

KỸ THUẬT SỐ

ThS. Phạm Thị Đan Ngọc
Khoa Kỹ Thuật Điện Tử 2
Email: ngocptd@ptithcm.edu.vn

Ngày 26 tháng 10 năm 2014

Chương 3: Mạch tổ hợp

- 1 Khái niệm chung
- 2 Phân tích và thiết kế mạch logic tổ hợp
- 3 Mạch mã hóa và giải mã
- 4 Mạch ghép kênh và phân kênh
- 5 Mạch số học
- 6 Mạch tạo bit và kiểm tra chẵn - lẻ
- 7 Đơn vị số học và ALU

3.1 Khái niệm chung

3.1 Khái niệm chung

- Mạch tổ hợp là mạch số mà tín hiệu ngõ ra chỉ phụ thuộc tín hiệu ngõ vào tại thời điểm hiện tại.

3.1 Khái niệm chung

- Mạch tổ hợp là mạch số mà tín hiệu ngõ ra chỉ phụ thuộc tín hiệu ngõ vào tại thời điểm hiện tại.
- Mạch tổ hợp còn được gọi là mạch số không nhớ.

3.1 Khái niệm chung

- Mạch tổ hợp là mạch số mà tín hiệu ngõ ra chỉ phụ thuộc tín hiệu ngõ vào tại thời điểm hiện tại.
- Mạch tổ hợp còn được gọi là mạch số không nhớ.
- Được cấu trúc bằng những phần tử logic cơ bản.

3.1 Khái niệm chung

- Mạch tổ hợp là mạch số mà tín hiệu ngõ ra chỉ phụ thuộc tín hiệu ngõ vào tại thời điểm hiện tại.
- Mạch tổ hợp còn được gọi là mạch số không nhớ.
- Được cấu trúc bằng những phần tử logic cơ bản.
- Các phương pháp biểu diễn hàm chức năng của mạch tổ hợp: hàm logic, bảng trạng thái, bìa Karnaugh, giản đồ xung, ...

3.1 Khái niệm chung

3.1 Khái niệm chung

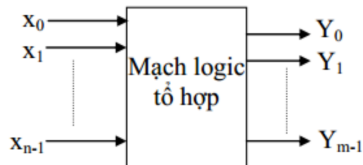
Hệ phương trình tổng quát:

$$Y_0 = f_1(x_0, x_1, \dots, x_{n-1});$$

$$Y_1 = f_2(x_0, x_1, \dots, x_{n-1});$$

...

$$Y_{m-1} = f_{m-1}(x_0, x_1, \dots, x_{n-1}).$$



3.2 Phân tích và thiết kế mạch logic tổ hợp

3.2 Phân tích và thiết kế mạch logic tổ hợp

3.2.1 Phân tích mạch tổ hợp

3.2 Phân tích và thiết kế mạch logic tổ hợp

3.2.1 Phân tích mạch tổ hợp

- Phân tích mạch logic tổ hợp là từ sơ đồ cho trước, xác định chức năng, dạng sóng, tính năng kỹ thuật, ... của mạch. Từ đó, có thể rút gọn, thay đổi dạng thực hiện của mạch điện để có được phương án tối ưu.

3.2 Phân tích và thiết kế mạch logic tổ hợp

3.2.1 Phân tích mạch tổ hợp

- Phân tích mạch logic tổ hợp là từ sơ đồ cho trước, xác định chức năng, dạng sóng, tính năng kỹ thuật, ... của mạch. Từ đó, có thể rút gọn, thay đổi dạng thực hiện của mạch điện để có được phương án tối ưu.
- Về phương pháp:

3.2 Phân tích và thiết kế mạch logic tổ hợp

3.2.1 Phân tích mạch tổ hợp

- Phân tích mạch logic tổ hợp là từ sơ đồ cho trước, xác định chức năng, dạng sóng, tính năng kỹ thuật, ... của mạch. Từ đó, có thể rút gọn, thay đổi dạng thực hiện của mạch điện để có được phương án tối ưu.
- Về phương pháp:
 - Với mạch đơn giản: tiến hành lập bảng trạng thái, viết biểu thức, rút gọn, tối ưu (nếu cần) và vẽ lại mạch điện.

3.2 Phân tích và thiết kế mạch logic tổ hợp

3.2.1 Phân tích mạch tổ hợp

- Phân tích mạch logic tổ hợp là từ sơ đồ cho trước, xác định chức năng, dạng sóng, tính năng kỹ thuật, ... của mạch. Từ đó, có thể rút gọn, thay đổi dạng thực hiện của mạch điện để có được phương án tối ưu.
- Về phương pháp:
 - Với mạch đơn giản: tiến hành lập bảng trạng thái, viết biểu thức, rút gọn, tối ưu (nếu cần) và vẽ lại mạch điện.
 - Với mạch phức tạp: **tiến hành phân đoạn mạch** để viết biểu thức, sau đó rút gọn, tối ưu (nếu cần) và vẽ lại mạch điện.

3.2 Phân tích và thiết kế mạch logic tổ hợp

3.2 Phân tích và thiết kế mạch logic tổ hợp

3.2.2 Thiết kế mạch tổ hợp

3.2 Phân tích và thiết kế mạch logic tổ hợp

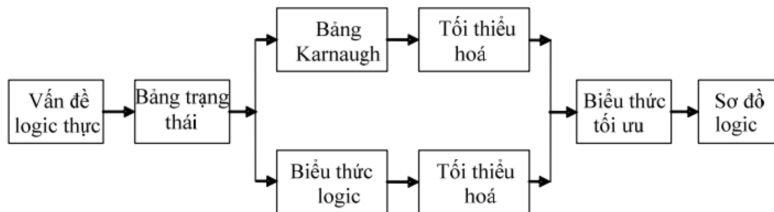
3.2.2 Thiết kế mạch tổ hợp

- Thiết kế mạch là bài toán được đưa ra từ các yêu cầu (chức năng, dạng sóng, tính năng kỹ thuật, ...) xây dựng cơ sở mạch thực hiện.

3.2 Phân tích và thiết kế mạch logic tổ hợp

3.2.2 Thiết kế mạch tổ hợp

- Thiết kế mạch là bài toán được đưa ra từ các yêu cầu (chức năng, dạng sóng, tính năng kỹ thuật, ...) xây dựng cơ sở mạch thực hiện.
- Phương pháp thiết kế



3.2 Phân tích và thiết kế mạch logic tổ hợp

3.2 Phân tích và thiết kế mạch logic tổ hợp

3.2.2 Thiết kế mạch tổ hợp

3.2 Phân tích và thiết kế mạch logic tổ hợp

3.2.2 Thiết kế mạch tổ hợp

- Ví dụ: Thiết kế hệ thống mạch logic mô tả quá trình tắt mở đèn tại hai tầng lầu trong nhà.

3.2 Phân tích và thiết kế mạch logic tổ hợp

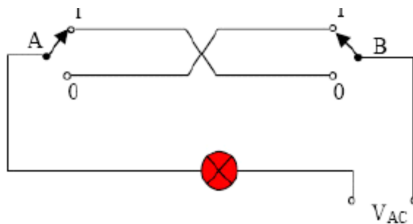
3.2.2 Thiết kế mạch tổ hợp

- Ví dụ: Thiết kế hệ thống mạch logic mô tả quá trình tắt mở đèn tại hai tầng lầu trong nhà.
- Sơ đồ thiết kế

3.2 Phân tích và thiết kế mạch logic tổ hợp

3.2.2 Thiết kế mạch tổ hợp

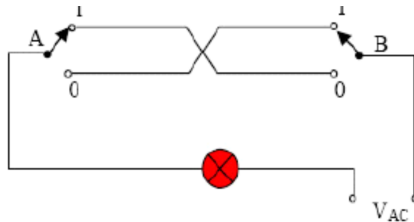
- Ví dụ: Thiết kế hệ thống mạch logic mô tả quá trình tắt mở đèn tại hai tầng lầu trong nhà.
- Sơ đồ thiết kế



3.2 Phân tích và thiết kế mạch logic tổ hợp

3.2.2 Thiết kế mạch tổ hợp

- Ví dụ: Thiết kế hệ thống mạch logic mô tả quá trình tắt mở đèn tại hai tầng lầu trong nhà.
- Sơ đồ thiết kế



- $$f = \overline{A}B + A\overline{B} = A \oplus B$$

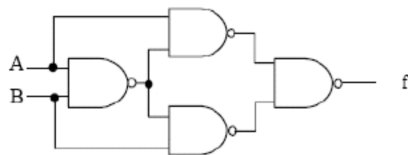
3.2 Phân tích và thiết kế mạch logic tổ hợp

3.2 Phân tích và thiết kế mạch logic tổ hợp

3.2.2 Thiết kế mạch tổ hợp

A	B	f
0	0	0
0	1	1
1	0	1
1	1	0

Bảng 4-1. Bảng trạng thái mô tả hoạt động của hệ chiếu sáng



Hình 4-3. Sơ đồ logic thể hiện hàm f

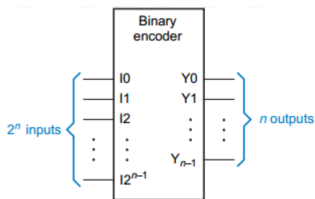
3.4 Mạch mã hóa và giải mã

3.4 Mạch mã hóa và giải mã

- Mã hóa là việc sử dụng ký hiệu để biểu diễn đặc trưng cho một đối tượng nào đó. Trong hệ thống thiết bị điện tử sử dụng mạch số, các lệnh hoặc dữ liệu được truyền đi dưới dạng mã nhị phân gồm có các bit 0 và 1. Như vậy, một số nhị phân 4 bit sẽ biểu diễn được 2^4 lệnh hoặc dữ liệu khác nhau. Một số nhị phân biểu diễn một lệnh hoặc một dữ liệu được gọi là một mã số. Mạch tạo ra các mã số được gọi là mạch mã hóa.

3.4 Mạch mã hóa và giải mã

- Mã hóa là việc sử dụng ký hiệu để biểu diễn đặc trưng cho một đối tượng nào đó. Trong hệ thống thiết bị điện tử sử dụng mạch số, các lệnh hoặc dữ liệu được truyền đi dưới dạng mã nhị phân gồm có các bit 0 và 1. Như vậy, một số nhị phân 4 bit sẽ biểu diễn được 2^4 lệnh hoặc dữ liệu khác nhau. Một số nhị phân biểu diễn một lệnh hoặc một dữ liệu được gọi là một mã số. Mạch tạo ra các mã số được gọi là mạch mã hóa.



3.4 Mạch mã hóa và giải mã

3.4 Mạch mã hóa và giải mã

- Một số loại mạch mã hóa - giải mã như sau:

3.4 Mạch mã hóa và giải mã

- Một số loại mạch mã hóa - giải mã như sau:

3.4.1 Mạch mã hóa thập phân

3.4 Mạch mã hóa và giải mã

- Một số loại mạch mã hóa - giải mã như sau:

3.4.1 Mạch mã hóa thập phân

3.4.2 Mạch mã hóa ưu tiên

3.4 Mạch mã hóa và giải mã

- Một số loại mạch mã hóa - giải mã như sau:

3.4.1 Mạch mã hóa thập phân

3.4.2 Mạch mã hóa ưu tiên

3.4.3 Led 7 đoạn

3.4 Mạch mã hóa và giải mã

- Một số loại mạch mã hóa - giải mã như sau:

3.4.1 Mạch mã hóa thập phân

3.4.2 Mạch mã hóa ưu tiên

3.4.3 Led 7 đoạn

3.4.4 Mạch giải mã nhị phân

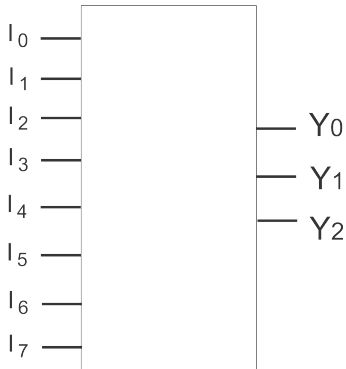
3.4.1 Mạch mã hóa thập phân

3.4.1 Mạch mã hóa thập phân

a) Mã hóa thập phân sang nhị phân (8 đường sang 3 đường)

3.4.1 Mạch mã hóa thập phân

a) Mã hóa thập phân sang nhị phân (8 đường sang 3 đường)



3.4.1 Mạch mã hóa thập phân

3.4.1 Mạch mã hóa thập phân

a) Mã hóa thập phân sang nhị phân (8 đường sang 3 đường)

3.4.1 Mạch mã hóa thập phân

a) Mã hóa thập phân sang nhị phân (8 đường sang 3 đường)

I7	I6	I5	I4	I3	I2	I1	I0	Y2	Y1	Y0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

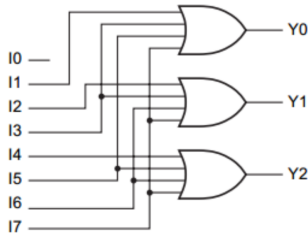
3.4.1 Mạch mã hóa thập phân

3.4.1 Mạch mã hóa thập phân

a) Mã hóa thập phân sang nhị phân (8 đường sang 3 đường)

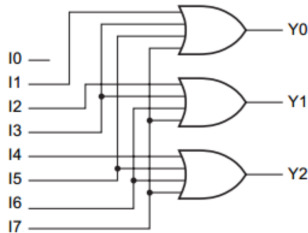
3.4.1 Mạch mã hóa thập phân

a) Mã hóa thập phân sang nhị phân (8 đường sang 3 đường)



3.4.1 Mạch mã hóa thập phân

a) Mã hóa thập phân sang nhị phân (8 đường sang 3 đường)



- Viết biểu thức hàm

3.4.1 Mạch mã hóa thập phân

3.4.1 Mạch mã hóa thập phân

b) Mã hóa thập phân sang nhị phân (10 đường sang 4 đường)

3.4.1 Mạch mã hóa thập phân

b) Mã hóa thập phân sang nhị phân (10 đường sang 4 đường)

- Vẽ sơ đồ ma trận diode

3.4.1 Mạch mã hóa thập phân

b) Mã hóa thập phân sang nhị phân (10 đường sang 4 đường)

- Vẽ sơ đồ ma trận diode
- Viết biểu thức hàm

2.4.2 Mạch mã hóa ưu tiên

2.4.2 Mạch mã hóa ưu tiên

- Mạch mã hóa ưu tiên nếu ấn hai hay nhiều phím cùng một lúc thì mã số tạo ra sẽ ưu tiên cho phím có số lớn.

2.4.2 Mạch mã hóa ưu tiên

- Mạch mã hóa ưu tiên nếu ấn hai hay nhiều phím cùng một lúc thì mã số tạo ra sẽ ưu tiên cho phím có số lớn.
- Ví dụ: mạch mã hóa ưu tiên từ thập phân sang nhị phân.

2.4.2 Mạch mã hóa ưu tiên

- Mạch mã hóa ưu tiên nếu ấn hai hay nhiều phím cùng một lúc thì mã số tạo ra sẽ ưu tiên cho phím có số lớn.
- Ví dụ: mạch mã hóa ưu tiên từ thập phân sang nhị phân.
- Viết biểu thức hàm

2.4.2 Mạch mã hóa ưu tiên

Phím an	I1	I2	I3	I4	I5	I6	I7	I8	I9	O3	O2	O1	O0	Y3	Y2	Y1	Y0
1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1
2	x	0	1	1	1	1	1	1	1	1	1	0	1	0	0	1	0
3	x	x	0	1	1	1	1	1	1	1	0	0	0	0	0	1	1
4	x	x	x	0	1	1	1	1	1	1	0	1	1	0	1	0	0
5	x	x	x	x	0	1	1	1	1	1	0	1	0	0	1	0	1
6	x	x	x	x	x	0	1	1	1	1	0	0	1	0	1	1	0
7	x	x	x	x	x	x	0	1	1	1	0	0	0	0	1	1	1
8	x	x	x	x	x	x	x	0	1	1	1	1	1	1	0	0	0
9	x	x	x	x	x	x	x	x	0	1	1	0	0	1	0	0	1

2.4.3 LED 7 đoạn

2.4.3 LED 7 đoạn

- Dùng để hiển thị chữ số của một hệ đếm bất kỳ.

2.4.3 LED 7 đoạn

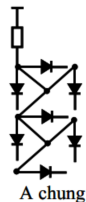
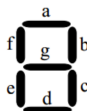
- Dùng để hiển thị chữ số của một hệ đếm bất kỳ.
- Cấu tạo: gồm 7 đoạn làm bằng vật liệu có khả năng phát sáng (LED, LCD, ...)

2.4.3 LED 7 đoạn

- Dùng để hiển thị chữ số của một hệ đếm bất kỳ.
- Cấu tạo: gồm 7 đoạn làm bằng vật liệu có khả năng phát sáng (LED, LCD, ...)
- Có hai loại: anode chung và kathode chung

2.4.3 LED 7 đoạn

- Dùng để hiển thị chữ số của một hệ đếm bất kỳ.
- Cấu tạo: gồm 7 đoạn làm bằng vật liệu có khả năng phát sáng (LED, LCD, ...)
- Có hai loại: anode chung và kathode chung



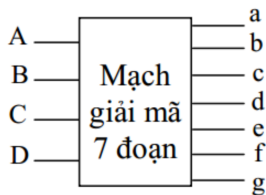
2.4.3 LED 7 đoạn

2.4.3 LED 7 đoạn

- Giải mã LED 7 đoạn

2.4.3 LED 7 đoạn

- Giải mã LED 7 đoạn



	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0							
1	0	0	0	1							
2	0	0	1	0							
3	0	0	1	1							
4	0	1	0	0							
5	0	1	0	1							
6	0	1	1	0							
7	0	1	1	1							
8	1	0	0	0							
9	1	0	0	1							

2.4.4 Mạch giải mã

2.4.4 Mạch giải mã

a) Mạch giải mã led 7 đoạn.

2.4.4 Mạch giải mã

- a) Mạch giải mã led 7 đoạn.
- b) Mạch giải mã địa chỉ.

a) Mạch giải mã led 7 đoạn

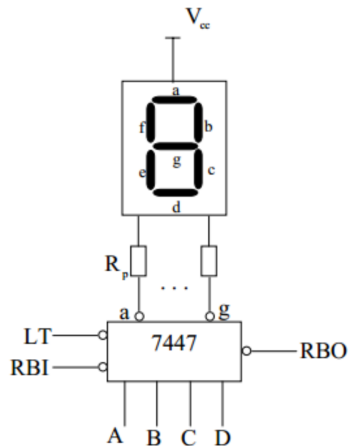
a) Mạch giải mã led 7 đoạn

➤ IC giải mã 7 đoạn:

TTL: A chung: 7447, 74247 (đầu ra ở mức tích cực thấp)

K chung: 7448 (đầu ra ở mức tích cực cao)

CMOS: 4511



a) Mạch giải mã địa chỉ

a) Mạch giải mã địa chỉ

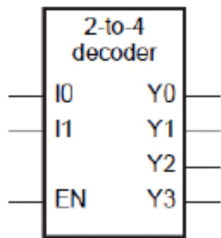
- Là mạch có n ngõ vào và 2^n ngõ ra. Trong đó, mỗi ngõ ra được tác động một lần tương ứng với địa chỉ áp vào của ngõ vào dưới dạng mã nhị phân.

a) Mạch giải mã địa chỉ

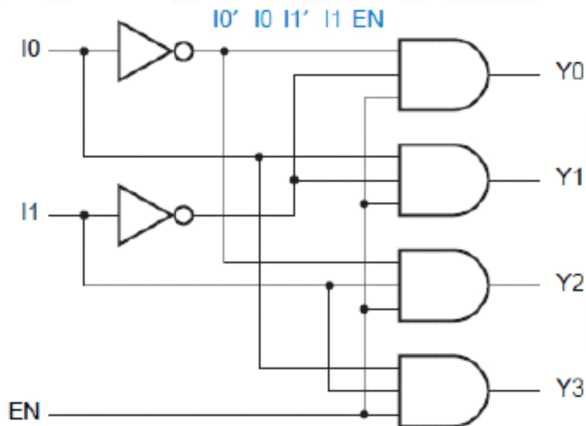
- Là mạch có n ngõ vào và 2^n ngõ ra. Trong đó, mỗi ngõ ra được tác động một lần tương ứng với địa chỉ áp vào của ngõ vào dưới dạng mã nhị phân.
- Giải mã địa chỉ tác động mức thấp, khi ta giải mã đến địa chỉ nào thì ngõ ra tương ứng với địa chỉ đó sẽ ở mức thấp, tất cả các ngõ còn lại sẽ ở mức cao.

Mạch giải mã 2 đường sang 4 đường

Mạch giải mã 2 đường sang 4 đường



(a)



(b)

3.5 Mạch ghép kênh và phân kênh

3.5 Mạch ghép kênh và phân kênh

3.5.1 Mạch ghép kênh (MUX-Multiplexer)

3.5 Mạch ghép kênh và phân kênh

3.5.1 Mạch ghép kênh (MUX-Multiplexer)

3.5.2 Mạch phân kênh (DEMUX-DeMultiplexer)

3.5.2 Mạch phân kênh - DEMUX

3.5.2 Mạch phân kênh - DEMUX

- Định nghĩa: mạch phân kênh là mạch có một ngõ vào dữ liệu, n ngõ vào điều khiển, một ngõ vào chọn mạch và 2^n ngõ ra.

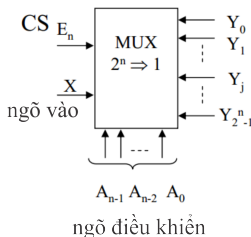
3.5.2 Mạch phân kênh - DEMUX

- Định nghĩa: mạch phân kênh là mạch có một ngõ vào dữ liệu, n ngõ vào điều khiển, một ngõ vào chọn mạch và 2^n ngõ ra.
- Tùy theo giá trị của n ngõ vào điều khiển mà ngõ ra thứ i (Y_i) sẽ bằng giá trị của ngõ vào. Cụ thể nếu gọi n ngõ vào điều khiển là $A_{n-1}A_{n-2}\dots A_0$ thì $Y_i = X$ khi $(A_{n-1}A_{n-2}\dots A_0 = i)_{10}$.

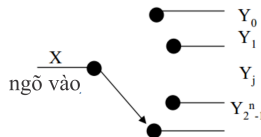
3.5.2 Mạch phân kênh - DEMUX

- Định nghĩa: mạch phân kênh là mạch có một ngõ vào dữ liệu, n ngõ vào điều khiển, một ngõ vào chọn mạch và 2^n ngõ ra.
- Tùy theo giá trị của n ngõ vào điều khiển mà ngõ ra thứ i (Y_i) sẽ bằng giá trị của ngõ vào. Cụ thể nếu gọi n ngõ vào điều khiển là $A_{n-1}A_{n-2}\dots A_0$ thì $Y_i = X$ khi $(A_{n-1}A_{n-2}\dots A_0)_2 = (i)_{10}$.

a) Sơ đồ khối



b) Chuyển mạch điện tử



3.5.2 Mạch phân kênh - DEMUX

3.5.2 Mạch phân kênh - DEMUX

- Bộ phân kênh còn được gọi là bộ giải mã 1 trong 2^n . Tại một thời điểm chỉ có 1 trong số 2^n ngõ ra ở mức tích cực.

3.5.2 Mạch phân kênh - DEMUX

- Bộ phân kênh còn được gọi là bộ giải mã 1 trong 2^n . Tại một thời điểm chỉ có 1 trong số 2^n ngõ ra ở mức tích cực.
- Ví dụ bộ 1:8, có 1 ngõ vào dữ liệu và 8 ngõ ra, IC 74138.

3.6 Mạch số học

3.6 Mạch số học

3.6.1 Mạch cộng

3.6 Mạch số học

3.6.1 Mạch cộng

3.6.2 Mạch trừ

3.6 Mạch số học

3.6.1 Mạch cộng

3.6.2 Mạch trừ

3.6.3 Mạch so sánh

3.6.1 Mạch cộng

3.6.1 Mạch cộng

- Chức năng: thực hiện phép cộng giữa hai số nhị phân.

3.6.1 Mạch cộng

- Chức năng: thực hiện phép cộng giữa hai số nhị phân.
- a) Cộng bán tổng (Half-Adder): thực hiện phép cộng giữa hai bit thấp nhất của phép cộng 2 số nhị phân.

3.6.1 Mạch cộng

- Chức năng: thực hiện phép cộng giữa hai số nhị phân.
- a) Cộng bán tổng (Half-Adder): thực hiện phép cộng giữa hai bit thấp nhất của phép cộng 2 số nhị phân.
- b) Mạch cộng đầy đủ (Full-Adder): thực hiện phép cộng giữa hai bit bất kỳ của phép cộng 2 số nhị phân.

3.6.1 Mạch cộng

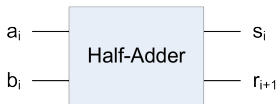
3.6.1 Mạch cộng

- a) Cộng bán tổng (Half-Adder): thực hiện phép cộng giữa hai bit thấp nhất của phép cộng 2 số nhị phân.

3.6.1 Mạch cộng

- a) Cộng bán tổng (Half-Adder): thực hiện phép cộng giữa hai bit thấp nhất của phép cộng 2 số nhị phân.

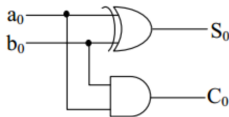
Sơ đồ khối



Bảng trạng thái

a_i	b_i	s_i	r_{i+1}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Sơ đồ mạch



Hàm chức năng

$$s_i = a_i \oplus b_i$$

$$r_{i+1} = a_i \cdot b_i$$

3.6.1 Mạch cộng

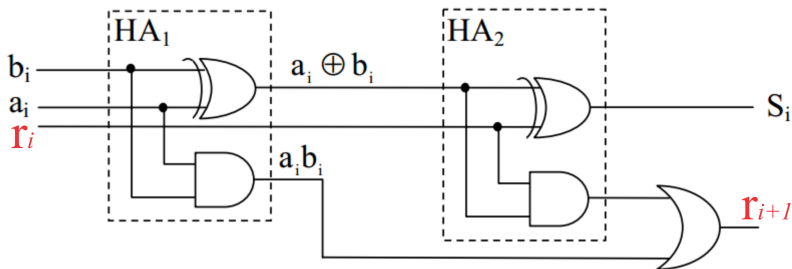
3.6.1 Mạch cộng

- b) Mạch cộng đầy đủ (Full-Adder): thực hiện phép cộng giữa hai bit bất kỳ của phép cộng 2 số nhị phân.

3.6.1 Mạch cộng

- b) Mạch cộng đầy đủ (Full-Adder): thực hiện phép cộng giữa hai bit bất kỳ của phép cộng 2 số nhị phân.

Mạch điện



3.6.1 Mạch cộng

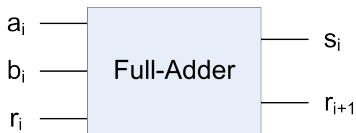
3.6.1 Mạch cộng

- b) Mạch cộng đầy đủ (Full-Adder): thực hiện phép cộng giữa hai bit bất kỳ của phép cộng 2 số nhị phân.

3.6.1 Mạch cộng

- b) Mạch cộng đầy đủ (Full-Adder): thực hiện phép cộng giữa hai bit bất kỳ của phép cộng 2 số nhị phân.

Sơ đồ khối



Bảng trạng thái

a_i	b_i	r_i	s_i	r_{i+1}
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

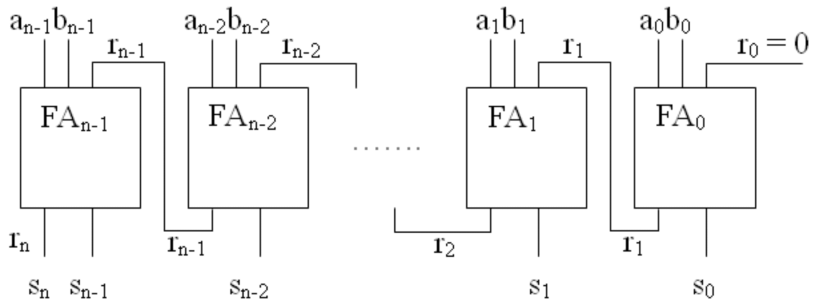
Hàm chức năng

$$s_i = a_i \oplus b_i \oplus r_i$$

$$r_{i+1} = a_i.b_i + r_i(a_i + b_i)$$

3.6.1 Mạch cộng

Mạch điện



3.6.2 Mạch trừ

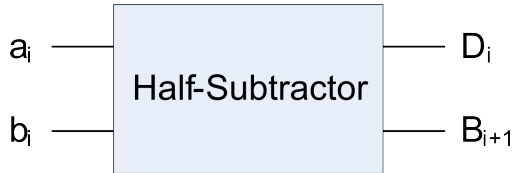
3.6.2 Mạch trừ

- a) Mạch bán hiệu: thực hiện phép trừ giữa 2 bit thấp nhất trong phép trừ giữa hai số nhị phân.

3.6.2 Mạch trừ

a) Mạch bán hiệu: thực hiện phép trừ giữa 2 bit thấp nhất trong phép trừ giữa hai số nhị phân.

- D_i : hiệu
- B_{i+1} bit mượn



3.6.2 Mạch trừ

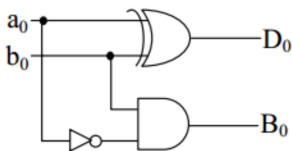
3.6.2 Mạch trừ

Hàm chức năng

$$D_i = a_i \oplus b_i$$

$$B_{i+1} = \overline{a_i} \cdot b_i$$

Hàm logic



Bảng trạng thái

a_i	b_i	D_i	B_{i+1}
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

3.6.3 Mạch so sánh

3.6.3 Mạch so sánh

- Trong các hệ thống số, đặc biệt là trong máy tính thường thực hiện việc so sánh hai số.

3.6.3 Mạch so sánh

- Trong các hệ thống số, đặc biệt là trong máy tính thường thực hiện việc so sánh hai số.
- Hai số cần so sánh có thể là các số nhị phân, có thể là các ký tự đã mã hóa nhị phân.

3.6.3 Mạch so sánh

- Trong các hệ thống số, đặc biệt là trong máy tính thường thực hiện việc so sánh hai số.
- Hai số cần so sánh có thể là các số nhị phân, có thể là các ký tự đã mã hóa nhị phân.
- Mạch so sánh có thể hoạt động theo kiểu nối tiếp hoặc song song.

3.6.3 Mạch so sánh

- Trong các hệ thống số, đặc biệt là trong máy tính thường thực hiện việc so sánh hai số.
- Hai số cần so sánh có thể là các số nhị phân, có thể là các ký tự đã mã hóa nhị phân.
- Mạch so sánh có thể hoạt động theo kiểu nối tiếp hoặc song song.
 - Mạch so sánh bằng nhau 1 bit và mạch so sánh bằng nhau 4 bit.

3.6.3 Mạch so sánh

- Trong các hệ thống số, đặc biệt là trong máy tính thường thực hiện việc so sánh hai số.
- Hai số cần so sánh có thể là các số nhị phân, có thể là các ký tự đã mã hóa nhị phân.
- Mạch so sánh có thể hoạt động theo kiểu nối tiếp hoặc song song.
 - Mạch so sánh bằng nhau 1 bit và mạch so sánh bằng nhau 4 bit.
 - Bộ so sánh 1 bit.

3.6.3 Mạch so sánh

- Trong các hệ thống số, đặc biệt là trong máy tính thường thực hiện việc so sánh hai số.
- Hai số cần so sánh có thể là các số nhị phân, có thể là các ký tự đã mã hóa nhị phân.
- Mạch so sánh có thể hoạt động theo kiểu nối tiếp hoặc song song.
 - Mạch so sánh bằng nhau 1 bit và mạch so sánh bằng nhau 4 bit.
 - Bộ so sánh 1 bit.
 - Bộ so sánh 4 bit (so sánh lớn hơn).

Mạch so sánh bằng nhau 1 bit

Mạch so sánh bằng nhau 1 bit

- Xét 2 bit a_i và b_i , gọi g_i là kết quả so sánh.

Mạch so sánh bằng nhau 1 bit

- Xét 2 bit a_i và b_i , gọi g_i là kết quả so sánh.
- Ta có:

Sơ đồ khối



$$g_i = \overline{a_i} \cdot \overline{b_i} + a_i \cdot b_i = \overline{a_i \oplus b_i}$$

Bảng trạng thái

a_i	b_i	g_i
0	0	1
0	1	0
1	0	0
1	1	1

Mạch so sánh bằng nhau 4 bit

Mạch so sánh bằng nhau 4 bit

- Cho hai số nhị phân 4 bit A và B như sau: $A = a_3a_2a_1a_0$ và $B = b_3b_2b_1b_0$, ta có

Mạch so sánh bằng nhau 4 bit

- Cho hai số nhị phân 4 bit A và B như sau: $A = a_3a_2a_1a_0$ và $B = b_3b_2b_1b_0$, ta có
- $A = B \Leftrightarrow a_3 = b_3, a_2 = b_2, a_1 = b_1, a_0 = b_0$

Mạch so sánh bằng nhau 4 bit

- Cho hai số nhị phân 4 bit A và B như sau: $A = a_3a_2a_1a_0$ và $B = b_3b_2b_1b_0$, ta có
- $A = B \Leftrightarrow a_3 = b_3, a_2 = b_2, a_1 = b_1, a_0 = b_0$
- Biểu thức ngõ ra tương ứng là $G = g_3g_2g_1g_0$, với:

Mạch so sánh bằng nhau 4 bit

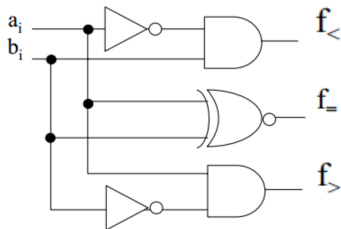
- Cho hai số nhị phân 4 bit A và B như sau: $A = a_3a_2a_1a_0$ và $B = b_3b_2b_1b_0$, ta có
- $A = B \Leftrightarrow a_3 = b_3, a_2 = b_2, a_1 = b_1, a_0 = b_0$
- Biểu thức ngõ ra tương ứng là $G = g_3g_2g_1g_0$, với:
 - $g_3 = \overline{a_3 \oplus b_3}, g_2 = \overline{a_2 \oplus b_2},$

Mạch so sánh bằng nhau 4 bit

- Cho hai số nhị phân 4 bit A và B như sau: $A = a_3a_2a_1a_0$ và $B = b_3b_2b_1b_0$, ta có
- $A = B \Leftrightarrow a_3 = b_3, a_2 = b_2, a_1 = b_1, a_0 = b_0$
- Biểu thức ngõ ra tương ứng là $G = g_3g_2g_1g_0$, với:
 - $g_3 = \overline{a_3 \oplus b_3}, g_2 = \overline{a_2 \oplus b_2},$
 - $g_1 = \overline{a_1 \oplus b_1}, g_0 = \overline{a_0 \oplus b_0}.$

So sánh 1 bit

Sơ đồ khối



Bảng trạng thái

Bảng trạng thái của mạch so sánh				
a_i	b_i	$f_{<}$	$f_{<=}$	$f_{>}$
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

$$f_{<} = \overline{a_i} \cdot b_i$$

Ngõ ra $f_{<=} = \overline{a_i \oplus b_i}$

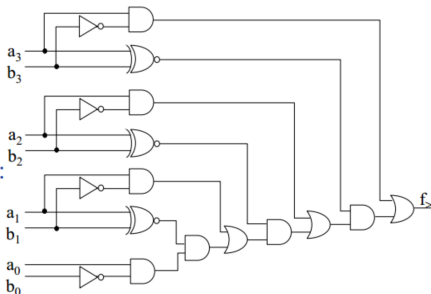
$$f_{>} = a_i \cdot \overline{b_i}$$

So sánh 4 bit

- hoặc $a_3 > b_3$,
- hoặc $a_3 = b_3$, và $a_2 > b_2$,
- hoặc $a_3 = b_3$, và $a_2 = b_2$,
và $a_1 > b_1$,
- hoặc $a_3 = b_3$, và $a_2 = b_2$,
và $a_1 = b_1$, và $a_0 > b_0$.

Từ đó ta có biểu thức hàm ra là:

$$f_{>} = a_3 \cdot \overline{b_3} + \overline{a_3 \oplus b_3} \cdot a_2 \cdot \overline{b_2} + \overline{a_3 \oplus b_3} \cdot \overline{a_2 \oplus b_2} \cdot a_1 \cdot \overline{b_1} + \overline{a_3 \oplus b_3} \cdot \overline{a_2 \oplus b_2} \cdot \overline{a_1 \oplus b_1} \cdot a_0 \cdot \overline{b_0}$$



3.7 Mạch tạo bit và kiểm tra chẵn - lẻ

3.7 Mạch tạo bit và kiểm tra chẵn - lẻ

- Có nhiều phương pháp mã hóa dữ liệu để phát hiện lỗi và sửa lỗi khi truyền dữ liệu từ nơi này sang nơi khác. Phương pháp đơn giản nhất là thêm một bit vào dữ liệu được truyền đi sao cho số chữ số 1 trong dữ liệu luôn là chẵn hoặc luôn là lẻ. Bit thêm vào đó được gọi là bit chẵn/lẻ.

3.7 Mạch tạo bit và kiểm tra chẵn - lẻ

- Có nhiều phương pháp mã hóa dữ liệu để phát hiện lỗi và sửa lỗi khi truyền dữ liệu từ nơi này sang nơi khác. Phương pháp đơn giản nhất là thêm một bit vào dữ liệu được truyền đi sao cho số chữ số 1 trong dữ liệu luôn là chẵn hoặc luôn là lẻ. Bit thêm vào đó được gọi là bit chẵn/lẻ.
- Để thực hiện được việc truyền dữ liệu theo kiểu đưa thêm bit chẵn/lẻ vào dữ liệu, chúng ta phải:

3.7 Mạch tạo bit và kiểm tra chẵn - lẻ

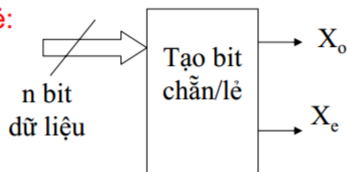
- Có nhiều phương pháp mã hóa dữ liệu để phát hiện lỗi và sửa lỗi khi truyền dữ liệu từ nơi này sang nơi khác. Phương pháp đơn giản nhất là thêm một bit vào dữ liệu được truyền đi sao cho số chữ số 1 trong dữ liệu luôn là chẵn hoặc luôn là lẻ. Bit thêm vào đó được gọi là bit chẵn/lẻ.
- Để thực hiện được việc truyền dữ liệu theo kiểu đưa thêm bit chẵn/lẻ vào dữ liệu, chúng ta phải:
 - Xây dựng sơ đồ tạo được bit chẵn/ lẻ để thêm vào n bit dữ liệu.

3.7 Mạch tạo bit và kiểm tra chẵn - lẻ

- Có nhiều phương pháp mã hóa dữ liệu để phát hiện lỗi và sửa lỗi khi truyền dữ liệu từ nơi này sang nơi khác. Phương pháp đơn giản nhất là thêm một bit vào dữ liệu được truyền đi sao cho số chữ số 1 trong dữ liệu luôn là chẵn hoặc luôn là lẻ. Bit thêm vào đó được gọi là bit chẵn/lẻ.
- Để thực hiện được việc truyền dữ liệu theo kiểu đưa thêm bit chẵn/lẻ vào dữ liệu, chúng ta phải:
 - Xây dựng sơ đồ tạo được bit chẵn/ lẻ để thêm vào n bit dữ liệu.
 - Xây dựng sơ đồ kiểm tra hệ xem đó là hệ chẵn/ lẻ với $(n + 1)$ bit ở ngõ vào (n bit dữ liệu, 1 bit chẵn/ lẻ).

3.7 Mạch tạo bit và kiểm tra chẵn - lẻ

❖ Mạch tạo bit chẵn/lẻ:



Bảng trạng thái của mạch tạo bit chẵn lẻ

Vào			Ra	
d_1	d_2	d_3	X_e	X_o
0	0	0	0	1
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	0

❖ Xét trường hợp 3 bit dữ liệu d_1, d_2, d_3

❖ Gọi X_e, X_o là 2 bit chẵn, lẻ thêm vào dữ liệu.

❖ Từ bảng trạng thái ta thấy

$$X_o = \overline{X_e} \quad \text{hay} \quad X_e = \overline{X_o}$$

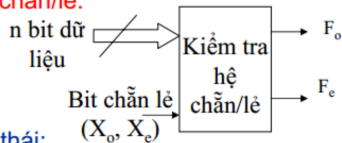
❖ Và biểu thức của X_o và X_e là

$$X_e = d_1 \oplus d_2 \oplus d_3$$

$$X_o = \overline{X_e} = \overline{d_1 \oplus d_2 \oplus d_3}$$

3.7 Mạch tạo bit và kiểm tra chẵn - lẻ

❖ Mạch kiểm tra chẵn/lẻ:



❖ Từ bảng trạng thái:

- $F_e = 1$ nếu hệ là chẵn (F_e chỉ ra tính chẵn của hệ).
- $F_o = 1$ nếu hệ là lẻ (F_o chỉ ra tính lẻ của hệ).

❖ Hai hàm kiểm tra chẵn/lẻ luôn là phủ định của nhau. Do tính chất của hàm cộng XOR, ta có:

- $F_o = d_1 \oplus d_2 \oplus d_3 \oplus X$
- $F_e = \overline{F_o}$

Vào				Ra	
d_1	d_2	d_3	X	F_o	F_e
0	0	0	0	0	1
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	1	0
1	0	0	0	1	0
1	0	0	1	0	1
1	0	1	0	0	1
1	0	1	1	1	0
1	1	0	0	0	1
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	0	1

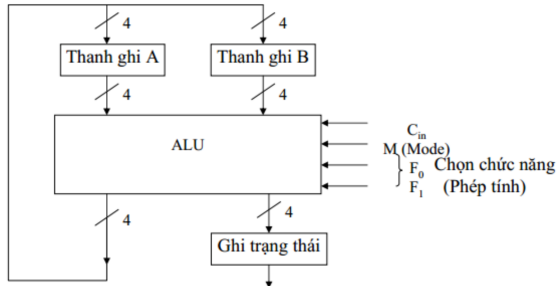
3.7 Mạch tạo bit và kiểm tra chẵn - lẻ

3.7 Mạch tạo bit và kiểm tra chẵn - lẻ

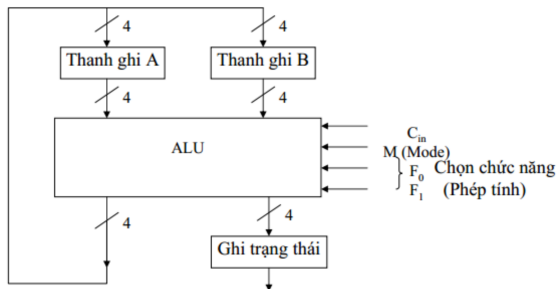
- Ví dụ: Thiết kế mạch tạo bit chẵn, lẻ cho từ mã 3 bit.

3.8 Đơn vị số học và logic (ALU - Arithmetic and Logic Unit)

3.8 Đơn vị số học và logic (ALU - Arithmetic and Logic Unit)



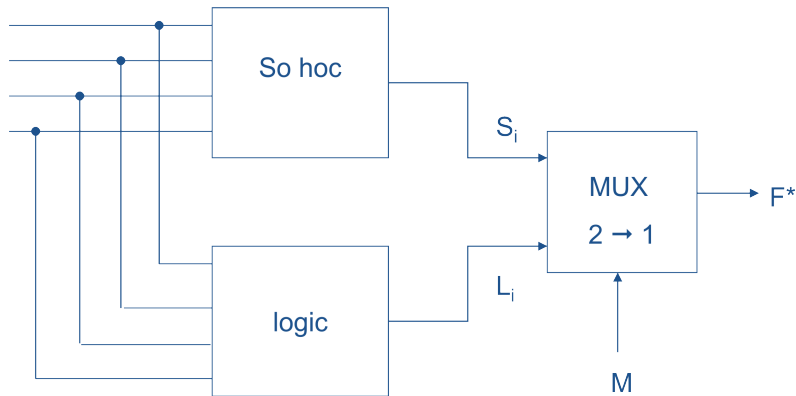
3.8 Đơn vị số học và logic (ALU - Arithmetic and Logic Unit)



- ALU bao gồm hai khối chính là khối logic và khối số học. Ngoài ra còn một khối phụ là khối ghép kênh.

3.8 Đơn vị số học và logic (ALU - Arithmetic and Logic Unit)

3.8 Đơn vị số học và logic (ALU - Arithmetic and Logic Unit)



Hình : Sơ đồ khối cho một cột của ALU.

3.8 Đơn vị số học và logic (ALU - Arithmetic and Logic Unit)

3.8 Đơn vị số học và logic (ALU - Arithmetic and Logic Unit)

- ALU được điều khiển thực hiện phép tính số học hay logic bằng tín hiệu M.

3.8 Đơn vị số học và logic (ALU - Arithmetic and Logic Unit)

- ALU được điều khiển thực hiện phép tính số học hay logic bằng tín hiệu M .
- $M = 0 \rightarrow$ thì ALU thực hiện phép toán logic và đưa kết quả ra ngõ ra f^* .

3.8 Đơn vị số học và logic (ALU - Arithmetic and Logic Unit)

- ALU được điều khiển thực hiện phép tính số học hay logic bằng tín hiệu M .
- $M = 0 \rightarrow$ thì ALU thực hiện phép toán logic và đưa kết quả ra ngõ ra f^* .
- $M = 1 \rightarrow$ thì ALU thực hiện phép toán số học và đưa kết quả ra ngõ ra f^* .

3.8 Đơn vị số học và logic (ALU - Arithmetic and Logic Unit)

- ALU được điều khiển thực hiện phép tính số học hay logic bằng tín hiệu M .
- $M = 0 \rightarrow$ thì ALU thực hiện phép toán logic và đưa kết quả ra ngõ ra f^* .
- $M = 1 \rightarrow$ thì ALU thực hiện phép toán số học và đưa kết quả ra ngõ ra f^* .
- Thực chất tín hiệu M ở đây là tín hiệu điều khiển chọn của MUX $2 \rightarrow 1$.

3.8 Đơn vị số học và logic (ALU - Arithmetic and Logic Unit)

3.8 Đơn vị số học và logic (ALU - Arithmetic and Logic Unit)

- ❖ **Hazzards/Glitch:** Hazard còn được gọi là sự "sai nhầm", hoạt động lúc được lúc không của mạch logic.
 - Xảy ra trong một mạch điện hoàn toàn không bị hỏng linh kiện làm cho mạch hoạt động không có sự tin cậy.
 - Hiện tượng của Hazard trong mạch tổ hợp có thể gặp là:
 - Hazard chỉ xuất hiện một lần và không bao giờ gặp lại nữa.
 - Hazard có thể xuất hiện nhiều lần (theo một chu kỳ nào đó hoặc không theo một chu kỳ nào).
 - Hazard có thể do chính chức năng của mạch điện gây ra. Đây là trường hợp khó giải quyết nhất khi thiết kế.

3.8 Đơn vị số học và logic (ALU - Arithmetic and Logic Unit)

3.8 Đơn vị số học và logic (ALU - Arithmetic and Logic Unit)

❖ Bản chất của hazards

- Do sự chạy đua giữa các tín hiệu
- VD: demo trên Logicworks

❖ Phân loại hazard

- Hazard tĩnh : Đầu ra chỉ xuống 0 hoặc 1 một lần
- Hazard động : Đầu ra có thể thay đổi nhiều hơn 1 lần

Bài tập

Bài tập

Bài tập

1. Xây dựng mạch ghép kênh với bốn kênh tín hiệu ($4 \rightarrow 1$)

Bài tập

1. Xây dựng mạch ghép kênh với bốn kênh tín hiệu ($4 \rightarrow 1$)
2. Thực hiện hàm ba biến f trên cơ sở dùng mạch ghép kênh $4 \rightarrow 1$ đã xây dựng ở bài 1.

Bài tập

1. Xây dựng mạch ghép kênh với bốn kênh tín hiệu ($4 \rightarrow 1$)
2. Thực hiện hàm ba biến f trên cơ sở dùng mạch ghép kênh $4 \rightarrow 1$ đã xây dựng ở bài 1.
3. Xây dựng mạch phân kênh với 4 đường tín hiệu ra.

Bài tập

1. Xây dựng mạch ghép kênh với bốn kênh tín hiệu ($4 \rightarrow 1$)
2. Thực hiện hàm ba biến f trên cơ sở dùng mạch ghép kênh $4 \rightarrow 1$ đã xây dựng ở bài 1.
3. Xây dựng mạch phân kênh với 4 đường tín hiệu ra.
4. Thiết kế mạch kiểm tra chẵn, lẻ cho từ mã 4 bit.