

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN TOÁN ỨNG DỤNG & TIN HỌC



Bài giảng

KIẾN TRÚC MÁY TÍNH

Giảng viên : Phạm Huyền Linh
Bộ môn : Toán Tin

Kiến trúc máy tính



CHƯƠNG 2

HỆ ĐẾM & LOGIC SỐ

(Number Systems & Digital Logic)

Giảng viên: Phạm Huyền Linh

Nội dung



2.1. Các hệ đếm cơ bản (Number Systems)

2.2. Đại số Boole (Boolean Algebra)

2.3. Cổng Logic (Logic Gate)

2.4. Mạch tổ hợp (Combinational Circuits)

2.5. Mạch tuần tự (Sequential Circuits)

2.1. Các hệ đếm cơ bản

- Hệ thập phân (Decimal system)
- Hệ nhị phân (Binary system)
- Chuyển đổi giữa số nhị phân và số thập phân
(Converting between Binary and Decimal)
- Hệ mười sáu (Hexadecimal system)

Hệ thập phân



- Cơ số 10
- 10 chữ số: 0,1,2,3,4,5,6,7,8,9
- Dùng n chữ số thập phân \rightarrow biểu diễn 10^n giá trị

$$\underbrace{00\dots00}_n = 0$$

$$\underbrace{99\dots99}_n = 10^n - 1$$

- 1 chữ số: 0 \rightarrow 9 10 giá trị
- 2 chữ số: 0 \rightarrow 99 $100 = 10^2$ giá trị
- 3 chữ số: 0 \rightarrow 999 $1000 = 10^3$ giá trị

Hệ thập phân



- Biểu diễn số thập phân

$$X = x_n x_{n-1} \dots x_1 x_0, x_{-1} x_{-2} \dots x_{-m}$$

Giá trị

$$X = x_n 10^n + x_{n-1} 10^{n-1} + \dots + x_1 10^1 + x_0 10^0 + x_{-1} 10^{-1} + \dots + x_{-m} 10^{-m}$$

$$X = \sum_{i=-m}^n x_i 10^i$$

$$356.23 = 3 \cdot 10^2 + 5 \cdot 10^1 + 6 \cdot 10^0 + 2 \cdot 10^{-1} + 3 \cdot 10^{-2}$$

Hệ thập phân



$$356.23 = 3.10^2 + 5.10^1 + 6.10^0 + 2.10^{-1} + 3.10^{-2}$$

■ Phần nguyên 356

- $356 : 10 = 35$ dư 6
- $35 : 10 = 3$ dư 5
- $3 : 10 = 0$ dư 3



■ Phần lẻ .23

- $0.23 * 10 = 2.3$ -> phần nguyên là 2
- $0.3 * 10 = 3$ -> phần nguyên là 3



Hệ nhị phân



- Cơ số 2
- Chữ số: 0, 1
- Được sử dụng trong máy tính
- Chữ số nhị phân: Bit (binary digit) – đơn vị thông tin nhỏ nhất
- Dùng n bit số (n - chiều dài bit của số): 2^n giá trị

$$\underbrace{00\dots00}_n = 0$$

$$\underbrace{11\dots11}_n = 2^n - 1$$

Hệ nhị phân



Hệ đếm cơ số 2				Biểu diễn số thập phân
1 – bit	2 – bit	3 – bit	4 – bit	
0	00	000	0000	0
1	01	001	0001	1
	10	010	0010	2
	11	011	0011	3
		100	0100	4
		101	0101	5
		110	0110	6
		111	0111	7
			1000	8
			1001	9
			1010	10
			1011	11
			1100	12
			1101	13
			1110	14
			1111	15

Hệ nhị phân



- Tổng quát

$$X_{(2)} = x_n \dots x_1 x_0, x_{-1} \dots x_{-m} \quad x_i \in \{0,1\}$$

- Giá trị thập phân

$$X = x_n 2^n + x_{n-1} 2^{n-1} + \dots + x_1 2^1 + x_0 2^0 + x_{-1} 2^{-1} + \dots + x_{-m} 2^{-m}$$

$$X = \sum_{i=-m}^n x_i 2^i$$

- Ví dụ

$$\begin{aligned} 1101.01_2 &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} \\ &= 13.25_{10} \end{aligned}$$

Chuyển đổi số nguyên thập phân sang nhị phân

- Phương pháp 1

Chia dần cho 2 rồi lấy phần dư

Ví dụ : 37_{10}

$37 : 2 = 18$	dư	1
$18 : 2 = 9$	dư	0
$9 : 2 = 4$	dư	1
$4 : 2 = 2$	dư	0
$2 : 2 = 1$	dư	0
$1 : 2 = 1$	dư	1



Biểu diễn số dư theo
chiều mũi tên

Kết quả: $37_{10} = 100101_2$

Minh họa

Chuyển đổi số nguyên thập phân sang nhị phân

- Phương pháp 2

Phân tích thành tổng của các số 2^i gần nhất

Ví dụ :

$37_{(10)}$

$$\begin{aligned} 37 &= 32 + 4 + 1 \\ &= 2^5 + 2^2 + 2^0 \\ &= 100101 \end{aligned}$$

$101_{(10)}$

$$\begin{aligned} 101 &= 64 + 32 + 4 + 1 \\ &= 2^6 + 2^5 + 2^2 + 2^0 \\ &= 1100101 \end{aligned}$$

$1132_{(10)}$

$$\begin{aligned} 1132 &= 1024 + 64 + 32 + 8 + 4 \\ &= 2^{10} + 2^6 + 2^5 + 2^3 + 2^2 \\ &= 10001101100 \end{aligned}$$

Chuyển đổi số lẻ thập phân sang nhị phân

• Phương pháp

Nhân phần lẻ ở hệ cơ số 10 với 2

- Phần nguyên đưa vào hệ nhị phân
- Phần dư nhân tiếp với 2, làm như vậy cho tới khi bằng 0 thì dừng
- Có trường hợp kết quả không thể bằng 0 thì ta dừng ở độ chính xác nào đó.

• Ý tưởng PP

$$0.b_{-1}b_{-2}b_{-3}\dots \quad b_i = 0 \text{ or } 1$$

$$(b_{-1} \times 2^{-1}) + (b_{-2} \times 2^{-2}) + (b_{-3} \times 2^{-3}) \dots$$

$$F = 2^{-1} \times (b_{-1} + 2^{-1} \times (b_{-2} + 2^{-1} \times (b_{-3} + \dots$$

$$2 \times F = b_{-1} + 2^{-1} \times (b_{-2} + 2^{-1} \times (b_{-3} + \dots$$

Chuyển đổi số lẻ thập phân sang nhị phân

Ví dụ 1 : 0.6875₍₁₀₎

$0.6875 * 2 = 1.375$	phần nguyên =	1
$0.375 * 2 = 0.75$		0
$0.75 * 2 = 1.5$		1
$0.5 * 2 = 1.0$		1



Biểu diễn
theo chiều
mũi tên

Kết quả: $0.6875_{(10)} = 0.1011_{(2)}$

Ví dụ 2 : 0.76₍₁₀₎

$0.76 * 2 = 1.52$	phần nguyên =	1
$0.52 * 2 = 1.04$		1
$0.04 * 2 = 0.08$		0
$0.08 * 2 = 0.16$		0
$0.16 * 2 = 0.32$		0
$0.32 * 2 = 0.64$		0
$0.64 * 2 = 1.28$		1

Kết quả : $0.76_{(10)} = 0.110001_{(2)}$

Hệ mười sáu (HEXA)



- Cơ số 16
- Biểu diễn: 0...9, A, B, C, D, E, F
- Dùng n chữ số hệ 16, biểu diễn 16^n giá trị

$$\underbrace{00\dots00}_n = 0$$

$$\underbrace{FF\dots FF}_n = 16^n - 1$$

- VD: 1 chữ số: 0,1,...,F 16^1
- 2 chữ số: 00,01,...,FF 16^2
- Viết gọn cho số nhị phân: cứ nhóm 4 bit lại rồi đổi sang hệ 16
- VD $A = 0111\ 1011_2$
 $= 7B_h$

Hệ mười sáu



Decimal (base 10)	Binary (base 2)	Hexadecimal (base 16)
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	0001 0000	10
17	0001 0001	11
18	0001 0010	12

$B = 0010\ 1101\ 1001\ 1011_2$

$= 2D9B_{16}$

$C = 1111\ 1010\ 1011\ 0100_2$

$= FAB4_{16}$

Chuyển đổi số nguyên thập phân sang hệ 16



- **PP1:** Chuyển sang số nhị phân, nhóm 4 bit lại rồi đổi sang hệ 16
- **PP2:** Lấy số cơ số 10 chia cho 16
 - Số dư đưa vào kết quả
 - Số nguyên đem chia tiếp cho 16
 - Quá trình lặp lại cho đến khi số nguyên bằng 0
- **VD: 123_{10}**

$$123 : 16 = 7 \quad \text{dư} \quad 11 \text{ (B)}$$

$$7 : 16 = 0 \quad \text{dư} \quad 7$$

$$\text{Kết quả: } 123_{10} = 7B_h$$



Biểu diễn theo chiều
mũi tên

2763_{10}

$$2763 : 16 = 172 \quad \text{dư} \quad 11 \text{ (B)}$$

$$172 : 16 = 10 \quad \text{dư} \quad 12 \text{ (C)}$$

$$12 : 16 = 0 \quad \text{dư} \quad 10 \text{ (A)}$$

2.1. Đại số Boole



- Đại số Boole sử dụng các biến Logic và phép toán Logic
- Giá trị biến Logic :0(True, Thấp, Mở, Tối)
1(False, Cao, Đóng, Sáng)
- Năm 1938 Claude Shannon: Dùng các quy tắc logic do George Boole đưa ra để thiết kế mạch điện
- Mạch điện hoạt động được xác định bởi một hàm Boole chỉ rõ giá Input và Output

Các phép toán cơ bản



■ Các phép toán Logic cơ bản

AND, OR và NOT

- A AND B $A \cdot B$ hay AB
- A OR B $A + B$
- NOT A \bar{A} hay A'
- Thứ tự ưu tiên: NOT > AND > OR

■ Ngoài ra

NAND, NOR, XOR

- A NAND B: $\overline{A \cdot B}$
- A NOR B: $\overline{A + B}$
- A XOR B: $A \oplus B = A \cdot \bar{B} + \bar{A} \cdot B$

Các phép toán cơ bản



P	Q	NOT P (\bar{P})	P AND Q ($P \cdot Q$)	P OR Q ($P + Q$)	P NAND Q ($\overline{P \cdot Q}$)	P NOR Q ($\overline{P + Q}$)	P XOR Q ($P \oplus Q$)
0	0	1	0	0	1	1	0
0	1	1	0	1	1	0	1
1	0	0	0	1	1	0	1
1	1	0	1	1	0	0	0

Operation	Expression	Output = 1 if
AND	$A \cdot B \cdot \dots$	All of the set $\{A, B, \dots\}$ are 1.
OR	$A + B + \dots$	Any of the set $\{A, B, \dots\}$ are 1.
NAND	$\overline{A \cdot B \cdot \dots}$	Any of the set $\{A, B, \dots\}$ are 0.
NOR	$\overline{A + B + \dots}$	All of the set $\{A, B, \dots\}$ are 0.
XOR	$A \oplus B \oplus \dots$	The set $\{A, B, \dots\}$ contains an odd number of ones.

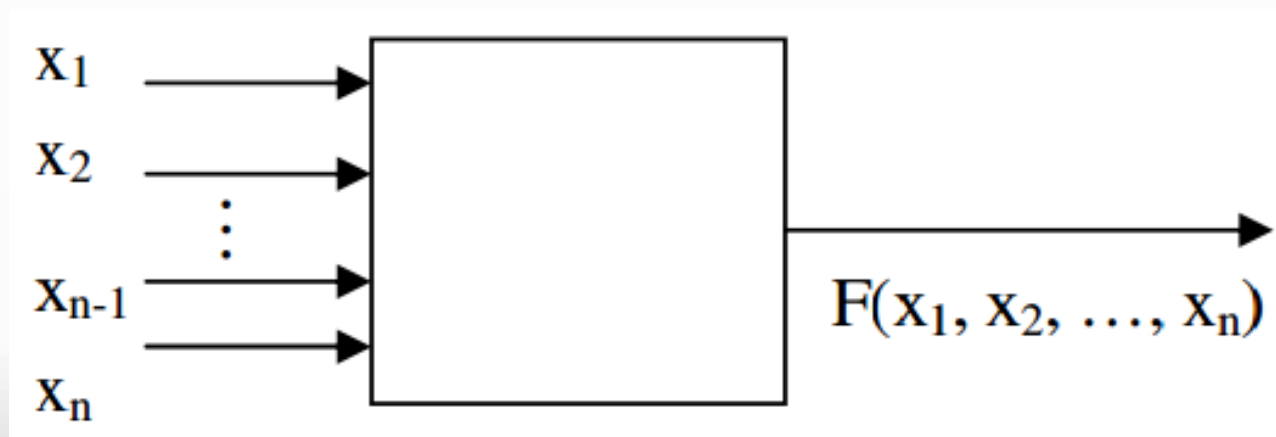
Các đồng nhất thức



Basic Postulates		
$A \cdot B = B \cdot A$	$A + B = B + A$	Commutative Laws
$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$	$A + (B \cdot C) = (A + B) \cdot (A + C)$	Distributive Laws
$1 \cdot A = A$	$0 + A = A$	Identity Elements
$A \cdot \bar{A} = 0$	$A + \bar{A} = 1$	Inverse Elements
Other Identities		
$0 \cdot A = 0$	$1 + A = 1$	Associative Laws
$A \cdot A = A$	$A + A = A$	
$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	$A + (B + C) = (A + B) + C$	DeMorgan's Theorem
$\overline{A \cdot B} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A} \cdot \bar{B}$	


2.3. Cổng LOGIC

- Thực hiện các hàm logic
- Một hoặc một vài đầu vào (Input), duy nhất 1 đầu ra Output
- Một đầu vào: NOT
- Hai đầu vào: AND, OR, NAND, NOR, XOR
- Nhiều đầu vào: Kết hợp các hàm logic cơ bản trên




2.3. Cổng AND



Name	Graphical Symbol	Algebraic Function	Truth Table															
AND		$F = A \bullet B$ or $F = AB$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	0	1	0	0	1	1	1
A	B	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																

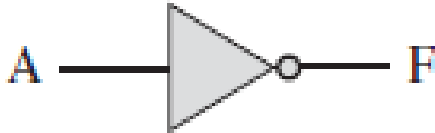
2.3. Cổng OR



Name	Graphical Symbol	Algebraic Function	Truth Table															
OR		$F = A + B$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	1
A	B	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																


2.3. Cổng NOT



Name	Graphical Symbol	Algebraic Function	Truth Table						
NOT		$F = \overline{A}$ or $F = A'$	<table><tr><th>A</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	F	0	1	1	0
A	F								
0	1								
1	0								

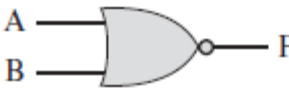
2.3. Cổng NAND



Name	Graphical Symbol	Algebraic Function	Truth Table															
NAND		$F = \overline{AB}$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	1	0	1	1	1	0	1	1	1	0
A	B	F																
0	0	1																
0	1	1																
1	0	1																
1	1	0																


2.3. Cổng NOR



Name	Graphical Symbol	Algebraic Function	Truth Table															
NOR		$F = \overline{A + B}$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	1	0	1	0	1	0	0	1	1	0
A	B	F																
0	0	1																
0	1	0																
1	0	0																
1	1	0																



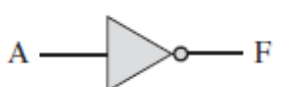



2.3. Cổng XOR



Name	Graphical Symbol	Algebraic Function	Truth Table															
XOR		$F = A \oplus B$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	0
A	B	F																
0	0	0																
0	1	1																
1	0	1																
1	1	0																

Cổng LOGIC



Name	Graphical Symbol	Algebraic Function	Truth Table															
AND		$F = A \cdot B$ or $F = AB$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	0	1	0	0	1	1	1
A	B	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$F = A + B$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	1
A	B	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
NOT		$F = \overline{A}$ or $F = A'$	<table><tr><th>A</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	F	0	1	1	0									
A	F																	
0	1																	
1	0																	
NAND		$F = \overline{AB}$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	1	0	1	1	1	0	1	1	1	0
A	B	F																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR		$F = \overline{A + B}$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	1	0	1	0	1	0	0	1	1	0
A	B	F																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
XOR		$F = A \oplus B$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	0
A	B	F																
0	0	0																
0	1	1																
1	0	1																
1	1	0																

Tập đầy đủ



- **Tập đầy đủ** (complete): Là tập các cổng có thể thực hiện được bất kỳ hàm logic nào từ các cổng của tập đó. (NAND, NOR)
- Ví dụ:
 - {AND, OR, NOT}
 - {AND, NOT} $\overline{A \cdot B} = \overline{A} + \overline{B}$
 - {OR, NOT} $\overline{A + B} = \overline{A} \cdot \overline{B}$
 - {NAND}
 - {NOR}

Sử dụng cổng NAND

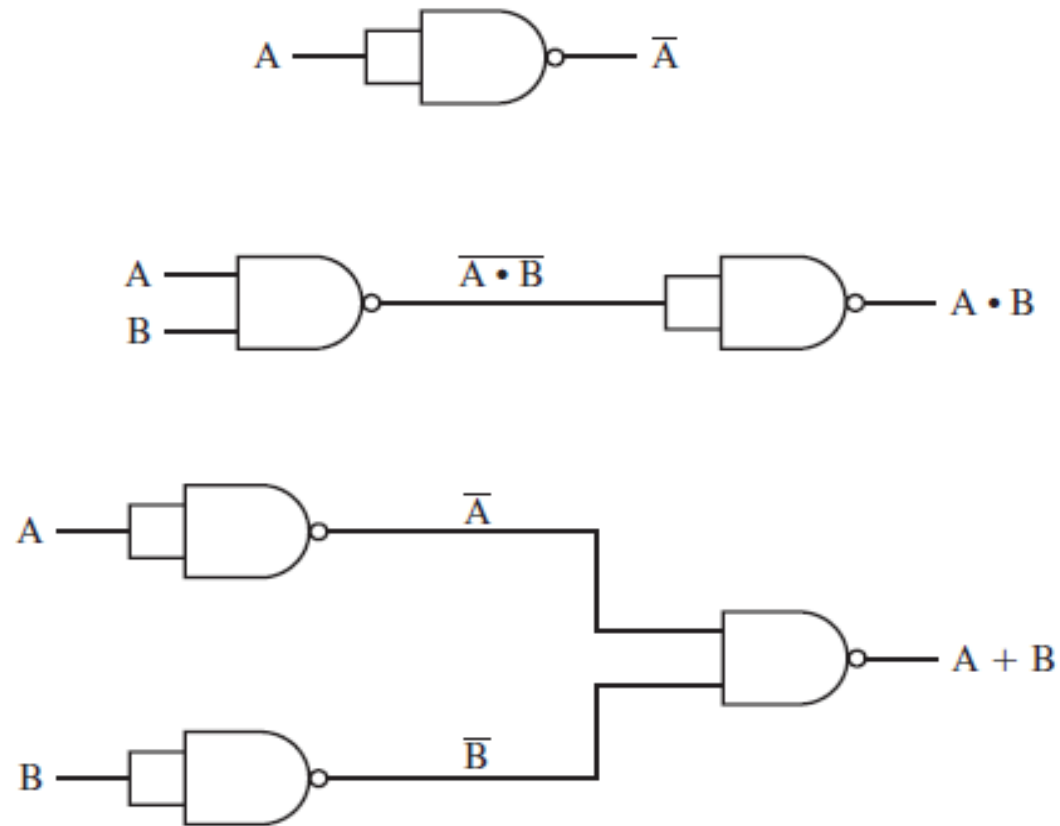


Figure 20.2 The Use of NAND Gates

Sử dụng cổng NOR

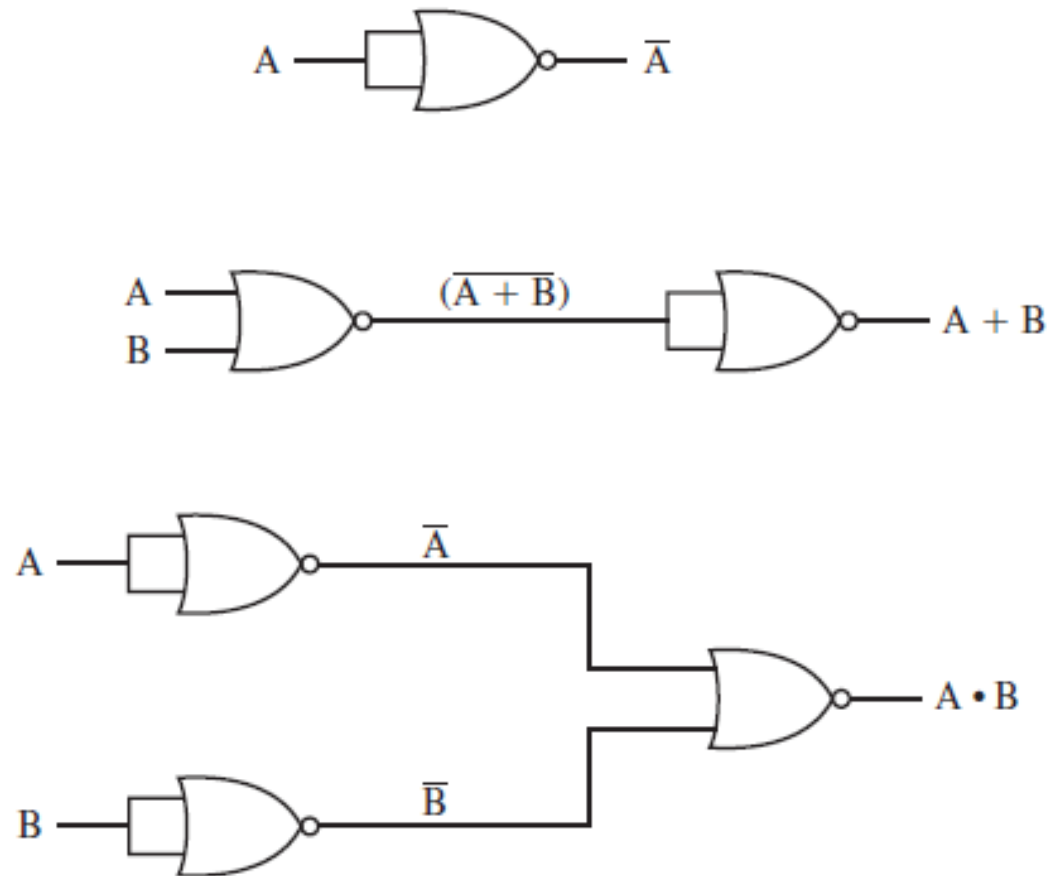


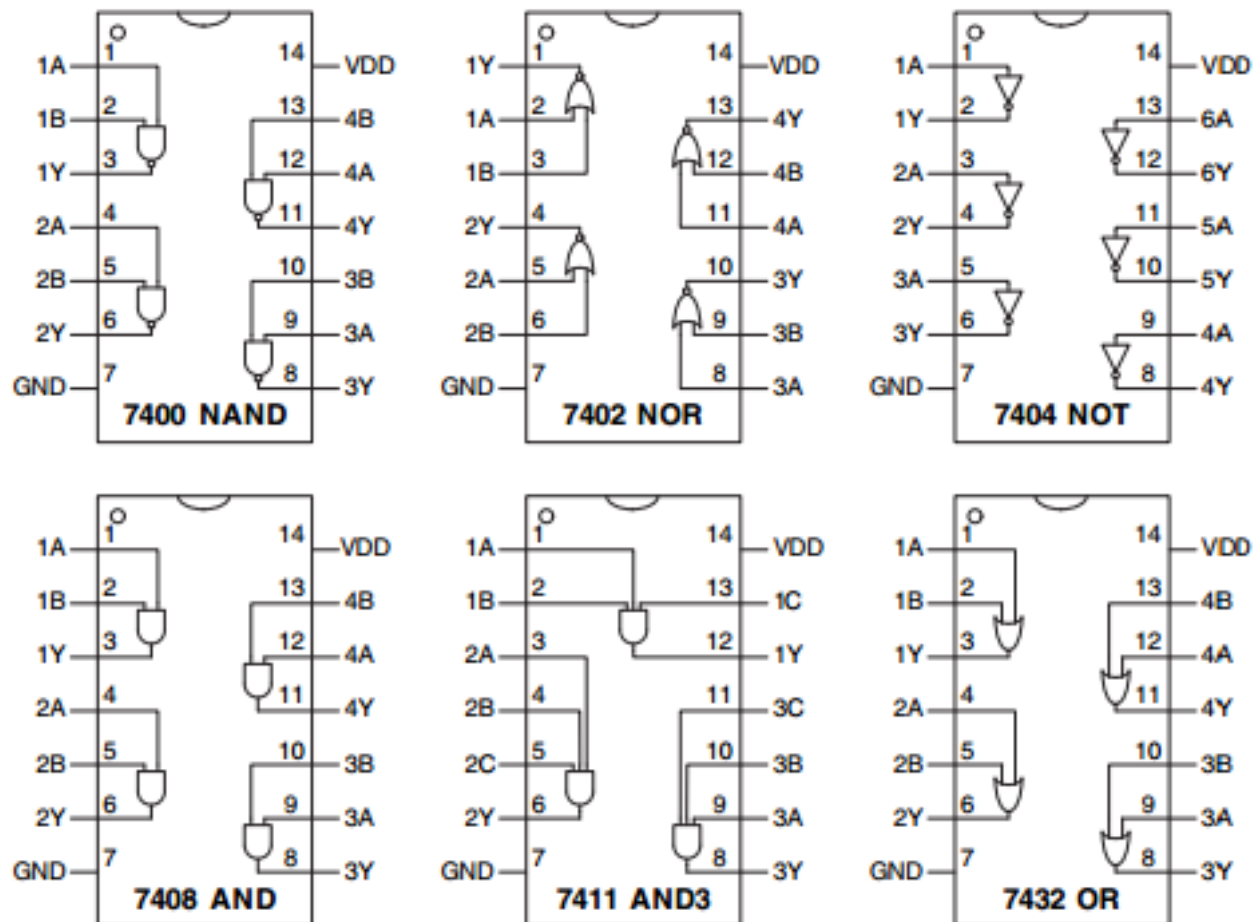
Figure 20.3 The Use of NOR Gates

2.4. Mạch tổ hợp



- Mạch logic bao gồm:
 - Các đầu vào (Inputs)
 - Đầu ra (Outputs)
 - Đặc tả chức năng hoạt động (Functional specification)
 - Đặc tả thời gian (Timing specification)
- Các kiểu mạch logic:
 - Mạch tổ hợp (Combinational Circuits)
 - Mạch dãy (Sequentail Circuits)

Ví dụ mạch LOGIC



Mạch tổ hợp

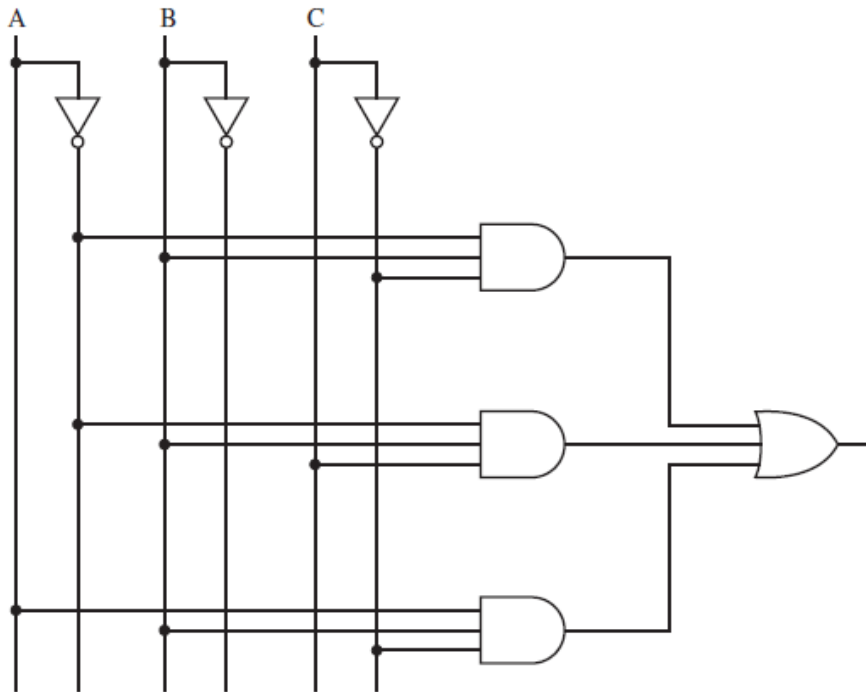


- Gồm một tập các cổng logic kết nối với nhau
- Tại một thời điểm đầu ra là hàm nhị phân của các đầu vào
- 3 cách xác định:
 - Bảng chân lý (Truth Table)
 - Sơ đồ
 - Phương trình Boole

Ví dụ mạch tổ hợp



$$F = \bar{A}B\bar{C} + \bar{A}BC + ABC\bar{C}$$



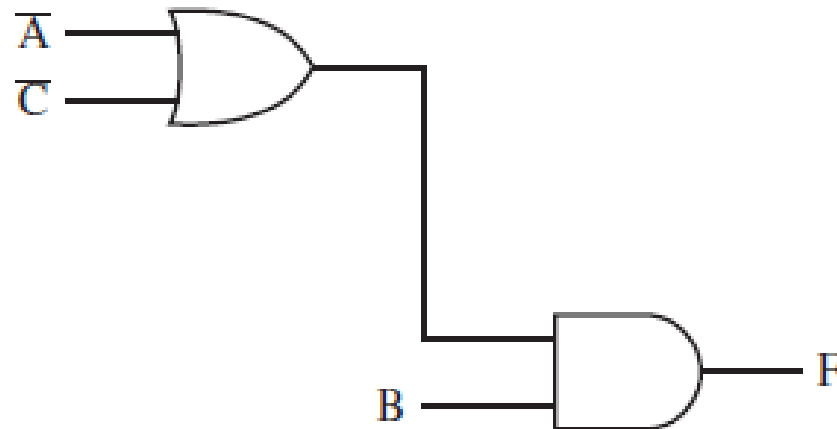
A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Ví dụ đơn giản mạch

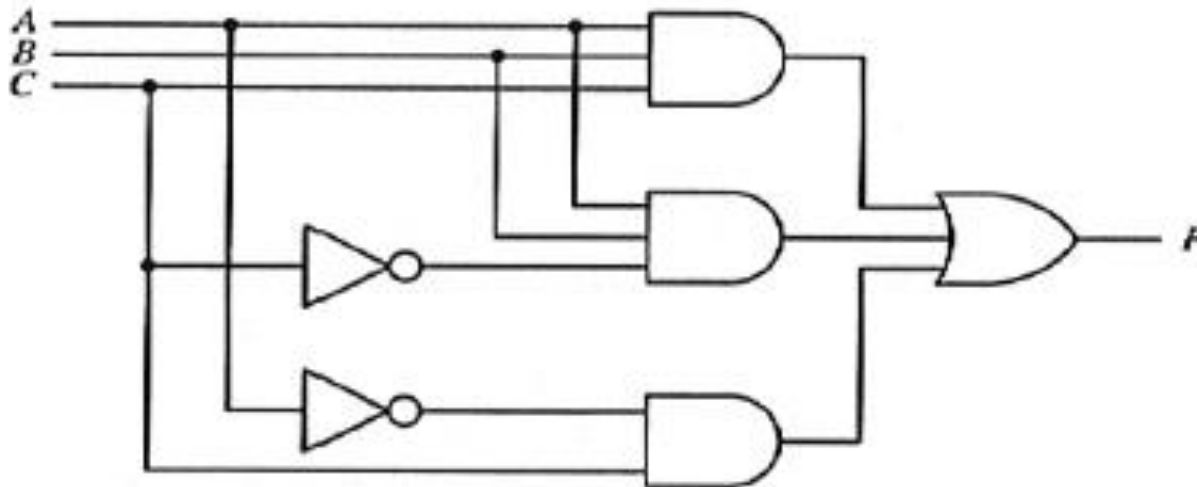


$$F = \overline{A}B + B\overline{C}$$

$$F = B(\overline{A} + \overline{C})$$

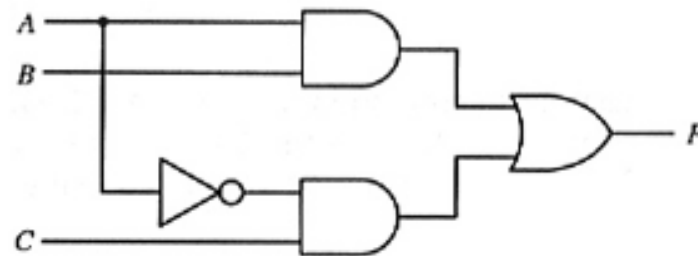


Ví dụ đơn giản mạch



(a) $F = ABC + ABC' + A'C$

- $F = AB(C + C') + A'C$
- $F = AB + A'C$



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

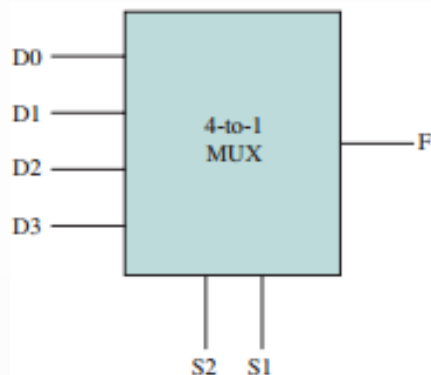
MULTIPLEXER - MUX



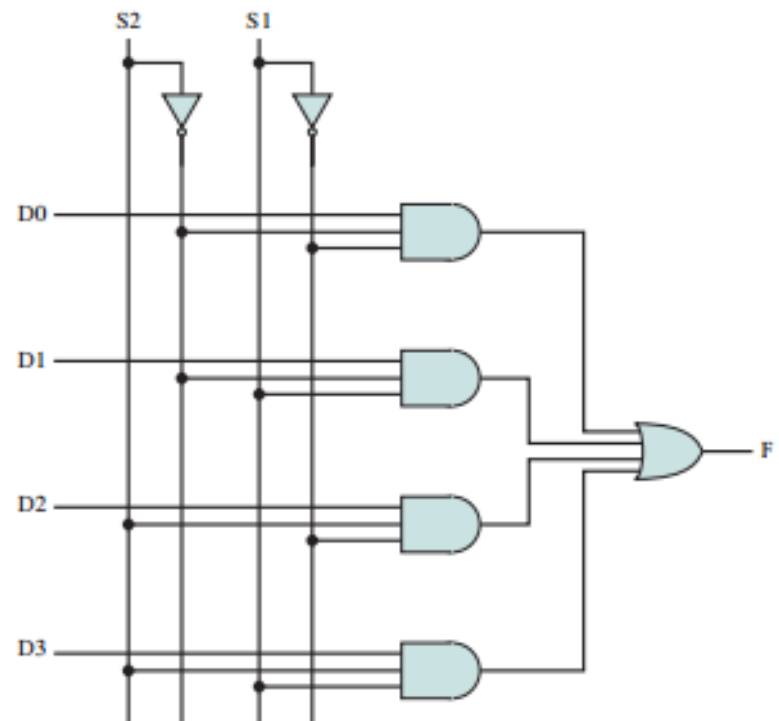
- Bộ chọn kênh
 - 2^n dữ liệu đầu vào
 - n đầu vào chọn
 - 1 đầu ra dữ liệu
 - Chức năng: Chọn 1 tín hiệu trong nhiều tín hiệu đầu vào để đưa ra đầu ra

Ví dụ bộ chọn 4 kênh

$$F = D_0 \overline{S_2} \overline{S_1} + D_1 \overline{S_2} S_1 + D_2 S_2 \overline{S_1} + D_3 S_2 S_1$$

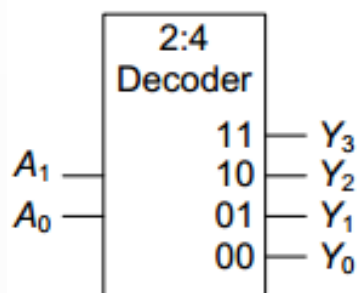


Đầu vào chọn		Đầu ra
S2	S1	F
0	0	D0
0	1	D1
1	0	D2
1	1	D3

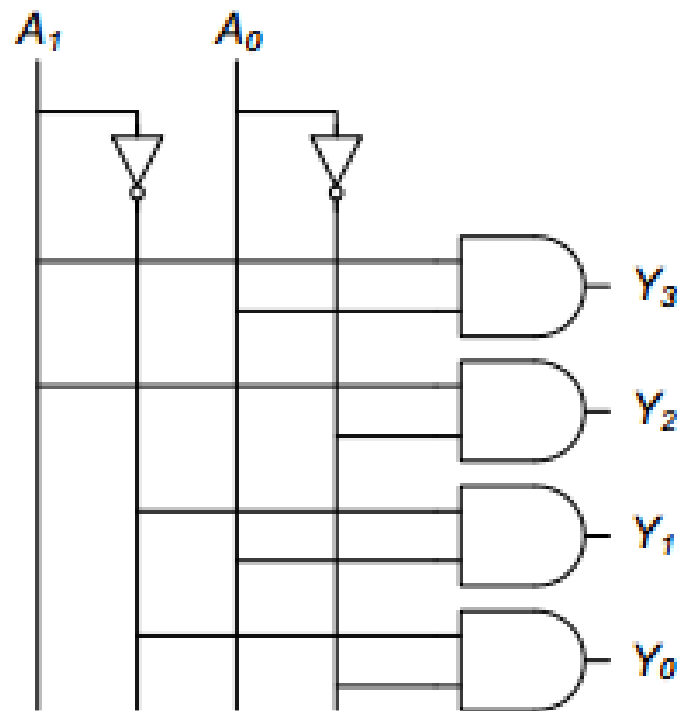


DECODER – Bộ giải mã

- Bộ giải mã:
 - n đầu vào , tối đa 2^n đầu ra
 - Mạch có n đầu vào, m đầu ra gọi là mạch giải mã n x m hoặc n-m
 - Chỉ có 1 đầu ra mang giá trị 1 trong số 2^n đầu ra

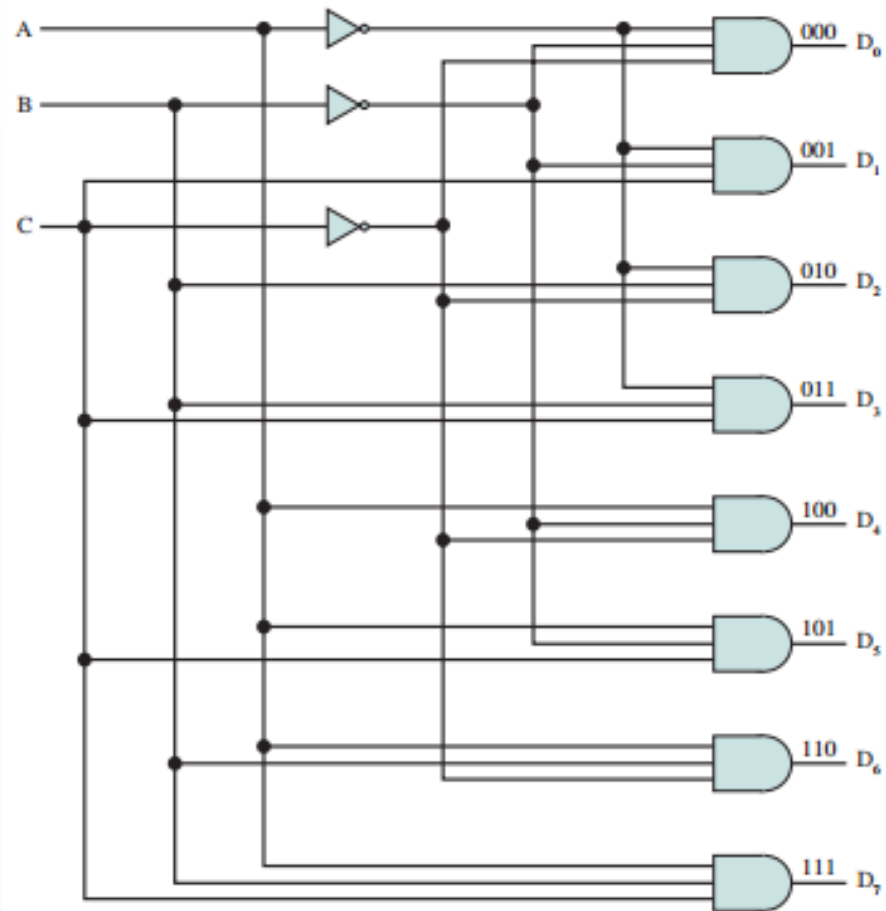


A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0



DECODER – Bộ giải mã

- Bộ giải mã 3 ra 8



ADDER – Bộ cộng



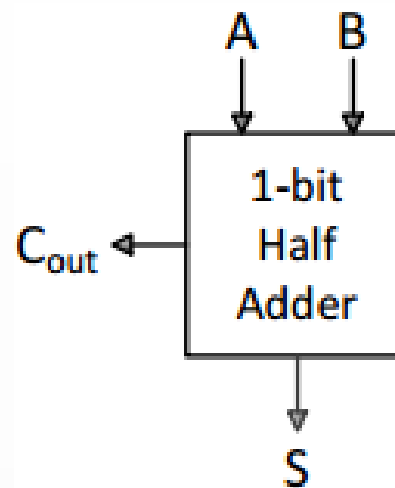
- Bộ cộng bán phần 1-bit (Half-adder)
 - Cộng hai bit tạo ra bit tổng và bit nhớ ra
- Bộ cộng toàn phần 1-bit (Full-adder):
 - Cộng 3 bit
 - Cho phép xây dựng N-bit

Bộ cộng bán phần



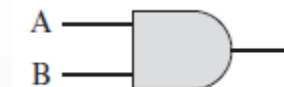
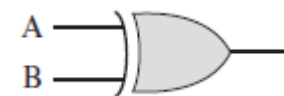
0	0	1	1
<u>+ 0</u>	<u>+ 1</u>	<u>+ 0</u>	<u>+ 1</u>
0	1	1	10

Đầu vào		Đầu ra	
A	B	S	C _{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



$$S = A \oplus B$$

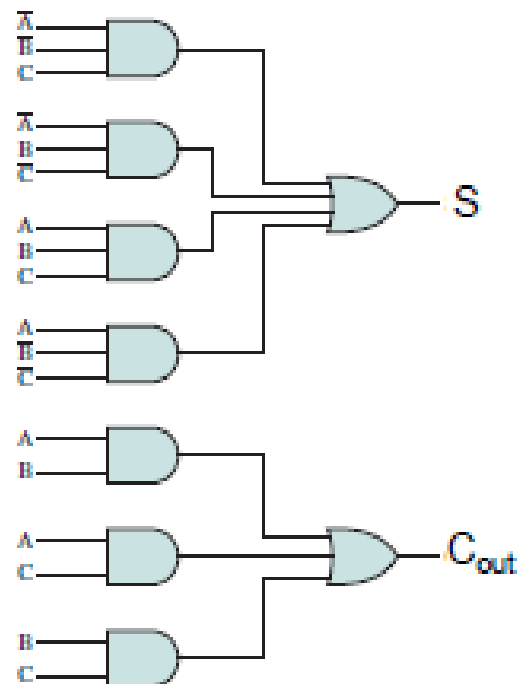
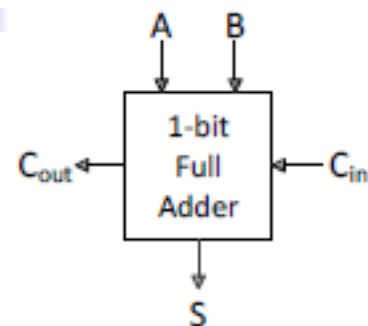
$$C_{out} = AB$$



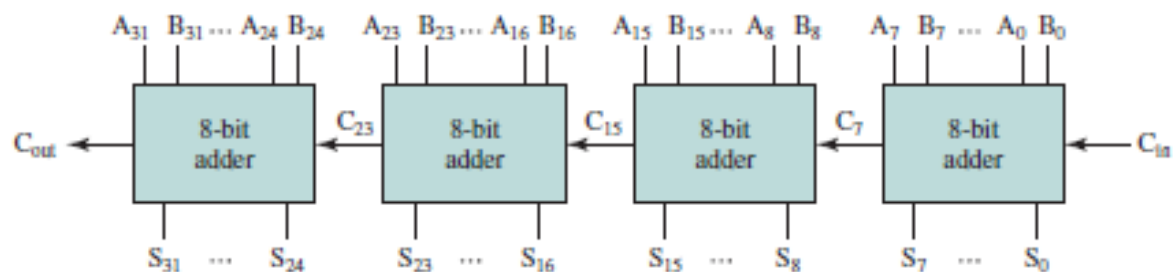
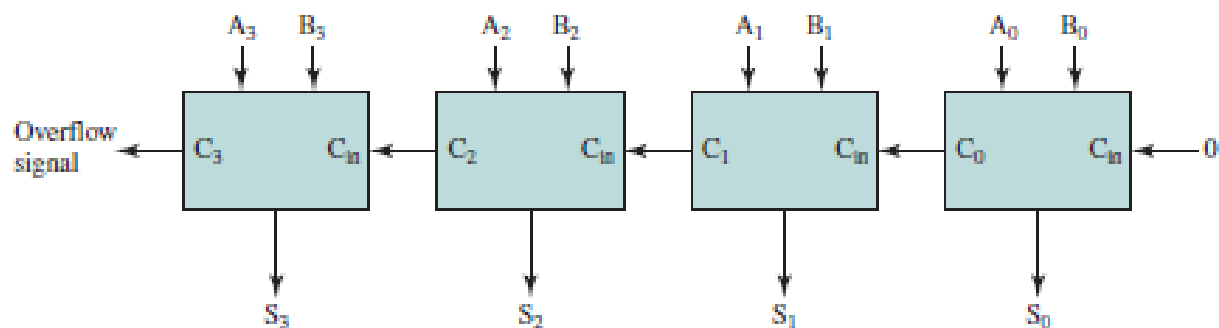
Bộ cộng toàn phần 1 bit

Đầu vào			Đầu ra	
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

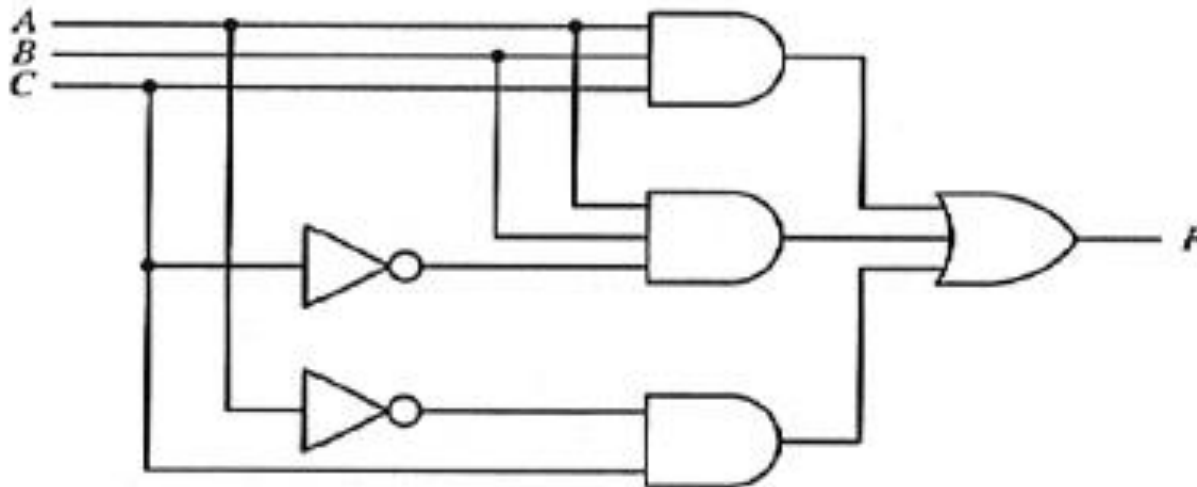
- $S = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$
- $C_{out} = AB + BC + CA$



Bộ cộng 4 bit & 32 bit

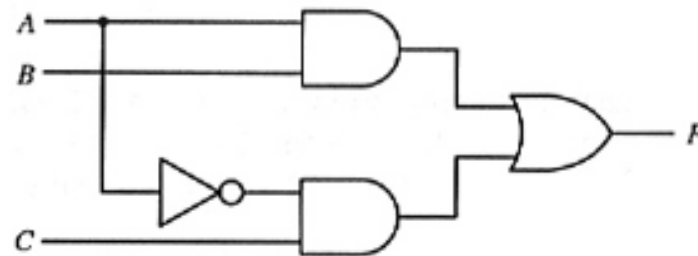


Đơn giản mạch



(a) $F = ABC + ABC' + A'C$

- $F = AB(C + C') + A'C$
- $F = AB + A'C$

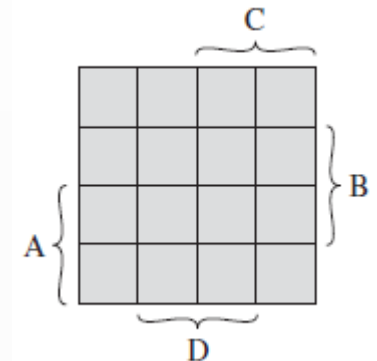
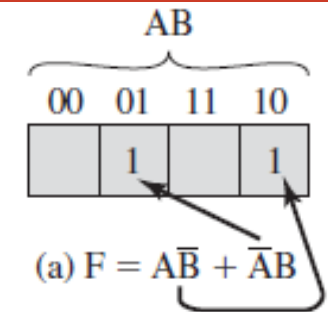
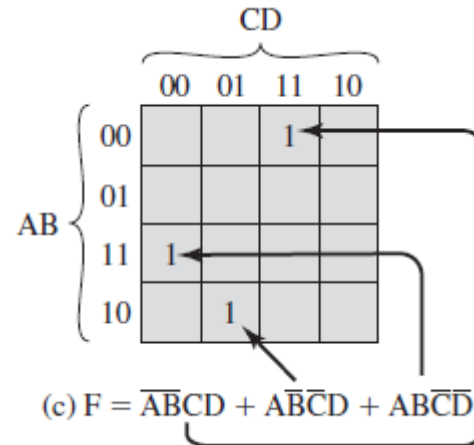
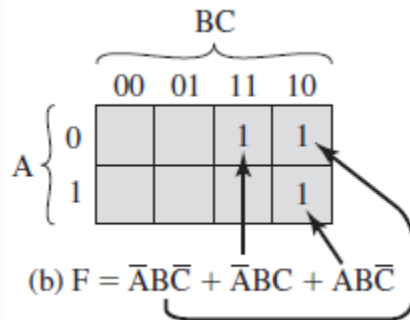


(B) $F = AB + A'C$

Bản đồ Karnaugh



- Dùng hình vẽ gọi là bản đồ Karnaugh hay bản đồ K
- Xuất phát từ bảng sự thật
 - Mỗi tổ hợp biến trong bảng sự thật là một bộ trị
 - Mỗi bộ trị tương ứng là một hình vuông trong bản đồ K
 - N đầu vào, có 2^n hình vuông
 - Giá trị 1 hay 0 được đặt vào hình vuông tương ứng



Bản đồ Karnaugh



■ Quy tắc đặt

- Các ô liền kề chỉ có 1 bit khác nhau
- Ô đầu dòng và cuối dòng là liền kề
- Ô đầu cột và cuối cột là liền kề

A \ B	B	
	0	1
0	0	1
1	2	3

(a) Two-variable map

A \ BC	B			
	00	01	11	10
0	0	1	3	2
1	4	5	7	6

(b) Three-variable map

AB \ CD	C			
	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

(c) Four-variable map

Bản đồ Karnaugh



■ Qui tắc đơn giản hàm

- Đặt giá trị 1 vào ô vuông tương ứng
- Gom các nhóm ô lớn nhất có thể có trị 1 liên nhau, sao cho số ô trong mỗi nhóm là 1,2,4,8..
- Các nhóm phải có ít nhất 1 ô 1 không nằm trong nhóm khác
- Chọn ra số nhóm ít nhất mà chứa hết các ô 1
- Mỗi nhóm là một toán hạng AND
- Trong mỗi toán hạng AND loại biến có giá trị liên tiếp bù nhau

		CD			
		00	01	11	10
AB	00				
	01		1	1	
	11				
	10				

(a) $\bar{A}BD$

		CD			
		00	01	11	10
AB	00		1		
	01				
	11				
	10		1		

(b) $\bar{B}\bar{C}D$

Bản đồ Karnaugh



		CD			
		00	01	11	10
AB	00	1	1	1	1
	01				
	11				
	10				

$\overline{A}\overline{B}$

		CD			
		00	01	11	10
AB	00				
	01	1	1		
	11	1	1		
	10				

$B\overline{C}$

		CD			
		00	01	11	10
AB	00				
	01	1			1
	11	1			1
	10				

$B\overline{D}$

		CD			
		00	01	11	10
AB	00	1	1	1	1
	01	1	1	1	1
	11				
	10				

\overline{A}

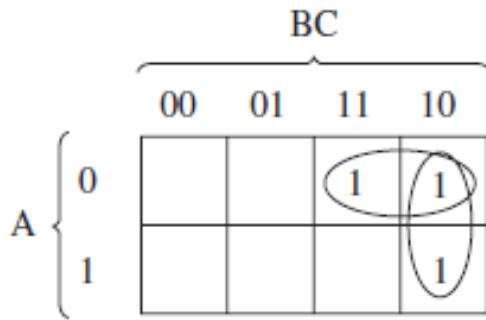
		CD			
		00	01	11	10
AB	00	1			1
	01	1			1
	11	1			1
	10	1			1

\overline{D}

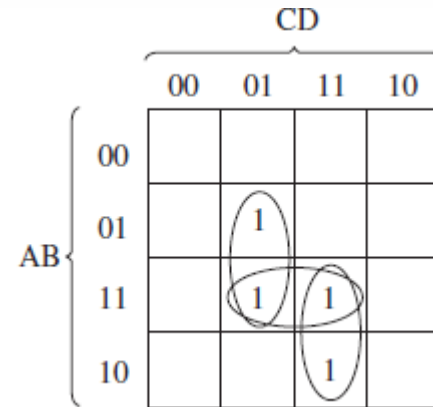
		CD			
		00	01	11	10
AB	00			1	1
	01			1	1
	11			1	1
	10			1	1

C

Bản đồ Karnaugh



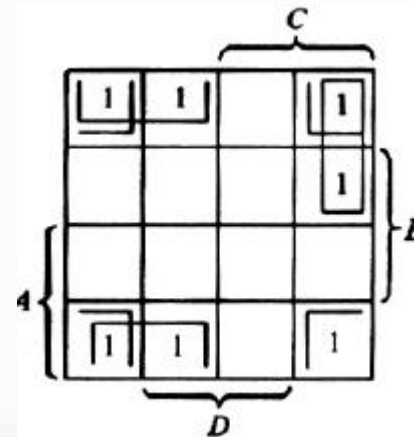
$$F = \bar{A}B + B\bar{C}$$



$$F = \bar{B}\bar{C}D + ACD$$

■ Ví dụ

$$F(A, B, C, D) = \sum (0, 1, 2, 6, 8, 9, 10)$$



$$F = B'C' + B'D' + A'CD'$$

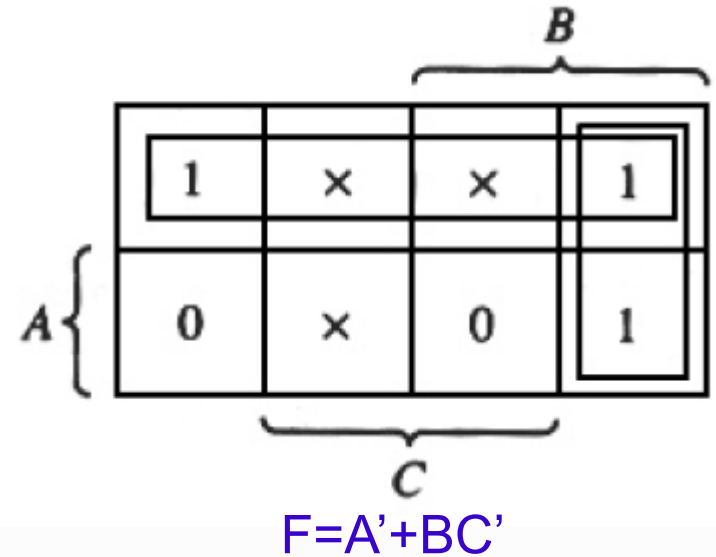
Bản đồ Karnaugh



- Bản đồ với bộ trị tùy chọn

$$F(A, B, C) = \sum (0, 2, 6)$$

$$d(A, B, C) = \sum (1, 3, 5)$$



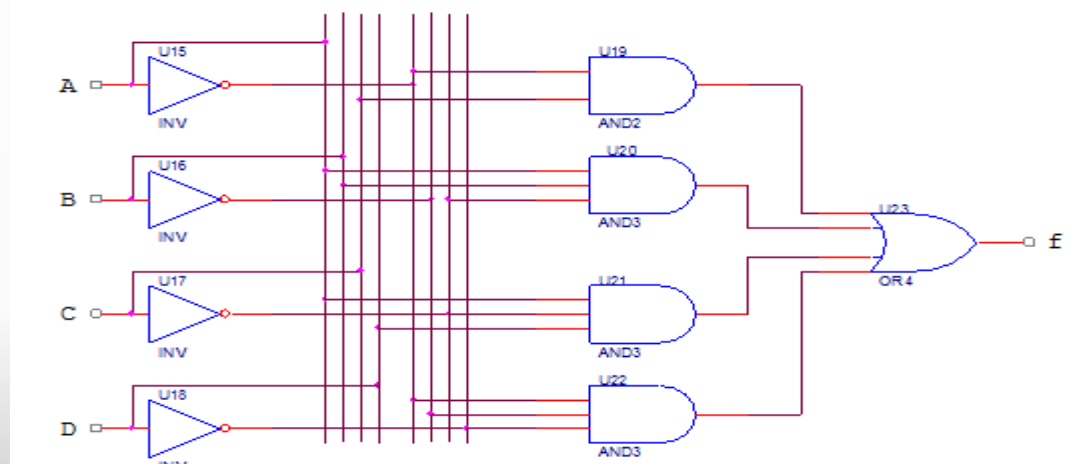
Bản đồ Karnaugh

- Bài tập: Cho sơ đồ K, xác định hàm và vẽ sơ đồ mạch

CD \ AB	00 01 11 10			
	00	01	11	10
00	1		1	1
01			1	1
11	1	1		
10		1		

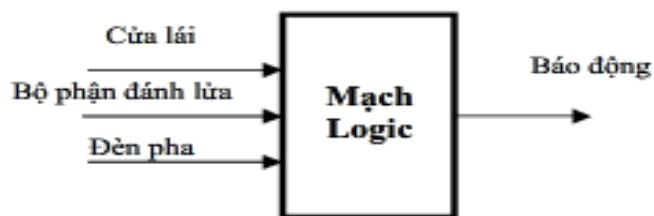
CD \ AB	00	01	11	10
	00	01	11	10
00	1		1	1
01			1	1
11	1	1		
10		1		

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



Bài tập

Để làm một bộ báo hiệu cho lái xe biết một số điều kiện, người ta thiết kế 1 mạch báo động như sau:



Tín hiệu từ :
Cửa lái: 1- cửa mở,
0 – cửa đóng;
Bộ phận đánh lửa:
1 – bật, 0 – tắt;
Đèn pha: 1 – bật, 0
– tắt.

Hãy thiết kế mạch logic với 3 đầu vào (cửa, bộ phận đánh lửa, đèn pha), 1 đầu ra (báo động), sao cho bộ phận báo động sẽ hoạt động (báo động = 1) khi tồn tại một trong 2 trạng thái sau:

- Đèn pha sáng trong lúc bộ phận đánh lửa tắt
- Cửa mở trong lúc bộ phận đánh lửa hoạt động

Lập bảng chân trị của hàm ra.

Đặt các ký hiệu tương ứng:

Cửa lái - A;

Bộ phận đánh lửa - B

Đèn pha - C

Báo động - f

$$f = C\bar{B} + AB$$

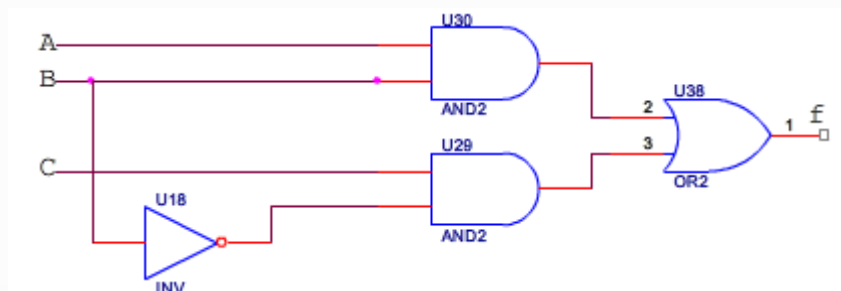
Bài tập



Bảng chân trị

A	B	C	AB	\bar{B}	$C\bar{B}$	f
0	0	0	0	1	0	0
0	0	1	0	1	1	1
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	0	1	0	0
1	0	1	0	1	1	1
1	1	0	1	0	0	1
1	1	1	1	0	0	1

$$f = C\bar{B} + AB$$



Bài tập



Construct a truth table for the following Boolean expressions:

- a. $ABC + \overline{A}\overline{B}\overline{C}$
- b. $ABC + \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C$
- c. $A(\overline{B}C + \overline{B}\overline{C})$
- d. $(A + B)(A + C)(\overline{A} + \overline{B})$

Simplify the following expressions according to the commutative law:

- a. $A \cdot \overline{B} + \overline{B} \cdot A + C \cdot D \cdot E + \overline{C} \cdot D \cdot E + E \cdot \overline{C} \cdot D$
- b. $A \cdot B + A \cdot C + B \cdot A$
- c. $(L \cdot M \cdot N)(A \cdot B)(C \cdot D \cdot E)(M \cdot N \cdot L)$
- d. $F \cdot (K + R) + S \cdot V + W \cdot \overline{X} + V \cdot S + \overline{X} \cdot W + (R + K) \cdot F$

Apply DeMorgan's theorem to the following equations:

- a. $F = \overline{V + A + L}$
- b. $F = \overline{A} + \overline{B} + \overline{C} + \overline{D}$

2.5. Mạch tuần tự



- Là mạch Logic trong đó đầu ra phụ thuộc giá trị đầu vào ở thời điểm hiện tại và đầu vào ở thời điểm quá khứ
- Là mạch có nhớ, được thực hiện bằng phần tử nhớ (Latch, Flip-Flop), có thể kết hợp với các cổng logic
- $Q_+ = f(Q, A, B, C \dots)$
- Phần tử cơ bản là Flip-Flop
- Chia làm 2 loại: Đồng bộ và không đồng bộ

FLIP - FLOP

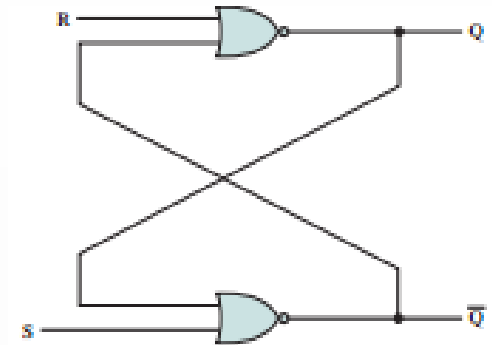


- Là mạch dao động lưỡng ổn
- Thay đổi khi có xung đồng hồ
- Bao gồm:
 - Một hoặc hai đầu vào dữ liệu, và một đầu vào C_k
 - Hai dữ liệu ra là Q(đầu ra chính) và Q_+ (đầu ra phụ)
- Tạo nên từ các mạch chốt (latch)

S-R LATCH



- Chốt RS tác động ở mức cao
 - SR = “set-reset”
 - Bao gồm: 2 dữ liệu đầu vào S và R ở mức cao



S-R Latch

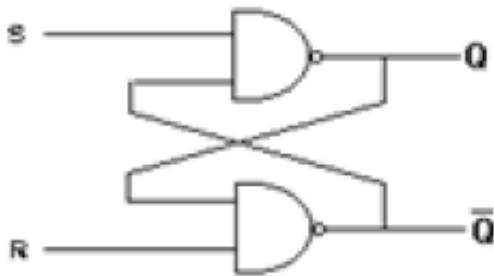
R	S	Q	Q ₊
0	0	0	0) Tác dụng nhớ
0	0	1	1) Q ₊ = Q
0	1	0	1) Đặt (Set)
0	1	1	1) Q ₊ = 1
1	0	0	0) Đặt lại (Reset)
1	0	1	0) Q ₊ = 0
1	1	0) Q ₊ = Q ₊ = 0 (Cấm)
1	1	1)

R	S	Q ₊
0	0	Q
0	1	1
1	0	0
1	1	Cấm

S-R LATCH



- Chốt RS tác động ở mức thấp
 - SR = “set-reset”
 - Bao gồm: 2 dữ liệu đầu vào S và R ở mức cao

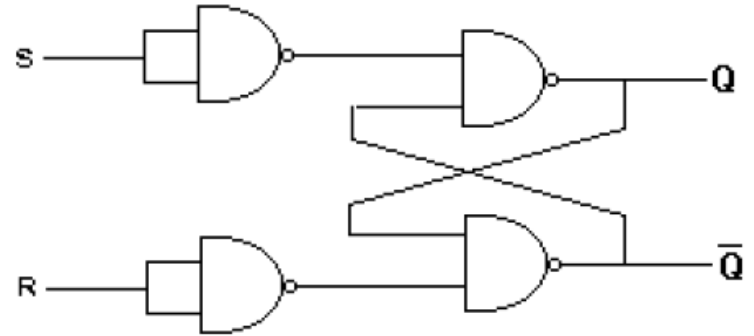
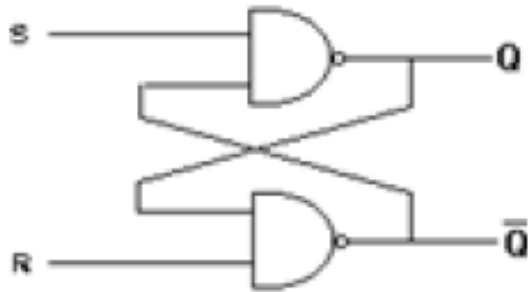


S	R	Q ₊
0	0	Cấm
0	1	1
1	0	0
1	1	Q

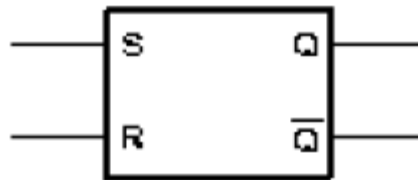
S-R LATCH



- Chuyển RS tác động ở mức thấp \rightarrow cao



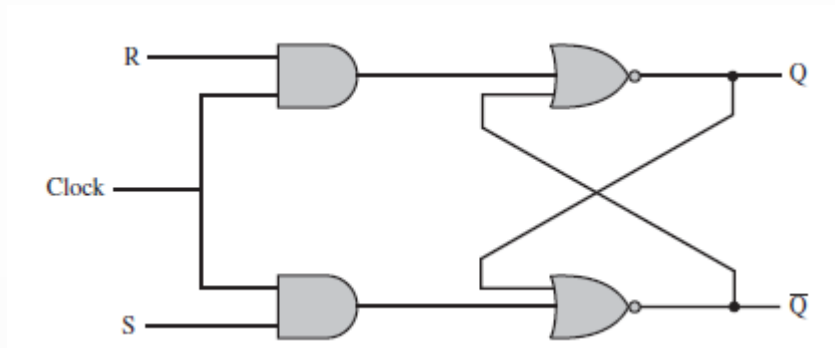
- Ký hiệu RS tác động ở mức cao và RS thấp



S-R FLIP FLOP

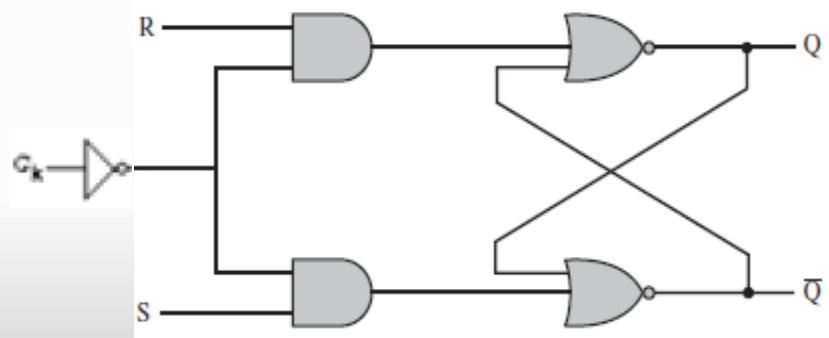
- Sử dụng cổng NOR

Tác động ở mức cao



Vào			Ra
C_K	S	R	Q_+
0	x	x	Q
1	0	0	Q
1	0	1	0
1	1	0	1
1	1	1	Cấm

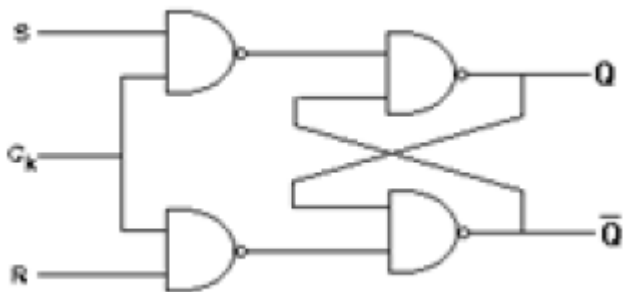
Xung đồng hồ ở mức thấp



S-R FLIP FLOP

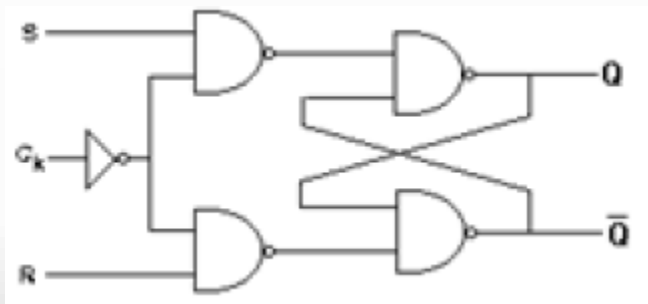
- Sử dụng cổng NAND

Tác động ở mức cao

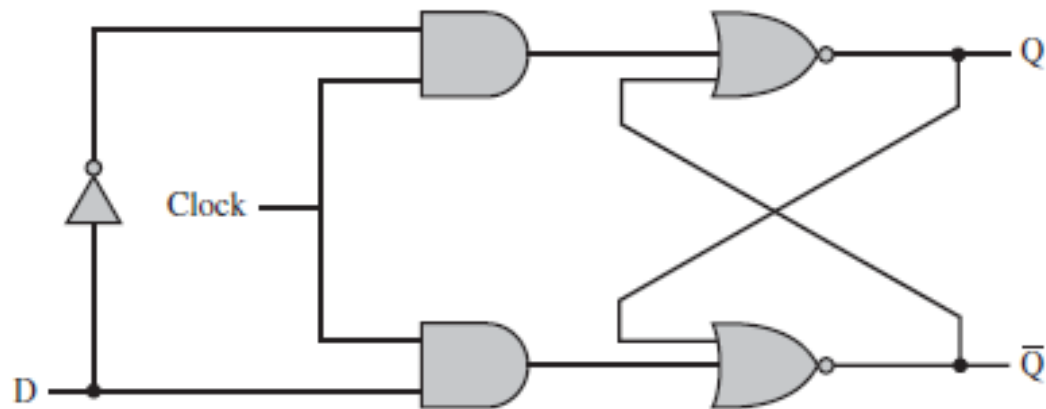


C_K	Vào		Ra
	S	R	Q_+
0	x	x	Q
1	0	0	Q
1	0	1	0
1	1	0	1
1	1	1	Cấm

Xung đồng hồ ở mức thấp

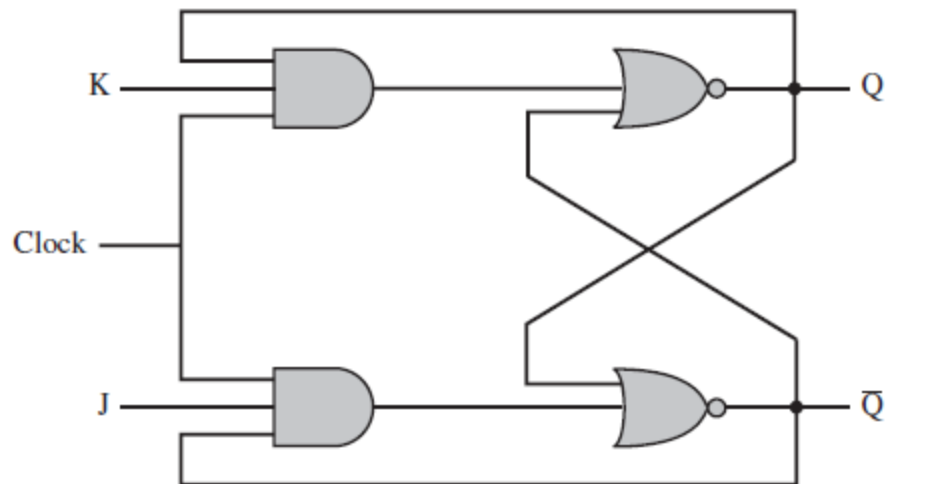


D FLIP FLOP



D	Q_{n+1}
0	0
1	1

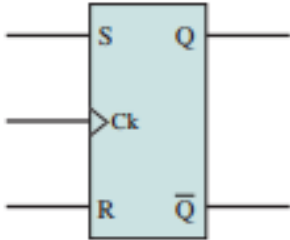
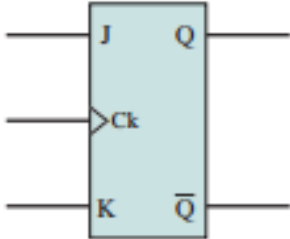
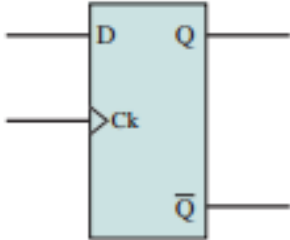
J-K FLIP FLOP



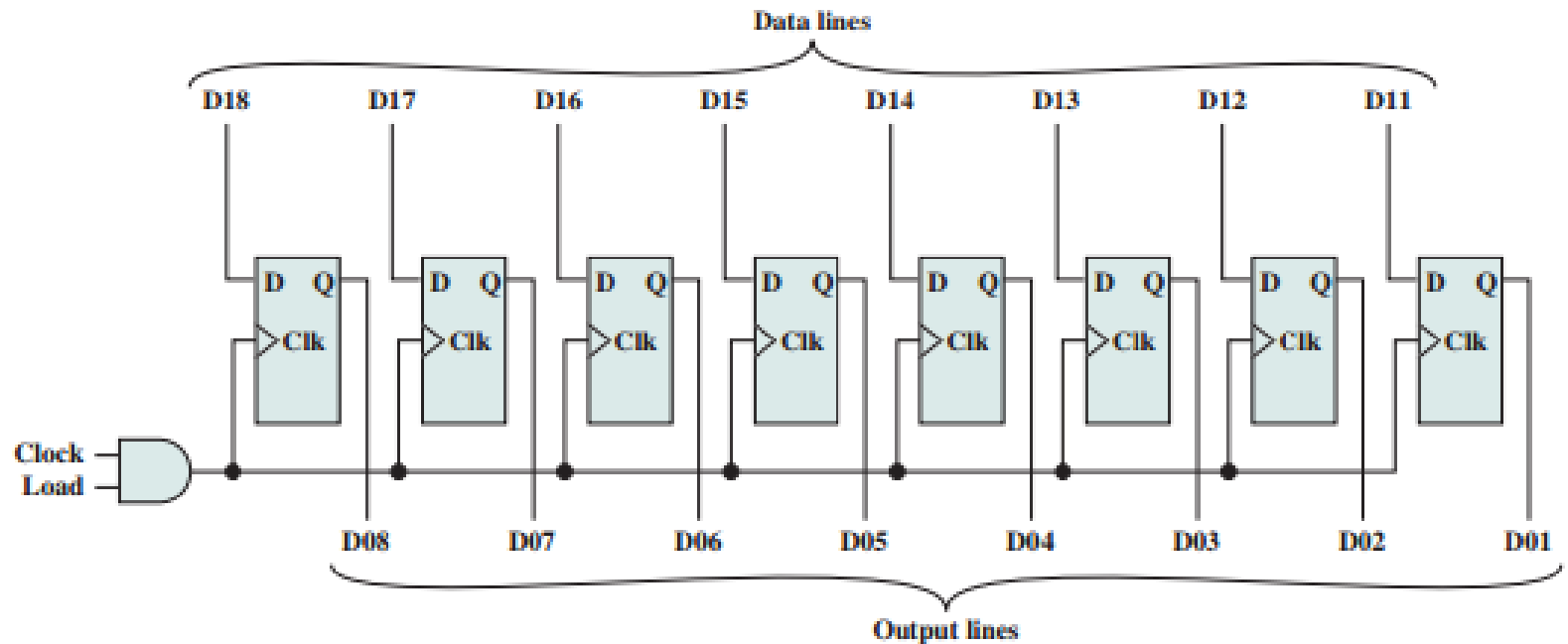
J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

FLIP-FLOP CƠ BẢN

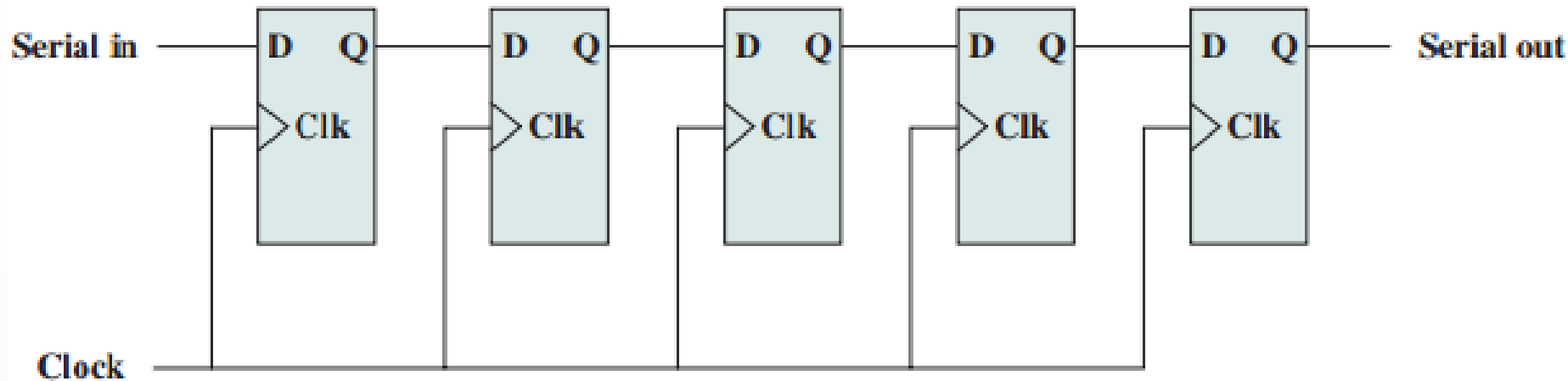


Name	Graphical Symbol	Truth Table															
S-R		<table><tr><th>S</th><th>R</th><th>Q_{n+1}</th></tr><tr><td>0</td><td>0</td><td>Q_n</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>-</td></tr></table>	S	R	Q_{n+1}	0	0	Q_n	0	1	0	1	0	1	1	1	-
S	R	Q_{n+1}															
0	0	Q_n															
0	1	0															
1	0	1															
1	1	-															
J-K		<table><tr><th>J</th><th>K</th><th>Q_{n+1}</th></tr><tr><td>0</td><td>0</td><td>Q_n</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>$\overline{Q_n}$</td></tr></table>	J	K	Q_{n+1}	0	0	Q_n	0	1	0	1	0	1	1	1	$\overline{Q_n}$
J	K	Q_{n+1}															
0	0	Q_n															
0	1	0															
1	0	1															
1	1	$\overline{Q_n}$															
D		<table><tr><th>D</th><th>Q_{n+1}</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	D	Q_{n+1}	0	0	1	1									
D	Q_{n+1}																
0	0																
1	1																

THANH GHI 8 BIT SONG SONG



THANH GHI DỊCH 5-BIT



HẾT CHƯƠNG 2