



Kiểm tra 15'

Dùng MUX 4-to-1 và các cổng luận lý cần thiết để hiện thực hàm sau:

$$F(a, b, c, d) = \text{SOP}(1, 3, 5, 6, 8, 11, 15)$$

□ Yêu cầu: ***a*** và ***b*** là các ngõ vào điều khiển của MUX 4-to-1

COMPUTER ENGINEERING



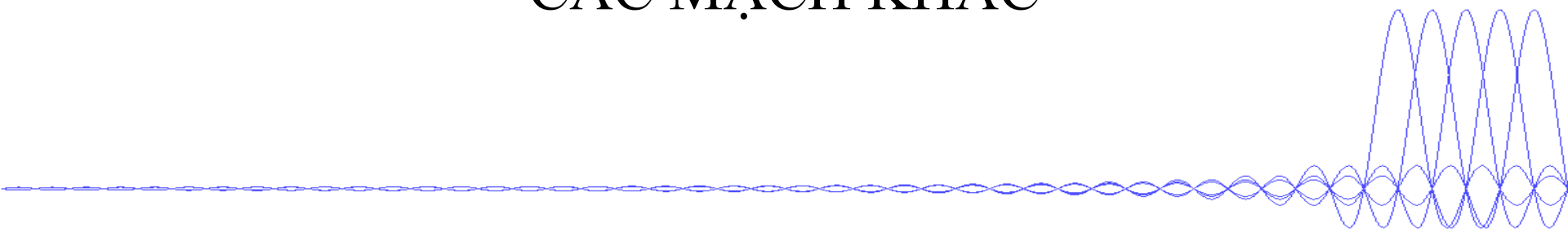
COMPUTER ENGINEERING



UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

NHẬP MÔN MẠCH SỐ

CHƯƠNG 5: MẠCH TỔ HỢP – CÁC MẠCH KHÁC





Nội dung

- Mạch giải mã (Decoder)/ Mạch mã hoá (Encoder)
- Mạch dồn kênh (Multiplexer)/ Mạch chia kênh (Demultiplexer)
- Thiết kế mạch logic sử dụng Mux
- Mạch tạo Parity/ Mạch kiểm tra Parity
- Mạch so sánh (Comparator)

COMPUTER ENGINEERING



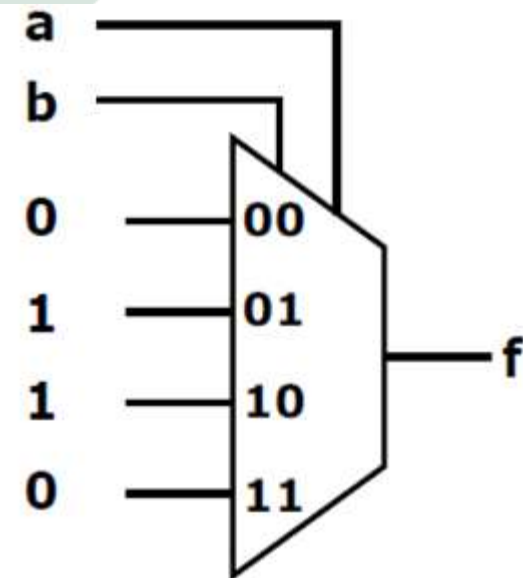
Thiết kế các mạch logic sử dụng MUX

■ Cách hiện thực LUT (Look-up table)

- Sử dụng MUX để chọn một giá trị (**hằng số**) từ 1 LUT

Ví dụ: Thiết kế mạch XOR sử dụng MUX



<i>a</i>	<i>b</i>	<i>f</i>
0	0	0
0	1	1
1	0	1
1	1	0

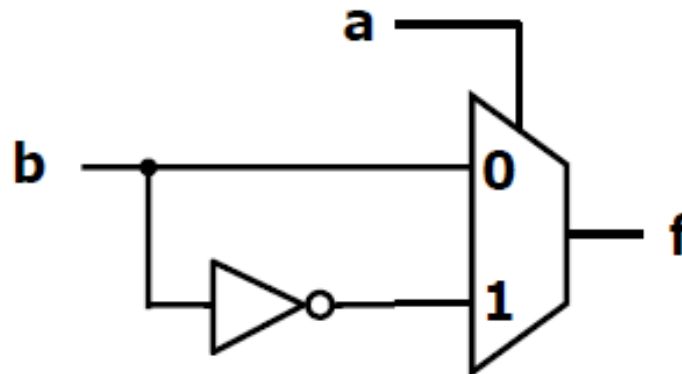




Thiết kế các mạch logic sử dụng MUX

- Giải pháp ở slide trước không hiệu quả vì phải sử dụng MUX 4-to-1
- Nhân xét:

a	b	f				a	f
0	0	0	}	when $a=0, f=b$		0	b
0	1	1				0	b
1	0	1	}	when $a=1, f=b'$		1	b'
1	1	0				1	b'





Thiết kế các mạch logic sử dụng MUX

- Ví dụ: Hiện thực mạch với bảng sự thật sau bằng một **MUX** và các cổng khác

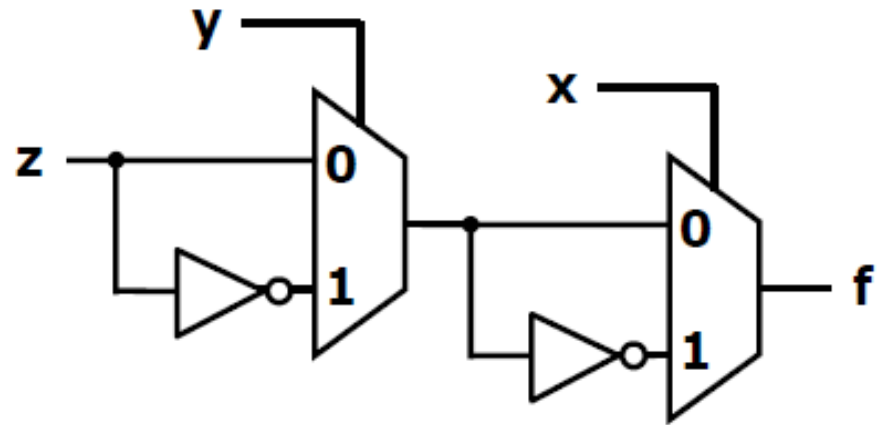
A	B	X
0	0	1
0	1	1
1	0	0
1	1	1



Thiết kế các mạch logic sử dụng MUX

- XOR 3 ngõ vào có thể hiện thực bằng 2 MUX 2-to-1

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





Thiết kế các mạch logic sử dụng MUX

- **Ví dụ:** Hiện thực mạch với bảng sự thật sau bằng một **MUX** và các cổng logic khác

A	B	C	X
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



Thiết kế các mạch logic sử dụng MUX

Biểu thức Shannon

- Bất kì hàm Boolean $f(w_1, w_2, \dots, w_n)$ có thể được viết dưới dạng:

$$f(w_1, w_2, \dots, w_n) = \overline{w_1} * f(0, w_2, \dots, w_n) + w_1 * f(1, w_2, \dots, w_n)$$

COMPUTER ENGINEERING



Thiết kế các mạch logic sử dụng MUX

Biểu thức Shannon

■ Ví dụ 1:

$$f(w_1, w_2, w_3) = w_1 w_2 + w_1 w_3 + w_2 w_3$$

■ Phân tích hàm này theo biến w_1 :

$$f(w_1, w_2, w_3) = \underbrace{w_1(w_2 + w_3)}_{f \text{ khi } w_1=1} + \underbrace{\overline{w_1}(w_2 w_3)}_{f \text{ khi } w_1=0}$$

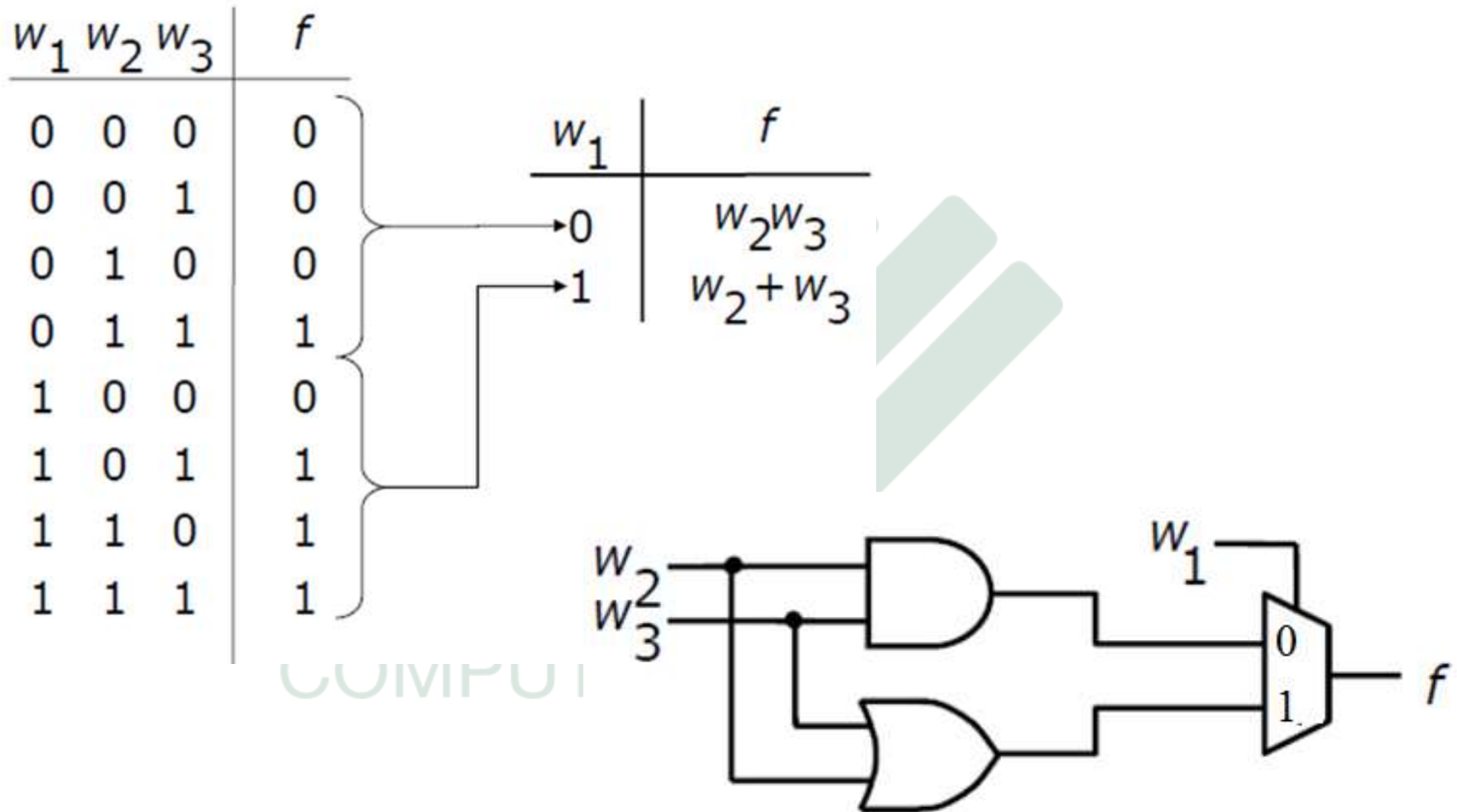
f khi $w_1=1$

f khi $w_1=0$



Thiết kế các mạch logic sử dụng MUX

Biểu thức Shannon





Thiết kế các mạch logic sử dụng MUX

Biểu thức Shannon

■ Ví dụ 2:

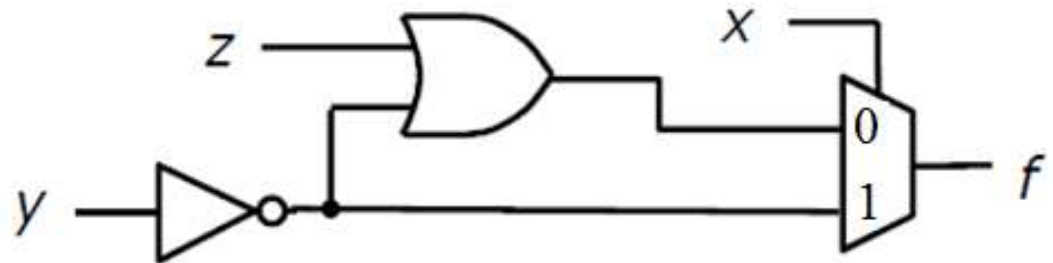
x	y	z	f
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

$$f = x'y'z' + x'y'z + x'yz + xy'z' + xy'z$$

Chọn **x** làm biến mở rộng

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

$$f = x'(y' + z) + x(y')$$





Thiết kế các mạch logic sử dụng MUX

Biểu thức Shannon

■ Ví dụ 3:

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

$$f = x'y'z' + x'y'z + x'yz + xy'z' + xy'z$$

Chọn z làm biến mở rộng

JTER ENGINEERING

Thiết kế các mạch logic sử dụng MUX

Biểu thức Shannon

■ Bài tập 1:

- Dùng MUX 4-to-1 và các cổng luận lý cần thiết để hiện thực hàm sau:

$$F(a, b, c, d) = \text{SOP}(1, 3, 5, 6, 8, 11, 15)$$

- Yêu cầu: ***c*** và ***d*** là các ngõ vào điều khiển của MUX 4-ra-1



Thiết kế các mạch logic sử dụng MUX

Biểu thức Shannon

■ Bài tập 1:

- ❑ Dùng MUX 4-to-1 và các công luận lý cần thiết để hiện thực hàm sau:

$$F(a, b, c, d) = \text{SOP}(1, 3, 5, 6, 8, 11, 15)$$

- ❑ Yêu cầu: **c** và **d** là các ngõ vào điều khiển của MUX 4-ra-1
- ❑ Yêu cầu: **b** và **c** là các ngõ vào điều khiển của MUX 4-ra-1

COMPUTER ENGINEERING



Thiết kế các mạch logic sử dụng MUX

Biểu thức Shannon

■ Bài tập 2:

- Dùng MUX 4-to-1 và các cổng luận lý cần thiết để hiện thực hàm sau:

$$F(a, b, c, d) = \text{SOP}(1, 3, 5, 6, 8, 11, 15)$$

- Yêu cầu: ***b*** và ***c*** là các ngõ vào điều khiển của MUX 4-to-1

COMPUTER ENGINEERING



Nội dung

- Mạch giải mã (Decoder)/ Mạch mã hoá (Encoder)
- Mạch dồn kênh (Multiplexer)/ Mạch chia kênh (Demultiplexer)
- Thiết kế mạch logic sử dụng Mux
- Mạch tạo Parity/ Mạch kiểm tra Parity
- Mạch so sánh (Comparator)

COMPUTER ENGINEERING



Mạch tạo/kiểm tra Parity bit

- Chức năng: Kiểm tra chuỗi bit dữ liệu truyền đúng hay sai tại đầu thu
- Phương pháp:
 - Tại đầu phát: một Parity bit được tạo ra từ chuỗi dữ liệu muốn truyền đi, sau đó Parity bit này được chèn vào cuối chuỗi bit dữ liệu này.
 - Tại đầu thu: Kiểm tra Parity bit để xác nhận chuỗi dữ liệu nhận được có bị sai hay không

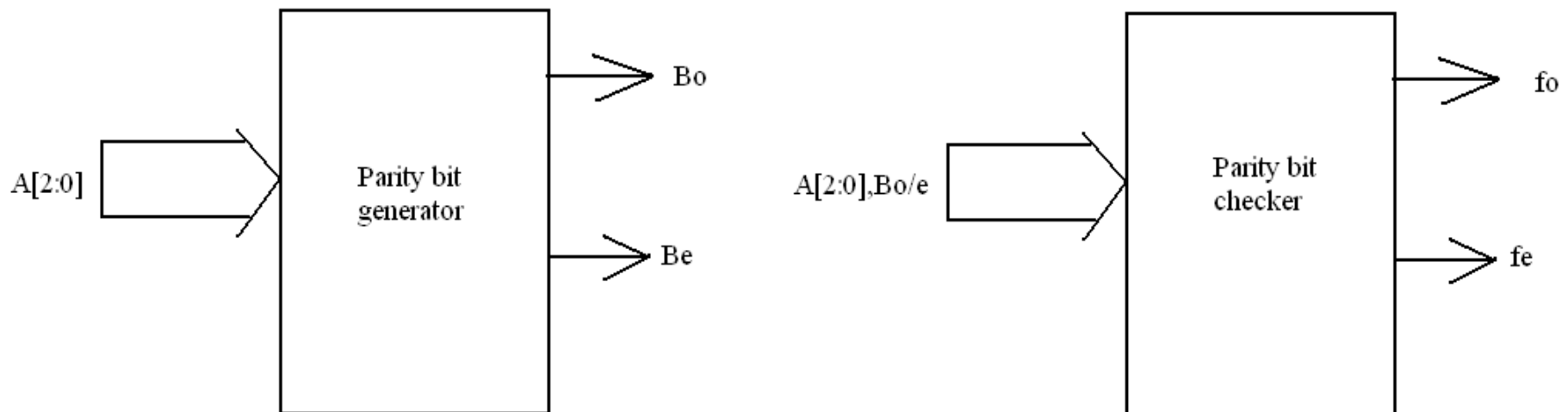
COMPUTER ENGINEERING



Mạch tạo/kiểm tra Parity bit

■ Hai loại Parity bit:

- Bit chẵn (Even parity bit - B_e): $B_e = 1$ khi tổng số bit 1 trong chuỗi bit (kể cả B_e) là số chẵn.
- Bit lẻ (Odd parity bit - B_o): $B_o = 1$ khi tổng số bit 1 trong chuỗi bit (kể cả B_o) là số lẻ



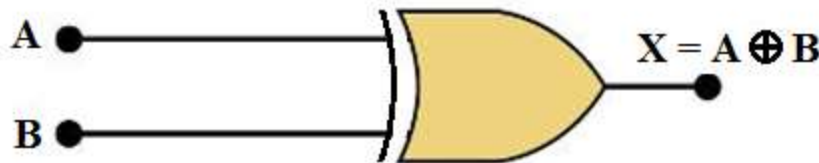


Nhắc lại: Cổng logic XOR, XNOR

■ XOR = Exclusive OR

□ Ngõ ra bằng 1 khi số ngõ vào bằng 1 là lẻ

□ $X = A \oplus B$

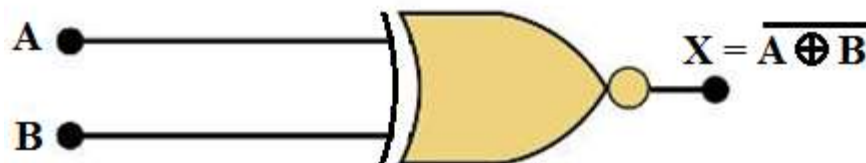


		XOR		XNOR	
		$A \oplus B$		$\overline{A \oplus B}$	
A	B	$A \oplus B$		$A \oplus B$	
0	0	0		1	
0	1	1		0	
1	0	1		0	
1	1	0		1	

■ XNOR = Exclusive NOR

□ Ngõ ra bằng 1 khi số ngõ vào bằng 1 là chẵn

□ $X = \overline{A \oplus B}$





Mạch tạo Parity bit

■ Tạo Even Parity bit

A2	A1	A0	Be
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

■ Tạo Odd Parity bit

A2	A1	A0	Bo
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

➡ $Be = f(A2, A1, A0) ?$ ➡ $Bo = f(A2, A1, A0) ?$



Mạch kiểm tra Even Parity bit

■ Bảng sự thật:

A2	A1	A0	Be	fe	A2	A1	A0	Be	fe
0	0	0	0	0	1	0	0	0	1
0	0	0	1	1	1	0	0	1	0
0	0	1	0	1	1	0	1	0	0
0	0	1	1	0	1	0	1	1	1
0	1	0	0	1	1	1	0	0	0
0	1	0	1	0	1	1	0	1	1
0	1	1	0	0	1	1	1	0	1
0	1	1	1	1	1	1	1	1	0

□ $fe = 1 \rightarrow$
Transmission
failed



$$fe = f(A2, A1, A0, Be) ?$$



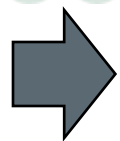
Mạch kiểm tra Odd Parity bit

■ Bảng sự thật:

A2	A1	A0	Bo	fo	A2	A1	A0	Bo	fo
0	0	0	0	1	1	0	0	0	0
0	0	0	1	0	1	0	0	1	1
0	0	1	0	0	1	0	1	0	1
0	0	1	1	1	1	0	1	1	0
0	1	0	0	0	1	1	0	0	1
0	1	0	1	1	1	1	0	1	0
0	1	1	0	1	1	1	1	0	0
0	1	1	1	0	1	1	1	1	1

□ $fo = 1 \rightarrow$

Transmission
failed



$$fo = f(A2, A1, A0, Bo) ?$$



Nội dung

- Mạch giải mã (Decoder)/ Mạch mã hoá (Encoder)
- Mạch dồn kênh (Multiplexer)/ Mạch chia kênh (Demultiplexer)
- Thiết kế mạch logic sử dụng Mux
- Mạch tạo Parity/ Mạch kiểm tra Parity
- Mạch so sánh (Comparator)

COMPUTER ENGINEERING



Mạch so sánh (Comparator)

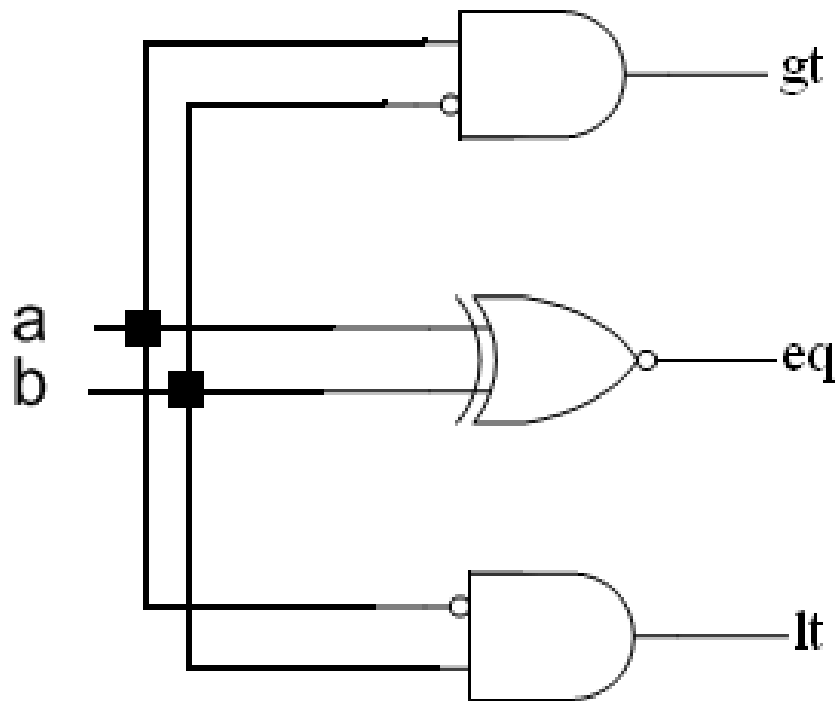
■ Mạch so sánh 2 số

- Xuất ra 1 nếu chúng bằng nhau
 - Xuất ra 0 nếu chúng khác nhau
-
- Dựa trên cổng **XOR**, trả về 0 nếu ngõ vào giống nhau và 1 nếu chúng khác nhau
 - Dựa trên cổng **XNOR**, trả về 1 nếu ngõ vào giống nhau và 0 nếu chúng khác nhau



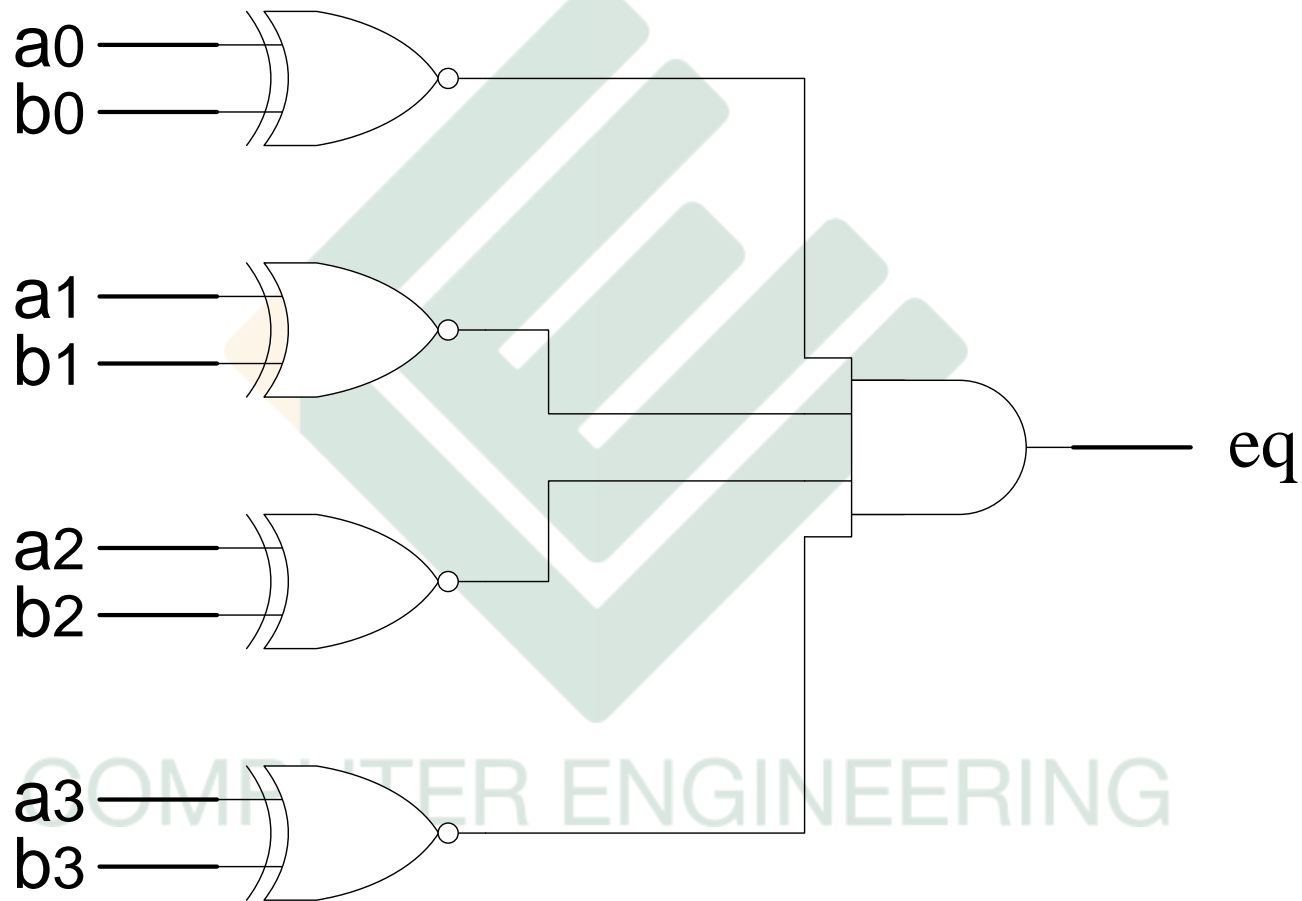
Mạch so sánh 1 bit

a	b	gt	eq	lt
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0



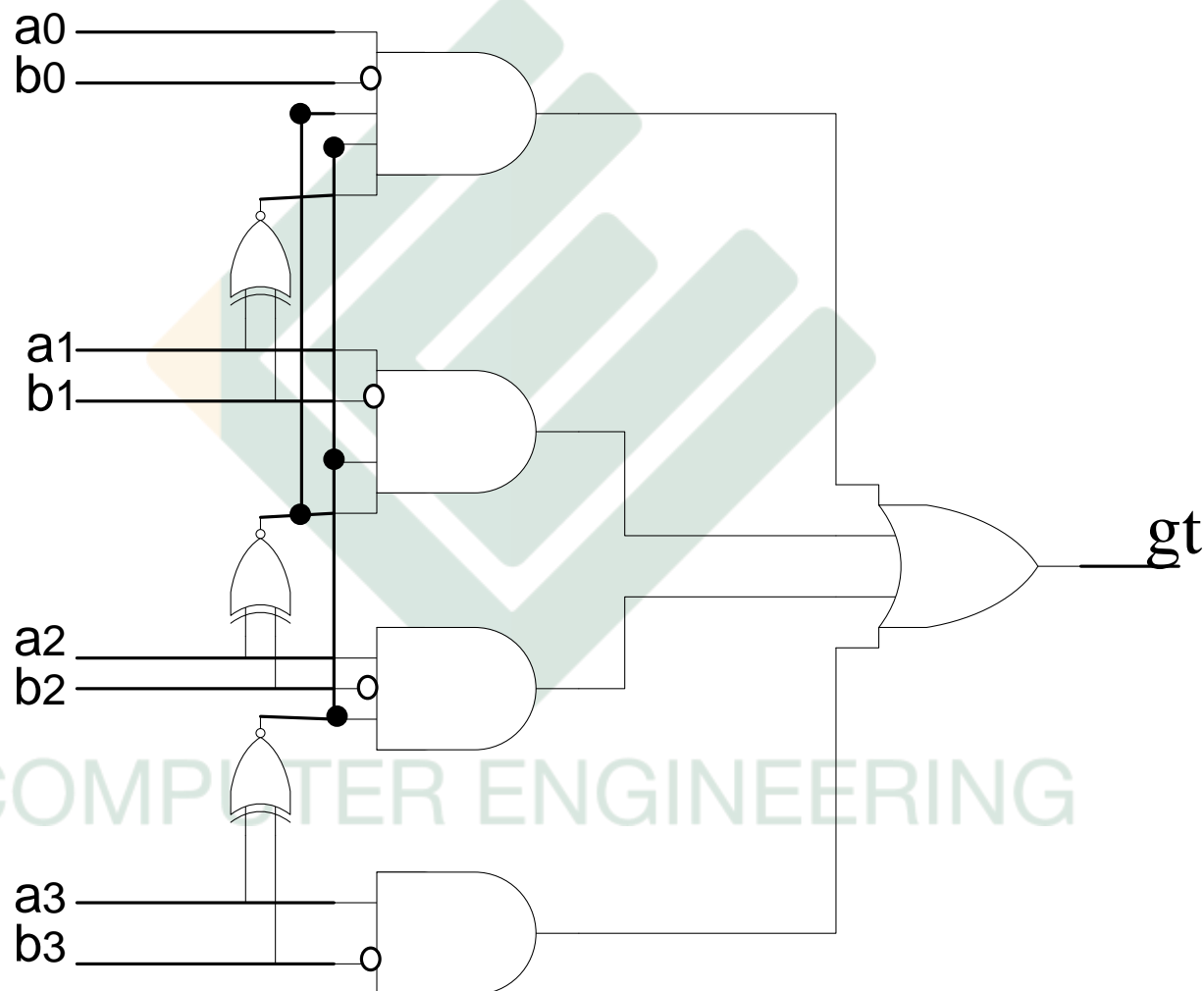


Mạch so sánh 4 bit



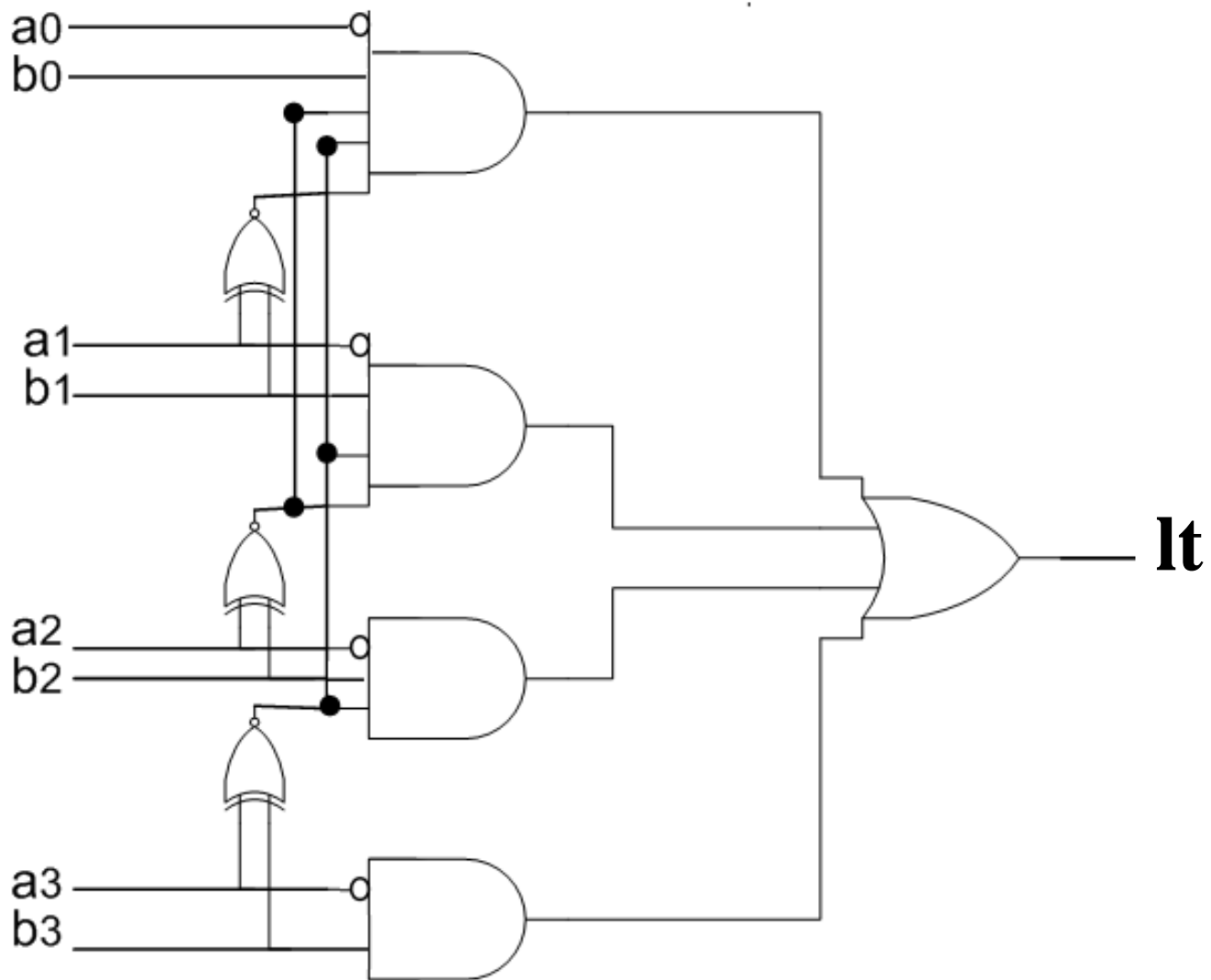


Mạch so sánh 4 bit





Mạch so sánh 4 bit





Mạch so sánh 4-bit

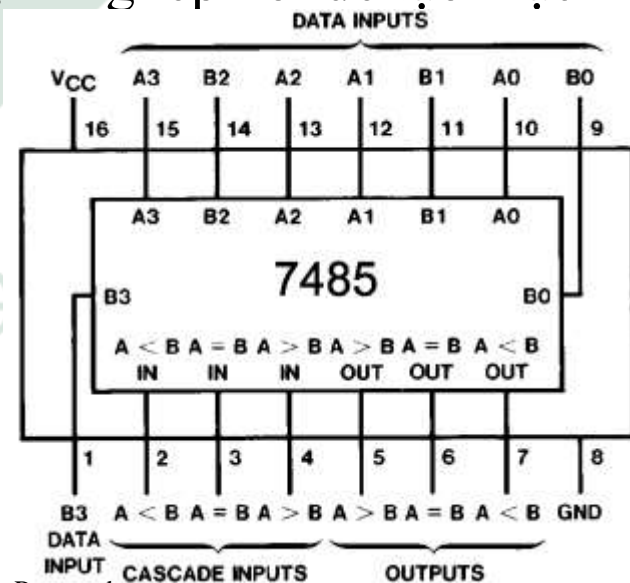
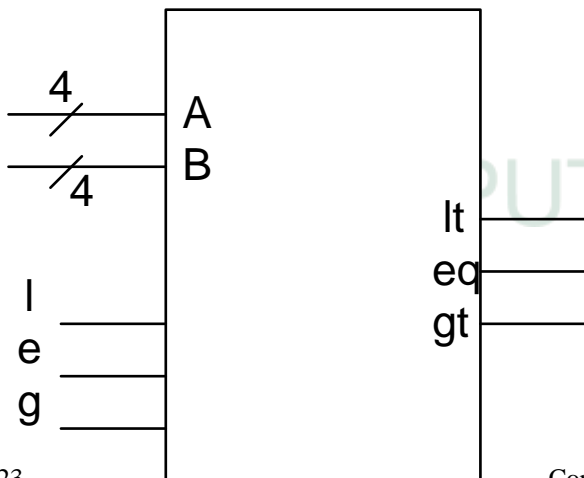
- **74x85** là mạch so sánh tiêu chuẩn với những đặc tính sau:

if ($A > B$) $lt=0$, $eq=0$, $gt=1$

if ($A < B$) $lt=1$, $eq=0$, $gt=0$

if ($A = B$) $lt=$ **l**, $eq=$ **e**, $gt=$ **g**

- **Chú ý:** 3 ngõ vào **l**, **e** và **g** được sử dụng khi ghép nối để tạo mạch so sánh với số bit nhiều hơn

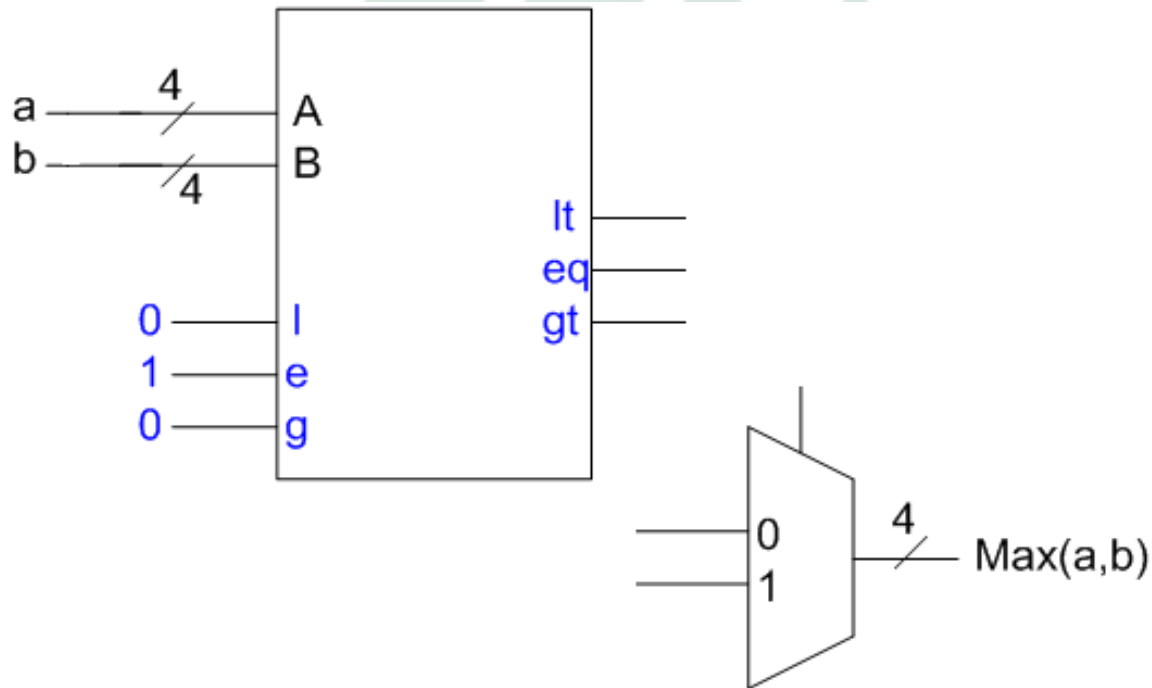




Ví dụ

COMPUTER ENGINEERING

- Thiết kế mạch tìm số lớn nhất, số nhỏ nhất trong 4 số 4-bit sử dụng mạch so sánh và MUXs





Tóm tắt nội dung chương học

- Qua Phần 3 - Chương 5, sinh viên cần nắm những nội dung chính sau:
 - ❑ Một số giải pháp thiết kế mạch số sử dụng mạch chọn kênh, mạch giải mã
 - ❑ Chức năng, ứng dụng và thiết kế của mạch có độ ưu tiên
 - ❑ Chức năng, ứng dụng và thiết kế của mạch tạo và kiểm tra Parity chẵn, lẻ.
 - ❑ Chức năng, ứng dụng và thiết kế của mạch tạo và kiểm tra Parity chẵn, lẻ.
 - ❑ Chức năng, ứng dụng và thiết kế của mạch so sánh



COMPUTER ENGINEERING



UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

Any question?

