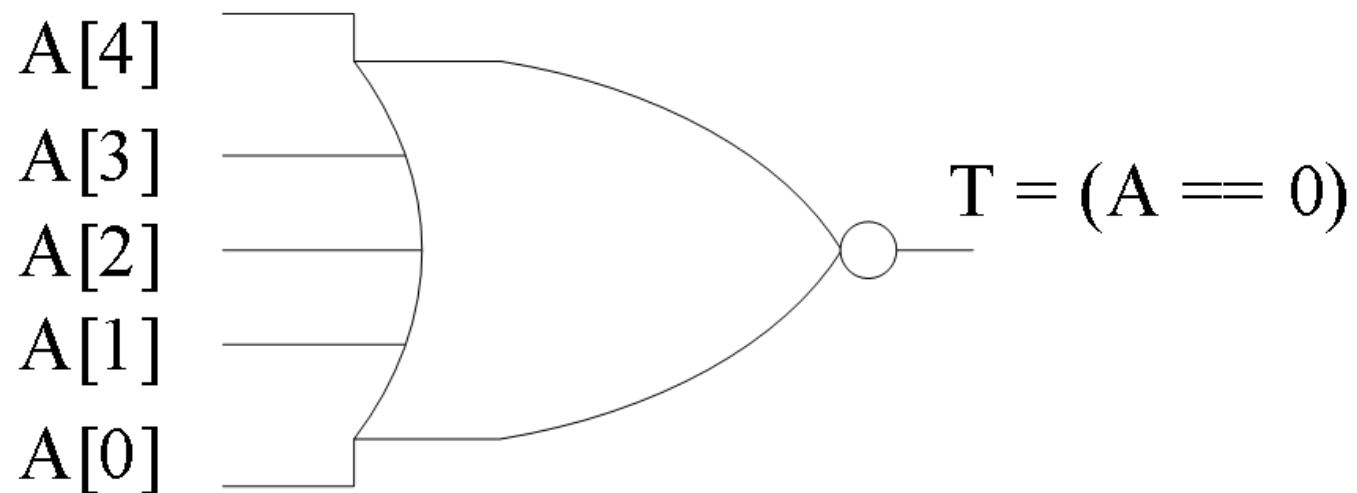
- So sánh với 0: $T = 1 \Rightarrow A \neq 0$



V. MẠCH SỐ TRONG MÁY TÍNH

D. BỘ SO SÁNH

- So sánh với 0: $T = 1 \Rightarrow A == 0$



V. MẠCH SỐ TRONG MÁY TÍNH

D. BỘ SO SÁNH

- So sánh với 2 số bất kì
- Việc thiết kế bộ so sánh 2 số bất kỳ tốn rất nhiều tài nguyên vì bảng chân trị cần tới 2^{2n} hàng, với n là độ rộng bit của toán hạng.
- Giải pháp được đề xuất là tận dụng ALU và bộ so sánh có sẵn



V. MẠCH SỐ TRONG MÁY TÍNH

D. BỘ SO SÁNH

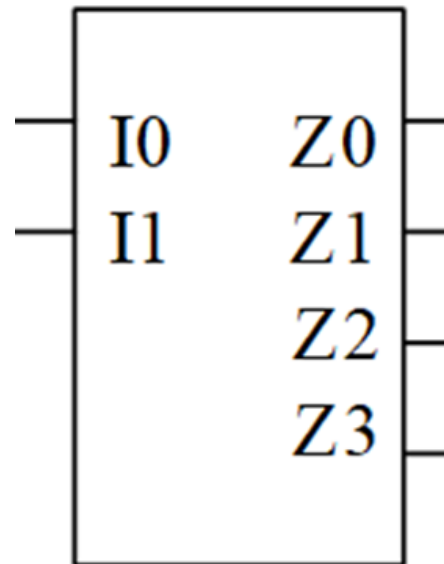
- So sánh 2 số A và B (8 bit) bất kỳ
- Dùng ALU để tính $A - B = Y$
 - Xét $Y = 0$ thì $A = B$
 - MSB (bit ở cực trái – bit dấu) $Y = 1$ thì $A < B$
 - MSB $Y = 0$ thì $A > B$



V. MẠCH SỐ TRONG MÁY TÍNH

E. BỘ GIẢI MÃ

- Một mạch tổ hợp có chức năng chuyển thông tin nhị phân từ các ngõ vào tới từng ngõ ra
- Dùng để quản lý thiết bị (lựa chọn thiết bị được hoạt động)



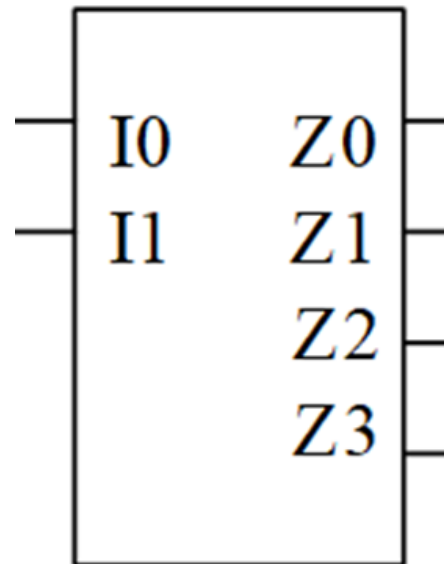
I1	I0	Z3	Z2	Z1	Z0
0	0				1
0	1			1	
1	0		1		
1	1	1			



V. MẠCH SỐ TRONG MÁY TÍNH

E. BỘ GIẢI MÃ

- Một bộ giải mã 2:4
- 2 ngõ vào => 4 tổ hợp => quản lý địa chỉ của 4 thiết bị
- Tại **1** thời điểm chỉ có **1** ngõ ra được **tích cực (1)** nên chỉ có **1 thiết bị** được lựa chọn để hoạt động



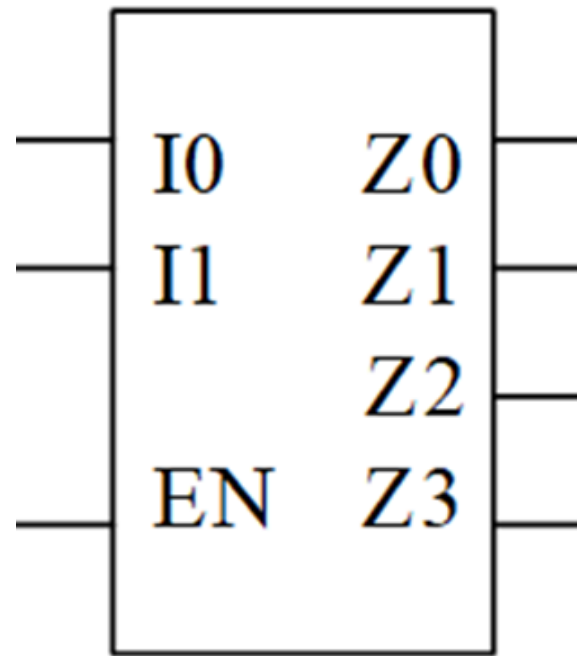
I1	I0	Z3	Z2	Z1	Z0
0	0				1
0	1			1	
1	0		1		
1	1	1			



V. MẠCH SỐ TRONG MÁY TÍNH

E. BỘ GIẢI MÃ

- Có thể thêm ngõ **EN (Enable)** để lựa chọn trạng thái hoạt động/ không hoạt động của bộ giải mã



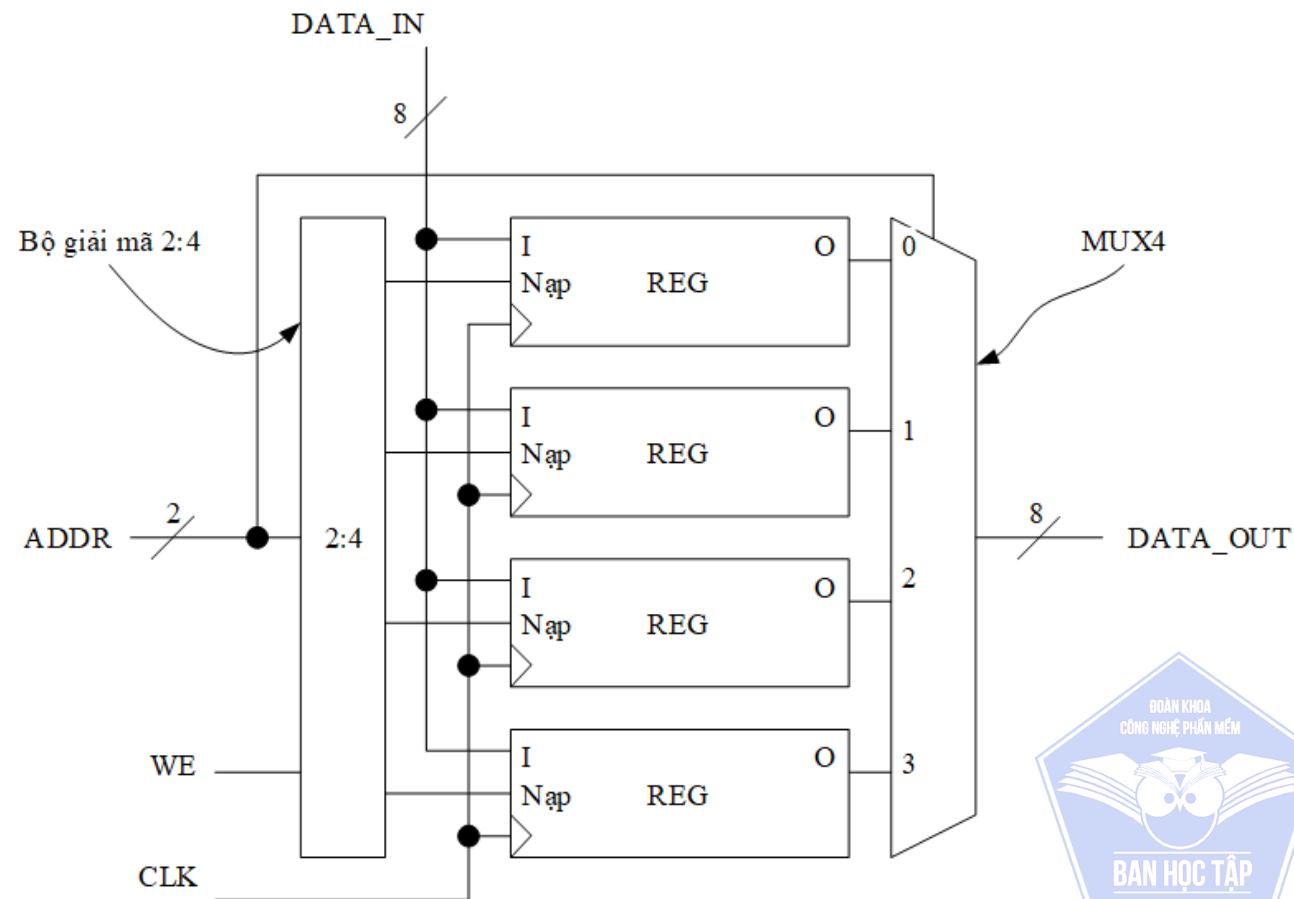
EN	I1	I0	Z3	Z2	Z1	Z0
0	X	X	0	0	0	0
1	0	0				1
1	0	1			1	
1	1	0		1		
1	1	1	1			



V. MẠCH SỐ TRONG MÁY TÍNH

F. TẬP THANH GHI

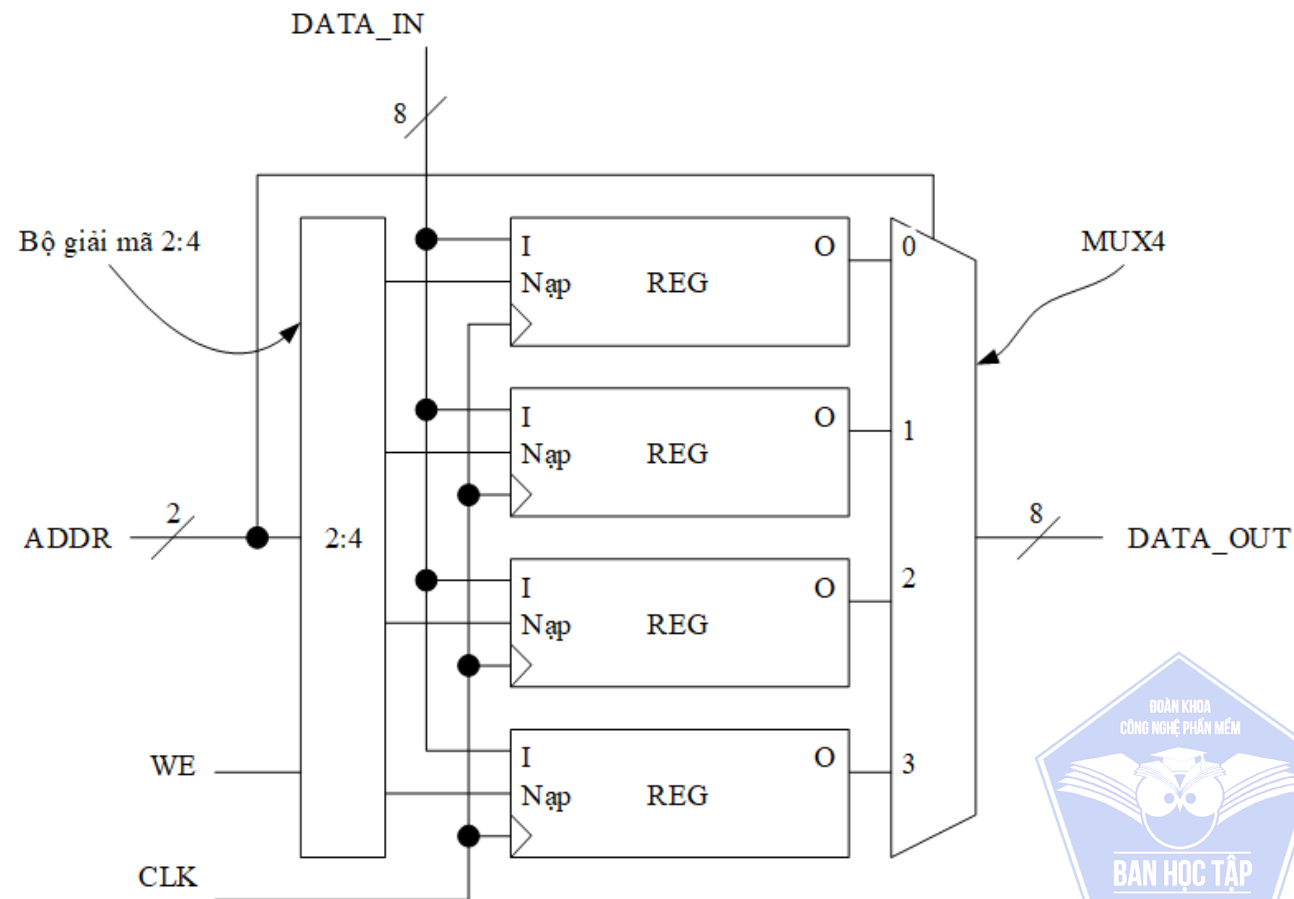
- Một bộ nhớ để lưu trữ dữ liệu tạm để được xử lý bởi các đơn vị xử lý (chẳng hạn như ALU)
- Cấu tạo: Mảng 1 chiều của các thanh ghi



V. MẠCH SỐ TRONG MÁY TÍNH

F. TẬP THANH GHI

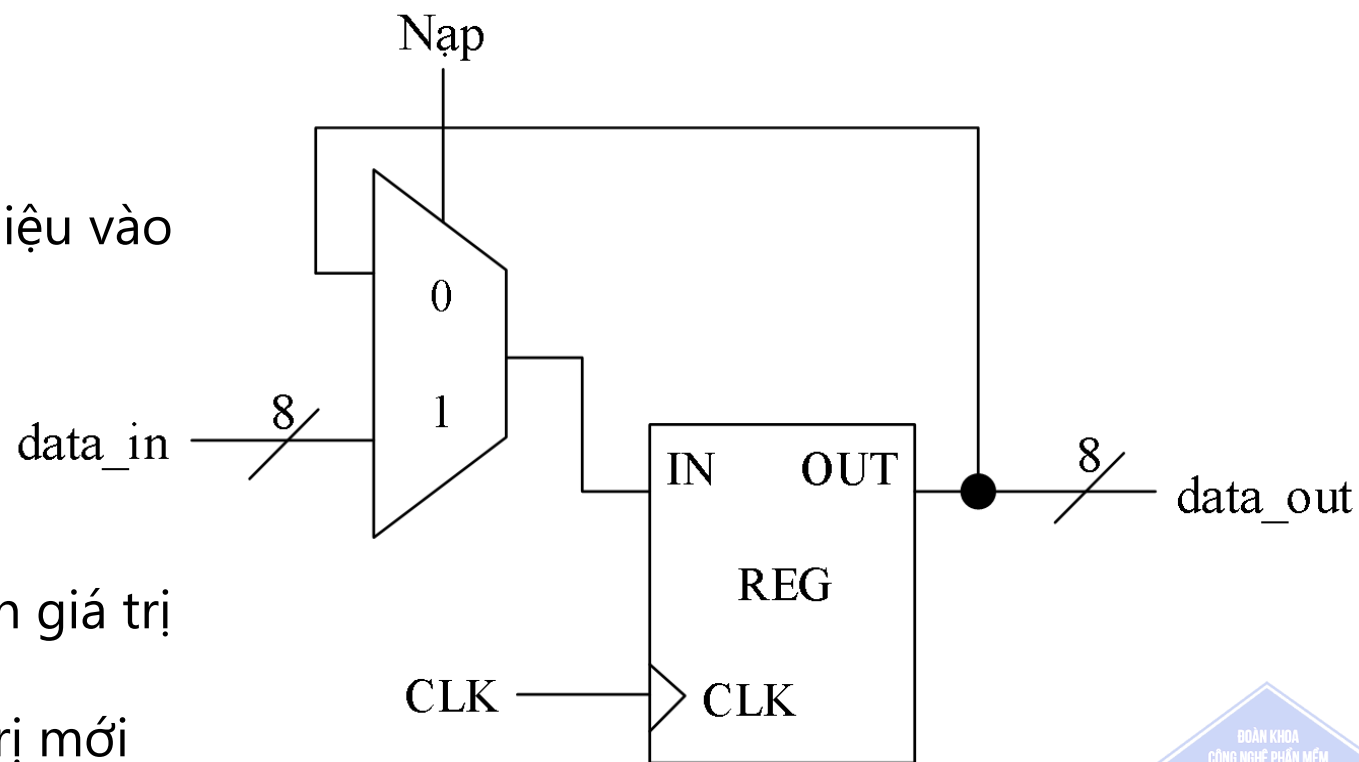
- **I và O:** ngõ vào và ngõ ra dữ liệu
- **Nạp (WE):** ngõ vào **cho phép ghi** dữ liệu vào thanh ghi hay không
- **Bộ giải mã:** chọn địa chỉ của thanh nào sẽ được thao tác
- **MUX4:** chọn thanh ghi được xuất dữ liệu ra bên ngoài



V. MẠCH SỐ TRONG MÁY TÍNH

F. TẬP THANH GHI

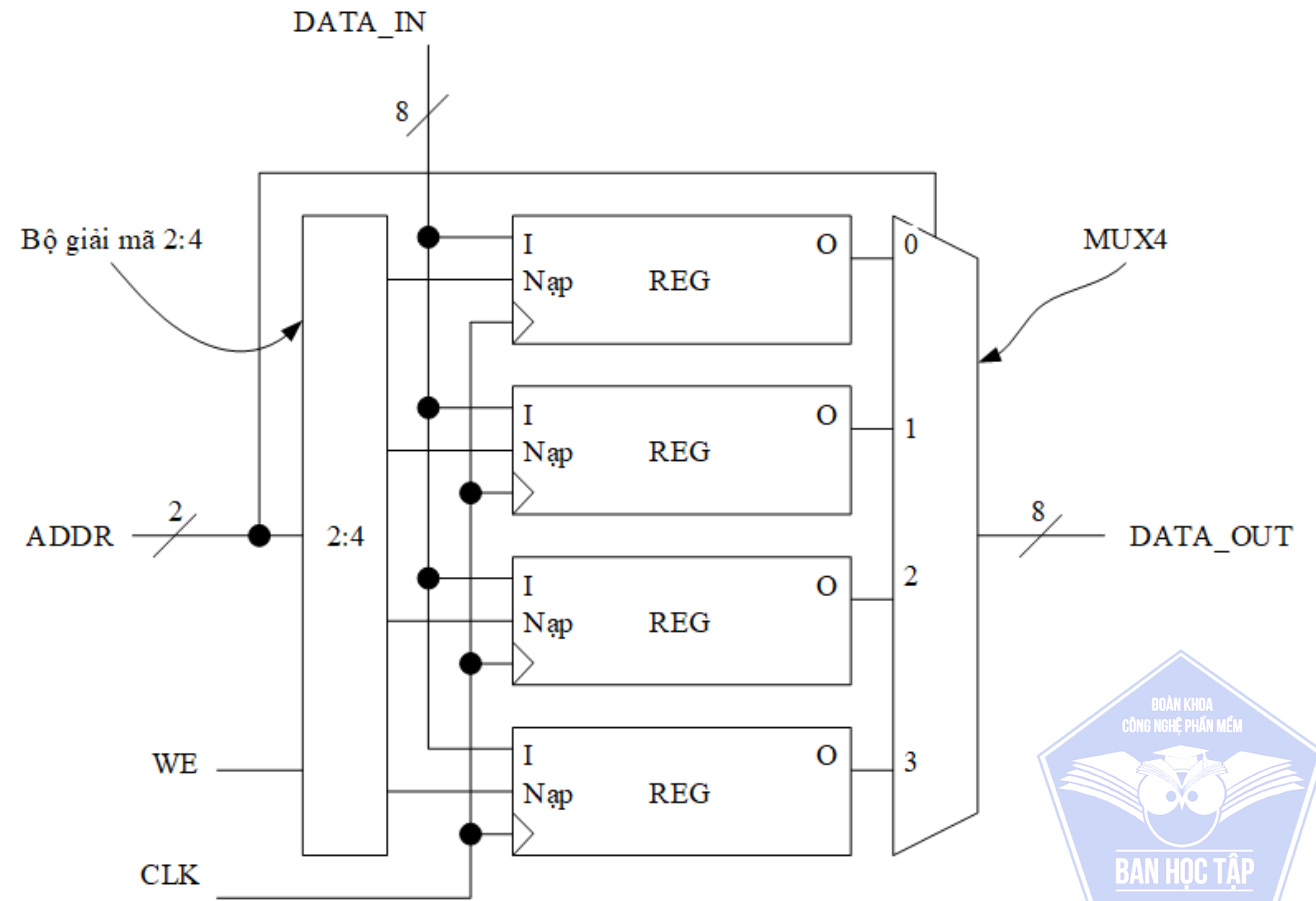
- **Nạp:** ngõ vào **cho phép ghi** dữ liệu vào thanh ghi hay không
- Nạp là một **bộ chọn**
- Nạp = 0 => thanh ghi giữ nguyên giá trị
- Nạp = 1 => thanh ghi nhận giá trị mới khi CLK tích cực



V. MẠCH SỐ TRONG MÁY TÍNH

F. TẬP THANH GHI

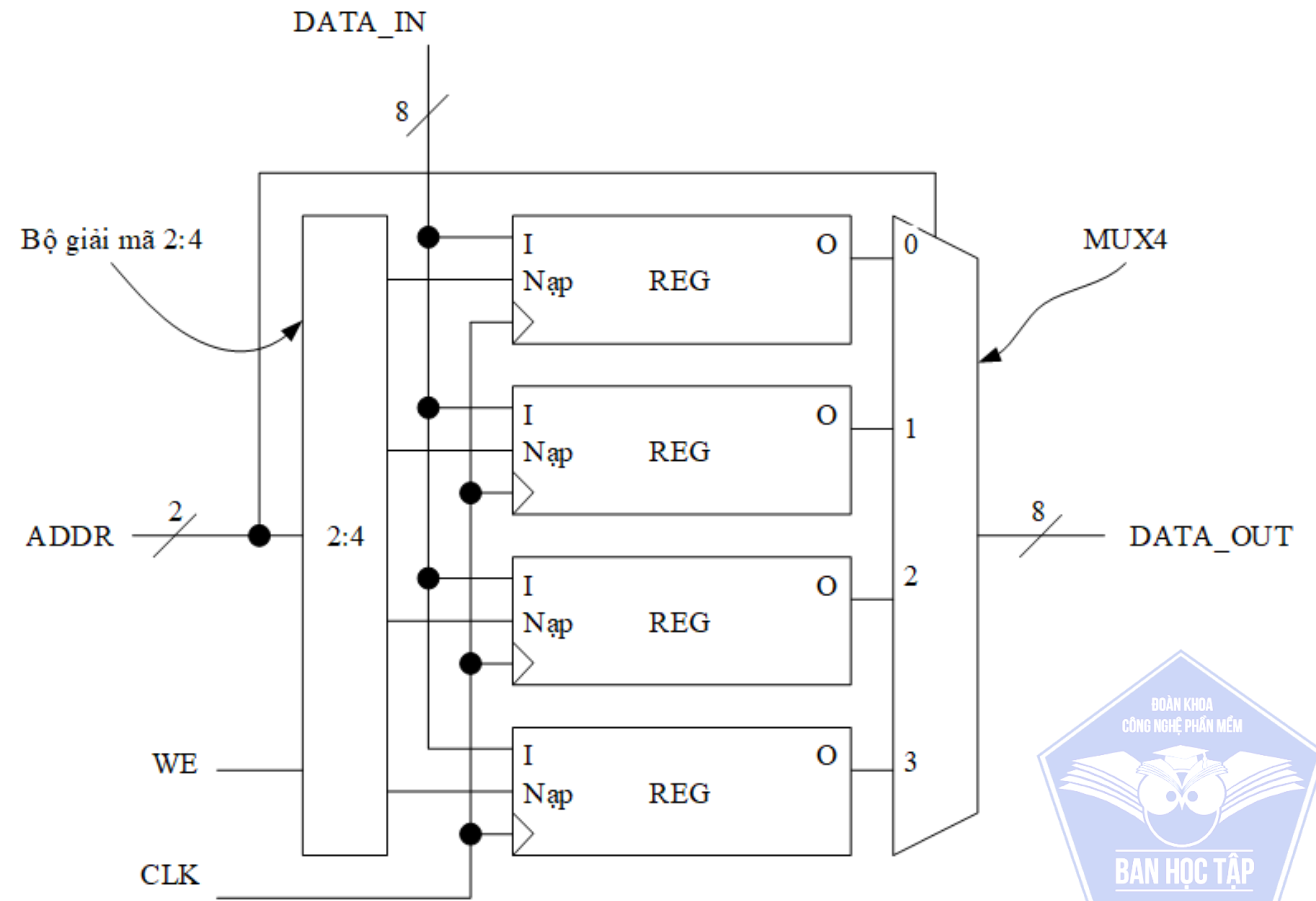
- Đọc dữ liệu từ một thanh ghi bất kỳ:
 - Thiết lập địa chỉ của thanh ghi cần đọc tới ADDR
 - Thiết lập WE (Nạp) = 0
 - ✓ Tránh không ghi dữ liệu tới thanh ghi muốn đọc khi CLK tích cực.



V. MẠCH SỐ TRONG MÁY TÍNH

F. TẬP THANH GHI

- Ghi dữ liệu tới một thanh ghi bất kỳ:
 - Thiết lập địa chỉ của thanh ghi cần ghi tới ADDR
 - Đặt dữ liệu vào (DATA_IN)
 - Thiết lập WE (Nạp) = 1 (ghi)
 - Dữ liệu sẽ được ghi khi CLK tích cực





ĐỀ THI THỬ



ĐIỂM DANH TRAINING OFFLINE

BAN HỌC TẬP CÔNG NGHỆ PHẦN MỀM

TRAINING GIỮA KỲ HỌC KỲ I NĂM HỌC 2022 – 2023



Sharing is learning

HẾT

**CẢM ƠN CÁC BẠN ĐÃ THEO DÕI
CHÚC CÁC BẠN CÓ KẾT QUẢ THI THẬT TỐT!**

 **BAN HỌC TẬP**

Khoa Công nghệ Phần mềm

Trường Đại học Công nghệ Thông tin

Đại học Quốc gia thành phố Hồ Chí Minh

 **CONTACT**

bht.cnpm.uit@gmail.com

fb.com/bhtcnpm

fb.com/groups/bht.cnpm.uit