

ĐẠI HỌC KHOA HỌC TỰ NHIÊN
ĐẠI HỌC QUỐC GIA TP. HCM

KHOA CÔNG NGHỆ THÔNG TIN
LỚP 19CTT4

Tài liệu ôn thi cuối kỳ môn
Hệ thống máy tính

PHẦN 4: MẠCH LOGIC

Ngày 05/07/2021

Mục lục

1	Khái niệm mạch số	2
1.1	Cổng luận lý(Logic gate)	2
1.2	Bảng chân trị	2
1.3	Lược đồ Venn	4
1.4	Một số đẳng thức cơ bản	4
2	Mạch tổ hợp	4
2.1	Khái niệm	4
2.2	Độ trễ mạch	4
2.3	Các bước thiết kế	4
2.3.1	SOP - Sum of Products	5
2.3.2	POS - Product of Sum	6
2.3.3	Đơn giản hóa hàm logic	6
2.3.3.1	Đại số Bun	6
2.3.3.2	Bản đồ Karnaugh	6
2.3.3.3	Đơn giản hàm theo dạng SOP	7
2.3.3.4	Đơn giản hàm theo dạng POS	8
2.4	Một số bài tập thiết kế mạch	9
2.5	Một số mạch tổ hợp cơ bản	11
2.5.1	Mạch toàn cộng (Full adder)	11
2.5.2	Mạch giải mã(Decoder)	11
2.5.3	Mạch mã hóa(Encoder)	11

1 Khái niệm mạch số

Là thiết bị điện tử hoạt động với **2 mức điện áp**:

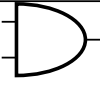
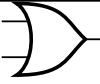
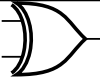

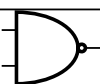
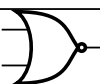
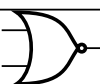
- **Cao**: thể hiện bằng giá trị luận lý (quy ước) là **1**.
- **Thấp**: thể hiện bằng giá trị luận lý (quy ước) là **0**.

Được xây dựng từ những thành phần cơ bản là **cổng luận lý (logic gate)**

- Cổng luận lý là thiết bị điện tử gồm 1/ nhiều tín hiệu đầu vào (input) - 1 tín hiệu đầu ra output.
- $output = F(input_1, input_2, ..., input_n)$.
- Tùy thuộc vào cách xử lý của hàm F sẽ tạo ra nhiều loại cổng luận lý.

Hiện nay linh kiện cơ bản tạo ra mạch số là **transistor**.

1.1 Cổng luận lý (Logic gate)

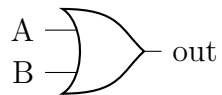
Tên cổng	Hình vẽ đại diện	Hàm đại số Bun
AND		$x.y$ hay xy
OR		$x + y$
XOR		$x \oplus y$
NOT		x' hay \bar{x}
NAND		$(x.y)'$ hay \overline{xy}
NOR		$(x + y)'$ hay $\overline{x + y}$
NXOR		$(x \oplus y)'$ hay $\overline{x \oplus y}$

1.2 Bảng chân trị

- **AND**

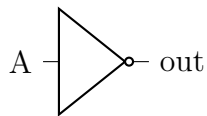

A	B	out
0	0	0
0	1	0
1	0	0
1	1	1

- OR**



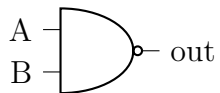
A	B	out
0	0	0
0	1	1
1	0	1
1	1	1

- NOT**



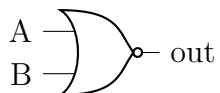
A	out
0	1
1	0

- NAND**



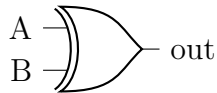
A	B	out
0	0	1
0	1	1
1	0	1
1	1	0

- NOR**



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

- XOR**



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

1.3 Lược đồ Venn

1.4 Một số đẳng thức cơ bản

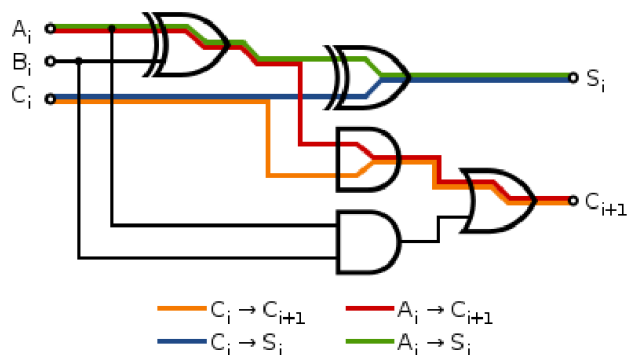
2 Mạch tổ hợp

2.1 Khái niệm

- Gồm n ngõ vào (input); m ngõ ra (output)
 - Mỗi ngõ ra là 1 hàm luận lý của các ngõ vào
- Mạch tổ hợp không mang tính ghi nhớ: Ngõ ra chỉ phụ thuộc vào Ngõ vào hiện tại, không xét những giá trị trong quá khứ

2.2 Độ trễ mạch

- Độ trễ mạch (**Propagation delay/ gate delay**) = Thời gian điểm tín hiệu ra ổn định – thời điểm tín hiệu vào ổn định.
- Mục tiêu thiết kế mạch: làm giảm thời gian độ trễ mạch.



2.3 Các bước thiết kế

Thường trải qua 3 bước:

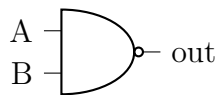
- **Bước 1:** Lập bảng chân trị:

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

- **Bước 2:** Viết hàm luận lý

$$F = \overline{AB}$$

- **Bước 3:** Vẽ sơ đồ mạch và thử nghiệm



2.3.1 SOP - Sum of Products

Giả sử đã có bảng chân trị cho mạch n đầu vào x_1, \dots, x_n và 1 đầu ra f .

- Ta dễ dàng lập công thức (hàm) logic theo thuật toán sau:
 - Ứng với mỗi hàng của bảng chân trị có đầu ra = 1, ta tạo thành 1 tích có dạng $u_1.u_2...u_n$ với:

$$f(x) = \begin{cases} x_i, & \text{nếu } x_i = 1 \\ \overline{x_i}, & \text{nếu } x_i = 0 \end{cases}$$

- Cộng các tích tìm được lại thành tổng -> Công thức của f

Ví dụ:

STT	x_1	x_2	x_3	f
0	0	0	0	0
1	0	0	1	1 $\rightarrow \overline{x_1}.\overline{x_2}.x_3$
2	0	1	0	1 $\rightarrow \overline{x_1}.x_2.\overline{x_3}$
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1 $\rightarrow x_1.\overline{x_2}.x_3$
6	1	1	0	0
7	1	1	1	0

$$f = \overline{x_1}.\overline{x_2}.x_3 + \overline{x_1}.x_2.\overline{x_3} + x_1.\overline{x_2}.x_3$$

2.3.2 POS - Product of Sum

- Trường hợp số hàng có giá trị **đầu ra = 1 nhiều hơn = 0**, ta có thể đặt $g = \bar{f}$.
- Viết công thức dạng **SOP** cho g .
- Lấy $f = \bar{g} = \bar{\bar{f}}$ để có công thức dạng POS (Tích các tổng) của f .

Ví dụ: (Thêm sau)

2.3.3 Đơn giản hóa hàm logic

- Sau khi viết được hàm logic, ta có thể vẽ sơ đồ của mạch tổ hợp từ những công luận lý cơ bản.
 - Ví dụ: $f = xy + xz$
- Tuy nhiên ta có thể viết lại hàm logic sao cho sơ đồ mạch sử dụng **ít cổng hơn**
 - Ví dụ: $f = xy + xz = x(y + z)$
- Cách đơn giản hóa hàm tổng quát? Một số cách phổ biến:
 - Dùng đại số Bun (Xem lại bảng "Một số đẳng thức cơ bản" để áp dụng)
 - Dùng bản đồ Karnaugh

2.3.3.1 Đại số Bun

- Dùng các phép biến đổi đại số Bun để lược giản hàm logic
- Khuyết điểm:
 - Không có cách là tổng quát cho mọi bài toán
 - Không chắc kết quả cuối cùng đã tối giản chưa

Ví dụ: Đơn giản hóa các hàm sau: $F(x, y, z) = xyz + \bar{x}yz + x\bar{y}z + xy\bar{z}$

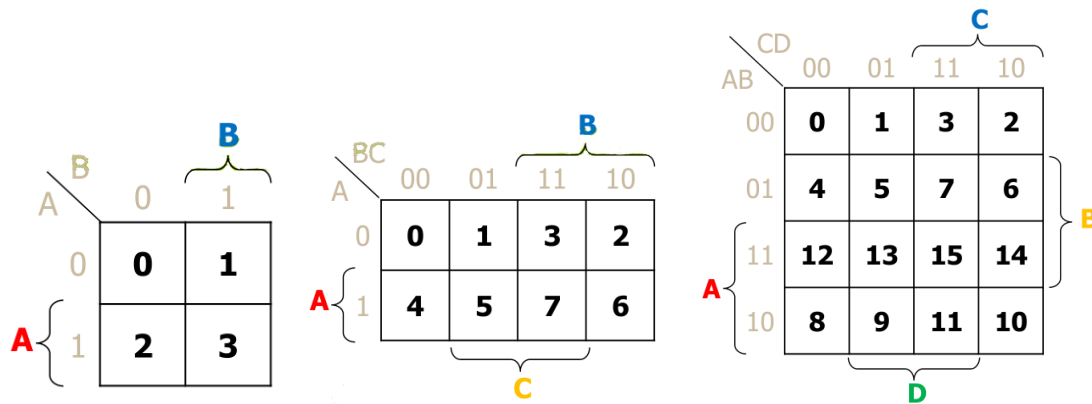
$$\begin{aligned} F(x, y, z) &= xyz + \bar{x}yz + x\bar{y}z + xy\bar{z} \\ &= \dots \end{aligned}$$

2.3.3.2 Bản đồ Karnaugh

Mỗi tổ hợp biến trong bảng chân trị gọi là bộ trị (tạm hiểu là 1 dòng)

- Biểu diễn hàm có n biến thì sẽ cho ra tương ứng 2^n bộ trị, với vị trí các bộ trị được đánh số từ 0
- Thông tin trong bảng chân trị có thể **cô đọng** bằng cách:
 - Liệt kê vị trí các bộ trị (**midterm**) với giá trị đầu ra = **1** (SOP)
 - Liệt kê vị trí các bộ trị (**maxterm**) với giá trị đầu ra = **0** (POS)

Các dạng bản đồ Karnaugh cơ bản:

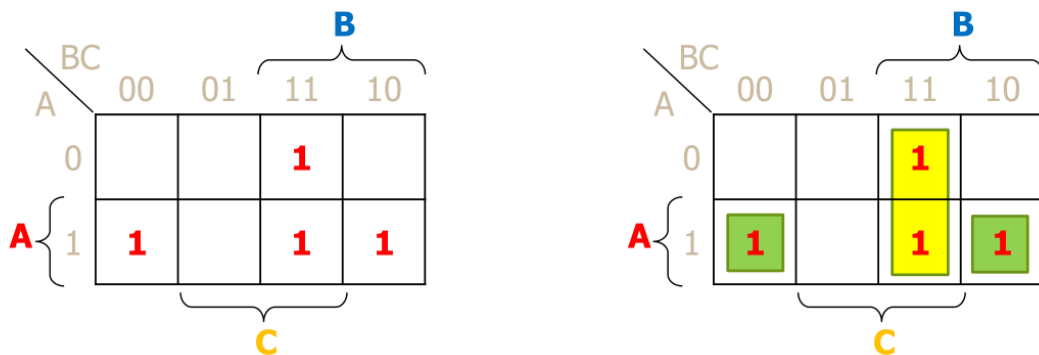


Ví dụ: $F(A, B, C) = \Sigma(1, 4, 5, 6, 7)$

2.3.3.3 Đơn giản hàm theo dạng SOP

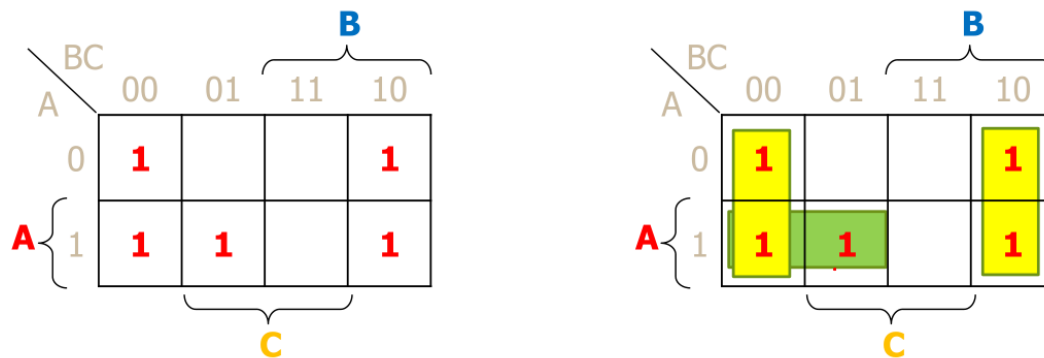
- Hàm logic F biểu diễn bằng chân trị được đưa vào bản đồ bằng các trị 1 tương ứng.
- Các ô liên kề có giá trị 1 được gom thành nhóm sao cho mỗi nhóm sau khi gom có tổng số ô là lũy thừa của 2 (2, 4, 8,...).
- Các nhóm có thể dùng chung ô có giá trị 1 để tạo thành nhóm lớn hơn. Cố gắng tạo những nhóm lớn nhất có thể.
- Nhóm 2/4/8 ô sẽ đơn giản bớt 1/2/3 biến trong số hạng.
- Mỗi nhóm biểu diễn 1 số hạng nhân (Product), Cộng(Sum - OR), các số hạng này ta sẽ được biểu thức tối giản của hàm logic F.

Ví dụ 1: $F(A, B, C) = \Sigma(3, 4, 6, 7)$



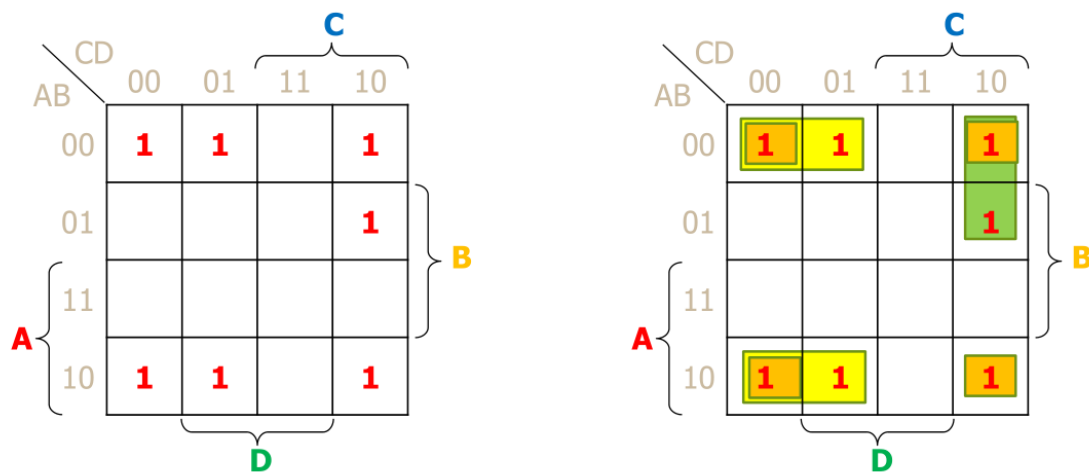
$$F(A, B, C) = \overline{B}.D + \overline{B}.C + \overline{A}.C.D$$

Ví dụ 2: $F(A, B, C) = \Sigma(0, 2, 4, 5, 6)$



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

Ví dụ 3: $F(A, B, C, D) = \Sigma(0, 1, 2, 6, 8, 9, 10)$



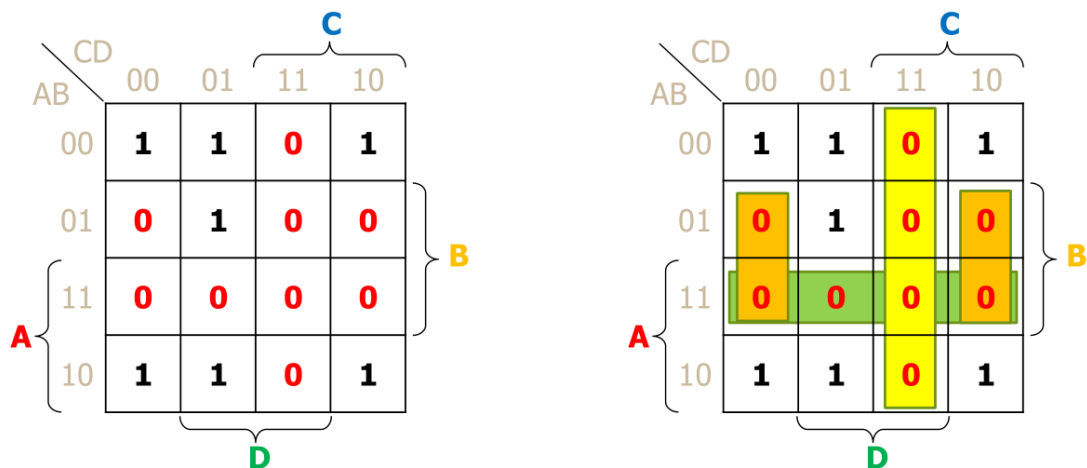
$$F(A, B, C, D) = \overline{B} \cdot \overline{D} + \overline{B} \cdot \overline{C} + \overline{A} \cdot C \cdot \overline{D}$$

2.3.3.4 Đơn giản hàm theo dạng POS

- Đôi khi biểu diễn dạng tổng các tích (SOP) sẽ khó làm khi số bộ trị có đầu ra = 1 < số bộ trị có đầu ra = 0
 - Dùng phương pháp tích các tổng (POS)

- Hoàn toàn giống phương pháp đơn giản hàm theo dạng SOP, chỉ khác ta nhóm các ô liền kề = 0 thay vì 1
 - Tìm được \overline{F}
 - $\rightarrow F = \overline{\overline{F}}$

Ví dụ: $F(A, B, C, D) = \Sigma(0, 1, 2, 5, 8, 9, 10)$



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

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

2.4 Một số bài tập thiết kế mạch

Ví dụ 1: Thiết kế mạch cộng 2 bits không nhớ

- Bước 1: Lập bảng chân trị

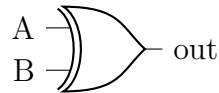
A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

- Bước 2: Lập biểu thức

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

$$= A \oplus B$$

- **Bước 3:** Vẽ mạch



Ví dụ 2: Thiết kế mạch kiểm tra số nguyên không dấu 3 bits có chia hết cho 3

- **Bước 1:** Lập bảng chân trị

Hệ 10	A	B	C	F
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

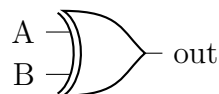
- **Bước 2:** Lập biểu thức

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

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

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

- **Bước 3:** Vẽ mạch



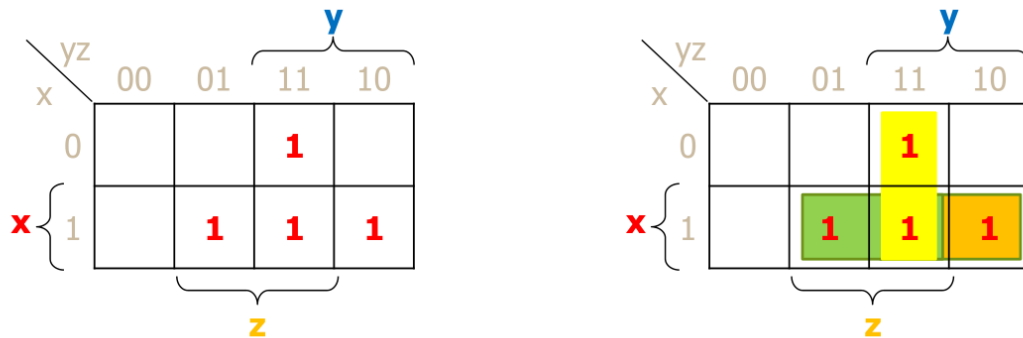
Ví dụ 3: Thiết kế mạch tổ hợp gồm 3 ngõ vào, 1 ngõ ra, sao cho giá trị logic ở ngõ ra là giá trị nào chiếm đa số trong các ngõ vào

- **Bước 1:** Lập bảng chân trị

Hệ 10	A	B	C	F
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

$$F(x, y, z) = \Sigma(3, 5, 6, 7)$$

- **Bước 2:** Lập biểu thức



$$\begin{aligned} F &= \overline{A}.\overline{B}.\overline{C} + \overline{A}.B.C + A.B.\overline{C} \\ &= \overline{A}.\overline{B}.\overline{C} + B.(\overline{A}.C + A.\overline{C}) \\ &= \overline{A}.\overline{B}.\overline{C} + B.(A \oplus B) \end{aligned}$$

- **Bước 3:** Vẽ mạch

2.5 Một số mạch tổ hợp cơ bản

2.5.1 Mạch toàn cộng (Full adder)

- Mạch tổ hợp thực hiện phép **cộng số học 3 bit**.
- Gồm **3 ngõ vào** (A,B: bit cần cộng - C_i : bit nhớ) và **2 ngõ ra** (kết quả có thể từ 0 đến 3 với giá trị 2 và 3 cần 2 bit biểu diễn -S: ngõ tổng, C_0 : ngõ nhớ)

2.5.2 Mạch giải mã(Decoder)

2.5.3 Mạch mã hóa(Encoder)