Chương 1

HỆ THỐNG SỐ VÀ MẠCH LOGIC

- Hệ thống số đếm
- Chuyển đổi cơ số
- Các phép tính trên số nhị phân
- Số có dấu
- Số BCD
- Số có dấu chấm tĩnh và chấm động

Hệ thống số đếm

- Cơ số của một hệ : là số chữ số được dùng để biểu diễn các giá trị
- Các hệ cơ bản
  - Hệ nhị phân (B) hay hệ cơ số 2 : dùng 2 chữ số 0,1 để biểu diễn tất cả các giá trị. Ví dụ : 1001_2 = 1001B
  - Hệ bát phân (O) hay hệ cơ số 8 : dùng 8 chữ số từ 0-7 để biểu diễn tất cả các giá trị. Ví dụ : 762_8 = 762O
  - Hệ thập phân (D) hay hệ cơ số 10 : dùng 10 chữ số từ 0-9 để biểu diễn tất cả các giá trị.
  - Ví dụ : 1981_{10} = 1981D = 1981

  - Hệ thập lục phân (H) hay hệ cơ số 16
    - Dùng 16 chữ - số : 10 số 0,1,2,3,4,5,6,7,8,9
      6 chữ cái A,B,C,D,E,F để biểu diễn tất cả các số
    - Ví dụ : $1B5E_{16}$ = 1B5EH
- Chuyển đổi cơ số

Chuyển đổi cơ số

- Chuyển từ hệ cơ số S bất kỳ sang thập phân
  - Công thức

    $N_S = C_n S^n + C_{n-1} S^{n-1} + C_{n-2} S^{n-2} + \ldots + C_0 S^0 + C_{-1} S^{-1} + \ldots$
  - Hay

    $$N_S = \sum C_i S^i$$
  - Trong đó :

    $$0 \le C_i \le S-1$$

    i là thứ tự vị trí các chữ số, i=0 tương ứng với vị trí đầu tiên đúng trước dấu chấm thập phân.

Các ví dụ

- Từ B – D
  - Ví dụ : 1001_2

    $$1001_2 = 1x2^3 + 0x2^2 + 0x2^1 + 1x2^0 = 9$$
- Từ O – D
  - Ví dụ : 162.43_8

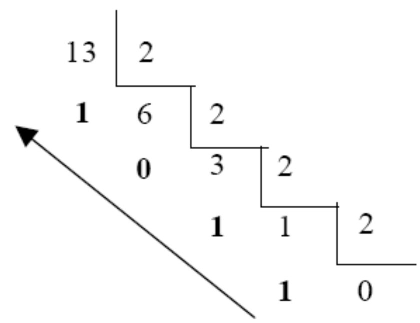
    $162.43_8 = 1x8^2+6x8^1+2x8^0+4x8^{-1}+3x8^{-2}$
- Từ H – D
  - Ví dụ : $1E4A.6B_{16}$

    $1x16^3+Ex16^2+4x16^1+Ax16^0+6x16^{-1}+Bx16^{-2}$

    $1x16^3+14x16^2+4x16^1+10x16^0+6x16^{-1}+11x16^{-2}$

Chuyển từ thập phân sang các hệ khác

- Ví dụ : $13=1101_2$

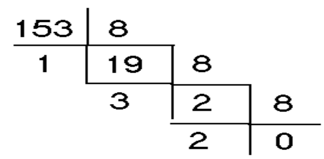


- Ví dụ 2 : 13.125
  - Đổi phần nguyên : $13=1101_2$
  - Đổi phần thập phân : 0.125

	Kết quả	Nguyên	Thập phân
0.125 x 2 = 0.250		0	0.25
0.25 x 2 = 0.50		0	0.5
0.5 x 2 = 1.0		1	0 (kết thúc)

- Kết quả : 001_2
- Kết hợp nguyên – thập phân : 1101.001

- Chuyển từ thập phân sang bát phân (D – O)
- Ví dụ : 153.513

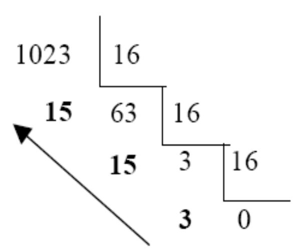


  - **Phần nguyên : 153=231$_8$**
  - Phần thập phân : 0.513

Nhân	Kết quả	Nguyên	Thập phân
0.513x 8	= 4.104	4	0.104
0.104x 8	= 0.832	0	0.832
0.832x 8	= 6.656	6	0.656
0.656x8	= 5.248	5	0.28
0.248x8	= 1.984	1	0.984…

- Kết quả :40651$_8$
- Kết hợp Nguyên – thập phân : 231.40651$_8$

- Chuyển từ thập phân sang thập lục phân (D – H)
- Ví dụ : 1023 = 3FFH



- **Chuyển từ B – O**
- **Chuyển từ B – H**
- **Chuyển từ O – B**
- **Chuyển từ H – B**
- **Cách 1 : Lấy D làm trung gian**
  - **B – D – O**
  - **B – D – H**
  - **O – D – B**
  - **H – D – B**
- **Ví dụ : 10111110111111110 : Khó thực hiện**

- Cách 2 : Dùng 2 nhận xét :
  - Một số trong hệ bát phân tương đương 3 chữ số trong hệ nhị phân
  - Một số trong hệ thập lục phân tương đương 4 chữ số trong hệ nhị phân
  - Quan hệ cụ thể trình bày trong bảng sau :

Bảng 1

STT	H	B
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111

STT	H	B
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Bảng 2

STT	O	B
0	0	000
1	1	001
2	2	010
3	3	011
4	4	100
5	5	101
6	6	110
7	7	111

- Ví dụ : Chuyển từ B sang O và H
- **1011111011111110**

1 3 7 3 7 6

1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0

B E F E

Các Phép Tính Số Học

- Khái niệm về LSB và MSB
- Xét số nhị phân :

7 6 5 4 3 2 1 0

10011110 **$= 1x2^7 + 0x2^6 + 0x2^5 + 1x2^4 + \dots$**

LSB : Least significant bit

MSB : Most significant bit

1Byte = 8 bit
1KB = 2^{10} Byte = 1024 Byte
1MB = 2^{10} KB = 1024 KB.
1GB = 2^{10} MB = 1024 MB

16

Các Phép Tính Số Học

- **Cộng nhị phân : 4 trường hợp có thể xảy ra :**

0 + 0 = 0

1 + 0 = 1

1 + 1 = 10 = 0 + nhớ 1 vào vị trí kế tiếp.

1 + 1 + 1 = 11 = 1 + nhớ 1 vào vị trí kế.

Ví dụ :

```
   1 1. 0 1 1 (3.375)
   1 0. 1 1 0 (2.750)
 1 1 0. 0 0 1 (6.125)
```

17

Các Phép Tính Số Học

- **Trừ nhị phân : 4 trường hợp có thể xảy ra :**

0 – 0 = 0

1 – 0 = 1

1 – 1 = 0

0 – 1 = 1 Nợ 1

Ví dụ :

Số mượn	11	111
Số bị trừ	100	111001
Số trừ	011	1011
Hiệu	001	101110

18

Các Phép Tính Số Học

- **Nhân nhị phân : 4 trường hợp có thể xảy ra :**

  0 x 0 = 0

  0 x 1 = 0

  1 x 0 = 0

  1 x 1 = 1

Ví dụ :

$$\begin{array}{r} 7 \\ \times\ 5 \\ \hline 35 \end{array} \qquad \begin{array}{c} \rightarrow \\ \rightarrow \end{array} \qquad \begin{array}{r} 0111 \\ \times\ 0101 \\ \hline 0111 \\ 0000 \\ 0111 \\ 0000 \\ \hline 0100011 \end{array}$$

$$0100011 = 1.2^5 + 1.2^1 + 1.2^0 = 35_{(10)}$$

19

Biểu Diễn Số Có Dấu

- Trong hệ D,để biểu diễn số có dấu, ta dùng :
  - Dấu + chỉ số dương
  - Dấu – chỉ số âm
- Trong hệ B, không có dấu +, - để biểu diễn số có dấu, dùng bit MSB làm dấu :
  - Bit 1 chỉ số âm
  - Bit 0 chỉ số dương
  - Các bit còn lại biểu diễn độ lớn
- Ví dụ : 0 1 1 0 1 0 0 = + 52 1 1 1 0 1 0 0 = - 52

  bit dấu bit dấu

20

Biểu Diễn Số Có Dấu

- Nhận xét :
  - Số 0 có hai kết quả : 000000(+0) và 100000 (-0)
  - Một số phép tính cho kết quả sai
- Ví dụ : với số có dấu 5 bit, ta xét :

	0 1 0 0 0 (+ 8)		0 1 0 0 0 (+8)
+	0 1 0 1 0 (+ 10)	+	1 0 0 1 0 (-2)
	1 0 0 1 0 (- 2) sai		1 1 0 1 0 (-10) sai

- Do đó phải biểu diễn số có dấu theo phương pháp khác là Bù – 1 và Bù – 2.

21

Biểu Diễn Số Có Dấu

- Số bù – 1 : bù – 1 của một số A là B sao cho khi lấy A+B, ta được tổng là 1 ở tất cả các vị trí.
- Ví dụ : A = 10110 , B = 01001
- Tìm bù – 1 của một số :
  - Đảo bit 1 thành 0
  - Đảo bit 0 thành 1
- Ví dụ : tìm bù – 1 của : 110010

  001101

22

Biểu Diễn Số Có Dấu

- Bù – 2 của một số = Bù – 1 của số đó + 1
- Ví dụ : tìm bù – 2 của : 10010
- Bù – 1 : 01101
- Bù – 2 : + 1

  01110
- Biểu diễn số có dấu dùng bù – 1: dùng quy tắc
  - Dùng bit MSB làm dấu : 0 – dương, 1 – âm
  - Các bit còn lại biểu diễn giá trị thực của số dương hoặc bù – 1 của số âm.
  - Số có dấu n bit, có tầm trị : $-(2^{n-1} - 1)$ tới $(2^{n-1} - 1)$

23

Biểu Diễn Số Có Dấu

- Ví dụ : Dùng 6 bit

17 : 010001	26 : 011010
-17 : 101110	-26 : 100101

- Biểu diễn số có dấu dùng bù – 2 :
  - Bit MSB làm bit dấu : 0 – dương, 1 – âm
  - Các bit còn lại biểu diễn giá trị số dương hoặc bù – 2 của số âm
  - Số n bit, có tầm trị : -2^{n-1} tới $(2^{n-1} - 1)$
- Ví dụ :

17 : 010001	26 : 011010
-17 : 101111	-26 : 100110

24

Cộng Trừ Số Có Dấu Dùng Bù -1

- **A+B : cộng bình thường, kể cả bit dấu. Nếu kết quả bị tràn, đem MSB cộng vào LSB.**
- **Ví dụ : Thực hiện phép tính:**

			11
13	**001101**	**-13**	**110010**
+11	**+001011**	**-11**	**+ 110100**
+ 24	**011000**	**-24**	**100110**
			+ 1
			100111

Cộng Trừ Số Có Dấu Dùng Bù -1

- A – B = A + (-B) : thực hiện giống như cộng.
- Ví dụ : Thực hiện phép trừ

		11
6	0110	0110
- 3	-0011	+1100
3		0010
		+ 1
		0011

Cộng Trừ Số Có Dấu Dùng Bù -2

- A+B : Cộng bình thường, kể cả bit dấu. Nếu kết quả bị tràn (tồn tại bit trọng số 2^n) thì bit này bị bỏ.
- Ví dụ :

	1		11 1
12	001100	-12	110100
+ 9	+ 001001	+ -9	+110111
21	010101	-21	1101011

Bỏ : Kết quả là : 101011

Cộng Trừ Số Có Dấu Dùng Bù -2

- A-B = A + (-B) : Thực hiện giống phép cộng.
- Ví dụ :

		11 1
10	00101	01010
- 5	- 01010	+ 11011
5		100101

Bỏ : kết quả : 00101

- Mã BCD (Binary Coded Decimal) :Biểu diễn mỗi chữ số thập phân bằng một tổ hợp 4 bit. Trong bảng sau :

STT	D	Mã BCD
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001

- **Các tổ hợp còn lại không dùng :**
- **1010, 1011, 1100, 1101, 1110, 1111**

gọi là các tổ hợp cấm.

- Ví dụ :

1941D = 11110010101 (dùng 11 bit – bình thường)

1941D = 0001 1001 0100 0001BCD (dùng 16 bit)

- Các phép toán trên số BCD
- A+B = S :
- Quy tắc
  - Số nhớ ở Decade thấp được chuyển lên decade cao và hiệu đính decade thấp
  - Nếu decade nào của tổng > 9 cũng phải hiệu đính
  - Việc hiệu đính được thực hiện bằng cách cộng cho 6

- Ví dụ :

18	0001	1000	
+ 26	+ 0010	0110	
44	0011	1110	
		+ 0110	hiệu đính S0 (decade S0)
	0100	0100	

- Ví dụ :

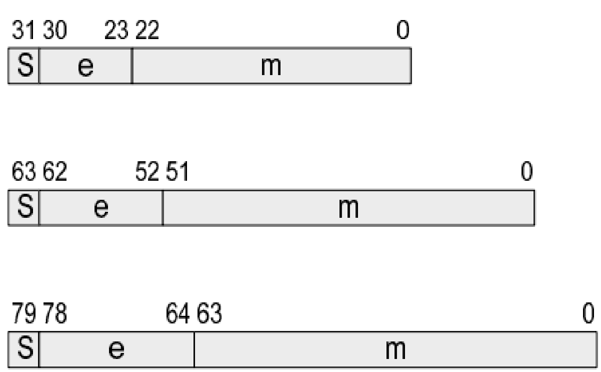
	1		(số nhớ từ decade S0)
28	0010	1000	
+ 19	0001	1001	
47	0100	0001	
	+	0110	hiệu đính S0
	0100	0111	

Số có dấu chấm động

- Nguyên tắc chung
  - Floating point number (số có dấu chấm động): dùng để biểu diễn số thực
  - Tổng quát : một số thực X được biểu diễn theo kiểu số chấm động như sau :
  - $X=M*R^E$
    - M : là phần định trị
    - R : là cơ số (radix)
    - E : là phần mũ (Exponent)

Chuẩn IEEE754/85

- Cơ số R=2
- Có các dạng :
  - Dạng 32 bit
  - Dạng 44 bit
  - Dạng 64 bit
  - Dạng 80 bit



Dạng 32bit

31	30 … 23	22 … 0
S	e	m

- S là bit dấu
  - S=0 : số dương
  - S=1 : số âm
- e (8 bit) là mã axcess 127 của phần mũ E :
  - e=E+127 → E= e – 127
  - Giá trị 127 được gọi là độ lệch (bias)
- m (23 bit) : là phần lẻ của phần định trị M
  - M=1.m
- Công thức xác định giá trị của số thực :

  $X=(-1)^{s} * 1.m*2^{e-127}$

Ví dụ 1:

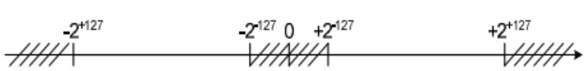
- Xác định giá trị của số thực được biểu diễn bằng 32 bit như sau :
- 1100 0001 0101 0110 0000 0000 0000 0000
- Giải :
  - S=1 → số âm
  - e=1000 0010 = 130 → E=130 – 127 = 3
  - Vậy $X=-1.101011*2^3$ = - 1101.011 = -13.375

Ví dụ 2

- Biểu diễn số thực X=83.75 về dạng số chấm động IEEE754 – 32 bit
- Giải :
- X=83.75 = 1010011.11 = 1.01001111×2^6
- ta có :
  - s=0 vì đây là số dương
  - E = e – 127 = 6 → e = 127+6=133= 1000 0101
  - Vậy : X=0100 0010 1010 0111 1000 000 000 000

Các quy ước đặc biệt

- Các bit của e bằng 0, các bit của m bằng 0 thì X= ± 0
  x000 0000 0000 0000 0000 0000 0000 0000 →X=±0
- Các bit của e bằng 1, các bit của m bằng 0, thì X= ±∞
  x111 1111 1000 0000 0000 0000 0000 0000 → X= ±∞
- Các bit của e bằng 1, m có ít nhất 1 bit bằng 1 thì nó không biểu diễn cho số nào cả (NaN – not a number).
- Tầm giá trị biểu diễn :



2^{-127} đến 2^{+127}

10^{-38} đến 10^{+38}

Dạng 64 bit

63	62 52	51 0
S	e	m

- S là bit dấu
- e (11 bit) là mã axcess – 1023 của phần mũ E

  → E=e – 1023
- m (52 bit) là phần lẻ của phần định trị M
- Giá trị của số thực X
- $X = (-1)^s * 1.m * 2^{e-1023}$
- Tầm giá trị biểu diễn : 10^{-308} đến 10^{+308}

Cổng logic và phép tính

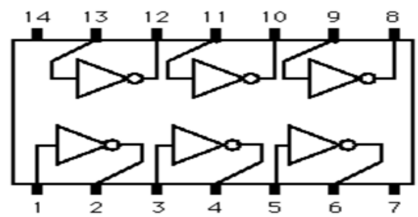
- Các phép tính logic

X	Y	X+Y	X.Y	$\overline{X}$	X⊕Y
0	0	0	0	1	0
0	1	1	0	1	1
1	0	1	0	0	1
1	1	1	1	0	0

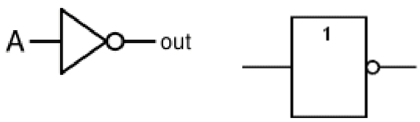
- Phép + gọi là phép OR
- Phép . Gọi là phép AND
- Phép _ gọi là phép NOT
- Phép ⊕ gọi là phép XOR

Các Cổng Logic

- Cổng ĐẢO (NOT inverter)
  - Ký hiệu:
  - Bảng sự thật:
  - Bảng sự thật



IC 7404



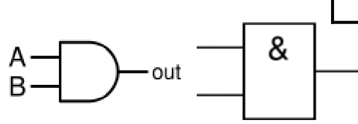
  - Quan hệ vào ra :

$$Out = \overline{A}$$

A	out
0	1
1	0

Các Cổng Logic

- Cổng AND
  - Chức năng:Thực hiện phép nhân
  - Đầu ra chỉ bằng 1 khi tất cả các đầu vào bằng 1
  - Ký hiệu:

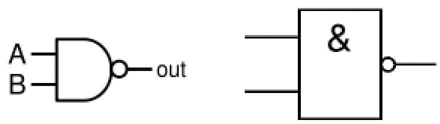


  - Bảng sự thật:
  - Quan hệ vào ra : out = A . B

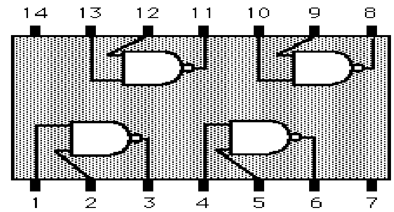
A	B	out
0	0	0
0	1	0
1	0	0
1	1	1

Các Cổng Logic

- Cổng NAND
  - Chức năng:Thực hiện phép ĐẢO của phép toán logic VÀ
  - Đầu ra chỉ bằng 0 khi tất cả các đầu vào bằng 1
  - Ký hiệu:



  - Bảng sự thật:
  - Quan hệ vào ra : out = $\overline{A . B}$



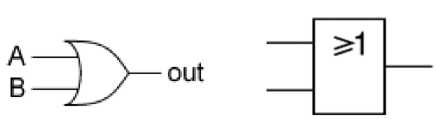
7400

A	B	out
0	0	1
0	1	1
1	0	1
1	1	0

43

Các Cổng Logic

- Cổng OR
  - Chức năng:Thực hiện phép toán logic OR
  - Đầu ra chỉ bằng 0 khi tất cả các đầu vào bằng 0
  - Ký hiệu:



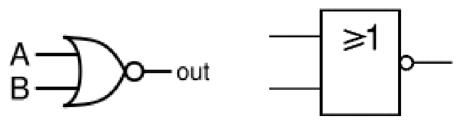
A	B	out
0	0	0
0	1	1
1	0	1
1	1	1

  - Bảng sự thật:
  - Quan hệ vào ra : out = A + B

44

Các Cổng Logic

- Cổng NOR
  - Thực hiện phép ĐẢO của phép toán logic HOẶC
  - Đầu ra chỉ bằng 1 khi tất cả các đầu vào bằng 0
  - Ký hiệu:



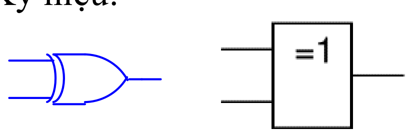
A	B	out
0	0	1
0	1	0
1	0	0
1	1	0

  - Bảng sự thật:
  - Hàm Boole : out = $\overline{A + B}$

45

Các Cổng Logic

- Cổng EXOR (Exclusive OR)
  - Chức năng:Thực hiện biểu thức logic XOR - hay còn là phép cộng module 2
  - Đầu ra chỉ bằng 0 khi tất cả các đầu vào giống nhau
  - Ký hiệu:



A	B	out
0	0	0
0	1	1
1	0	1
1	1	0

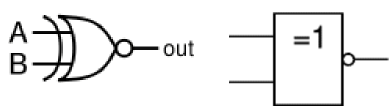
  - Bảng sự thật:
  - Hàm Boole:

$$out = A \oplus B = \overline{A}.B + A.\overline{B}$$

46

Các Cổng Logic

- Cổng EXNOR
  - Chức năng:Thực hiện phép ĐẢO của phép toán XOR
  - Đầu ra chỉ bằng 1 khi tất cả các đầu vào giống nhau
  - Ký hiệu:



  - Bảng sự thật:
  - Hàm Boole : $out = \overline{A \oplus B} = A.B + \overline{A}.\overline{B}$

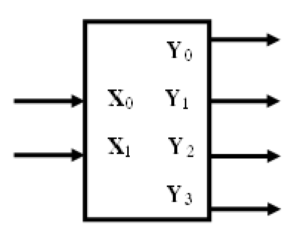
A	B	out
0	0	1
0	1	0
1	0	0
1	1	1

47

Mạch Giải Mã

- Mạch giải mã (Decoder)
  - Mỗi ngõ ra chỉ tích cực ứng với 1 tổ hợp các ngõ vào
- Ví dụ : Giải mã 2 sang 4, ngõ ra tích cực cao
  - Sơ đồ khối
  - Bảng sự thật (trang sau)

X_1	X_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

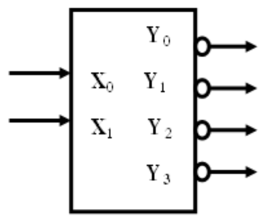


48

Mạch Mã Hóa và Giải Mã

- Ví dụ : Mạch giải mã 2 sang 4, ngõ ra tích cực thấp
  - Sơ đồ khối
  - Bảng sự thật

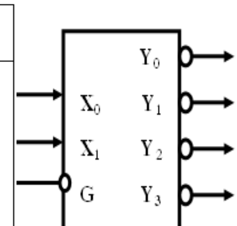
X_1	X_0	$\overline{Y_3}$	$\overline{Y_2}$	$\overline{Y_1}$	$\overline{Y_0}$
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1



Mạch Mã Hóa và Giải Mã

- Ví dụ : Mạch giải mã 2 sang 4 có ngõ ra tích cực thấp, 1 vào điều khiển tích cực thấp.
  - Sơ đồ khối
  - Bảng sự thật
  - Hàm Boole
  - Sơ đồ (tự vẽ)

$\overline{G}$	X_1	X_0	$\overline{Y_3}$	$\overline{Y_2}$	$\overline{Y_1}$	$\overline{Y_0}$
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1
1	x	x	1	1	1	1



$$\overline{Y_0} = \overline{G} + x_1 + x_0 = \overline{\overline{\overline{G}}\,\overline{x_1}\,\overline{x_0}} \qquad \overline{Y_1} = \overline{G} + x_1 + \overline{x_0} = \overline{\overline{\overline{G}}\,\overline{x_1}\,x_0}$$

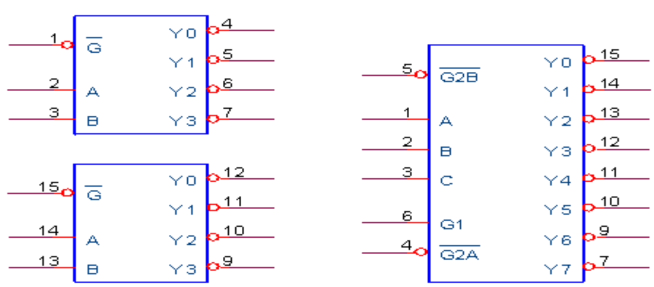
$$\overline{Y_2} = \overline{G} + \overline{x_1} + x_0 = \overline{\overline{\overline{G}}\,x_1\,\overline{x_0}} \qquad \overline{Y_3} = \overline{G} + \overline{x_1} + \overline{x_0} = \overline{\overline{\overline{G}}\,x_1\,x_0}$$

Lưu ý

- Tích cực cao
  - Khi tích cực thì ở mức cao
  - Không tích cực thì ở mức thấp
- Tích cực thấp
  - Khi tích cực ở mức thấp
  - Không tích cực ở mức cao

Mạch Giải Mã

- Giới thiệu vi mạch



74LS139 74LS138

74LS139 : Gồm 2 mạch giải mã 2 – 4

74LS138 : Mạch giải mã 3 sang 8

52

Mạch Giải Mã 74138

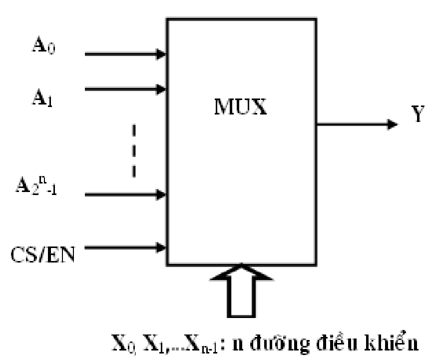
- Bảng sự thật của 74LS138

G1	$\overline{G2A}$	$\overline{G2B}$	C	B	A	$\overline{Y7}$	$\overline{Y6}$	$\overline{Y5}$	$\overline{Y4}$	$\overline{Y3}$	$\overline{Y2}$	$\overline{Y1}$	$\overline{Y0}$
0	X	x	x	x	x	1	1	1	1	1	1	1	1
x	1	x	x	x	x	1	1	1	1	1	1	1	1
x	x	1	x	x	x	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	0	1	1	1
1	0	0	1	0	0	1	1	1	0	1	1	1	1
1	0	0	1	0	1	1	1	0	1	1	1	1	1
1	0	0	1	1	0	1	0	1	1	1	1	1	1
1	0	0	1	1	1	0	1	1	1	1	1	1	1

53

Mạch Ghép kênh- Phân Kênh

- Mạch ghép kênh (MUX = Multiplexer)
  - Sơ đồ khối như hình vẽ
  - Có 2^n ngõ vào
  - Có 1 ngõ ra
  - Có n ngõ điều khiển
  - Có thể có ngõ vào CS

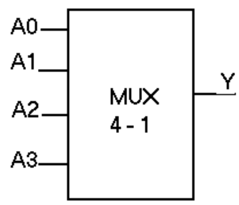


  - Tại mỗi thời điểm, chỉ 1 ngõ vào được nối tới ngõ ra, tổ hợp các ngõ điều khiển xác định ngõ vào nào được nối tới ngõ ra.

54

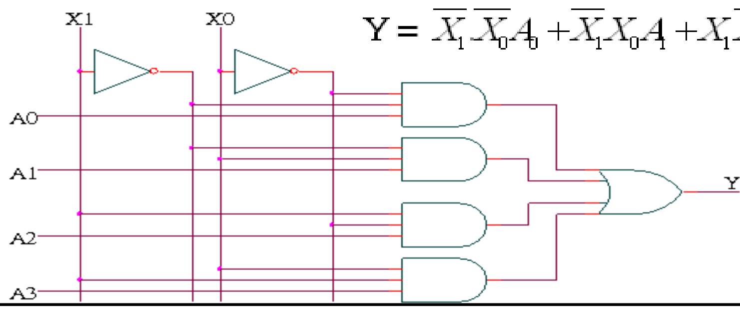
Mạch Ghép - Phân Kênh

- Ví dụ : MUX 4 – 1
  - Sơ đồ khối
  - Bảng sự thật
  - Hàm Boole
  - Sơ đồ mạch



X_1	X_0	Y
0	0	A_0
0	1	A_1
1	0	A_2
1	1	A_3

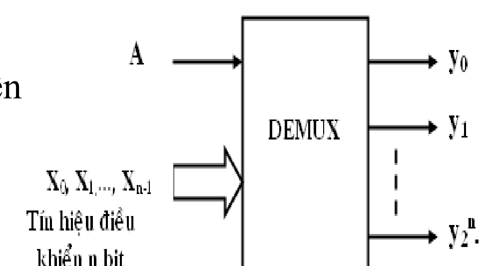
$$Y = \overline{X_1}\,\overline{X_0}A_0 + \overline{X_1}X_0A_1 + X_1\overline{X_0}A_2 + X_1X_0A_3$$



55

Mạch Ghép kênh - Phân Kênh

- Mạch phân kênh (DEMUX=DeMultiPlexer)
  - Có 1 ngõ vào
  - Có n ngõ vào điều khiển
  - Có 2^n ngõ ra



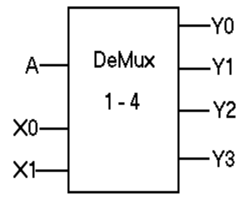
  - Tại mỗi thời điểm chỉ có một ngõ ra được nối với ngõ vào, tổ hợp các ngõ vào điều khiển quyết định ngõ ra nào được nối tới ngõ vào.

56

Mạch Ghép kênh - Phân Kênh

- Ví dụ : DEMUX 1 – 4
  - Sơ đồ khối
  - Bảng sự thật
  - Hàm Boole
  - Sơ đồ mạch

X_1	X_0	Y_0	Y_1	Y_2	Y_3
0	0	A	0	0	0
0	1	0	A	0	0
1	0	0	0	A	0
1	1	0	0	0	A



$Y_0 = \overline{X_1}\,\overline{X_0}A$ $\quad Y_1 = \overline{X_1}X_0A$

$Y_2 = X_1\overline{X_0}A$ $\quad Y_3 = X_1X_0\,A$