

Thiết kế luận lý 1

Khoa KH & KTMT

Bộ môn Kỹ Thuật Máy Tính



Tài liệu tham khảo

- “*Digital Systems, Principles and Applications*”, 11th Edition, Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss

cuu duong than cong . com

cuu duong than cong . com

Chương 4

cuu duong than cong . com

Các phép toán và mạch số học

cuu duong than cong . com



Nội dung

- Các phép toán cộng, trừ, nhân, chia 2 số nhị phân
- Số có dấu và tính toán trên số có dấu sử dụng hệ thống bù-2
- Cộng số BCD
- Cộng, trừ số thập lục phân (hex)
- Các loại mạch cộng/trừ

cuu duong than cong . com

Phép cộng nhị phân

- Phép cộng (**Addition**) là phép toán quan trọng nhất trong các hệ thống số
 - Phép trừ (**Subtraction**), phép nhân (**multiplication**) và phép chia (**division**) được hiện thực bằng cách sử dụng phép cộng
 - Luật cơ bản:
 - $0 + 0 = 0$
 - $1 + 0 = 1$
 - $1 + 1 = 10 = 0 + \text{carry of 1 into next position}$
 - $1 + 1 + 1 = 11 = 1 + \text{carry of 1 into next position}$

– Ví dụ

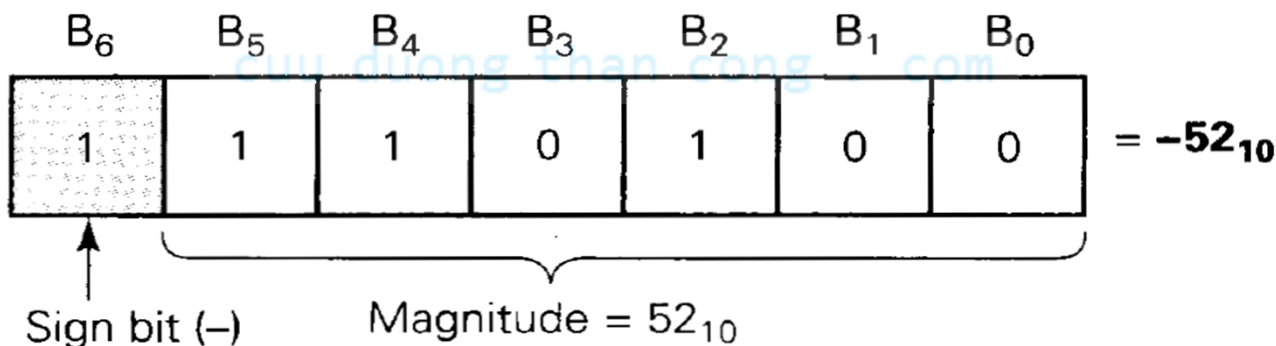
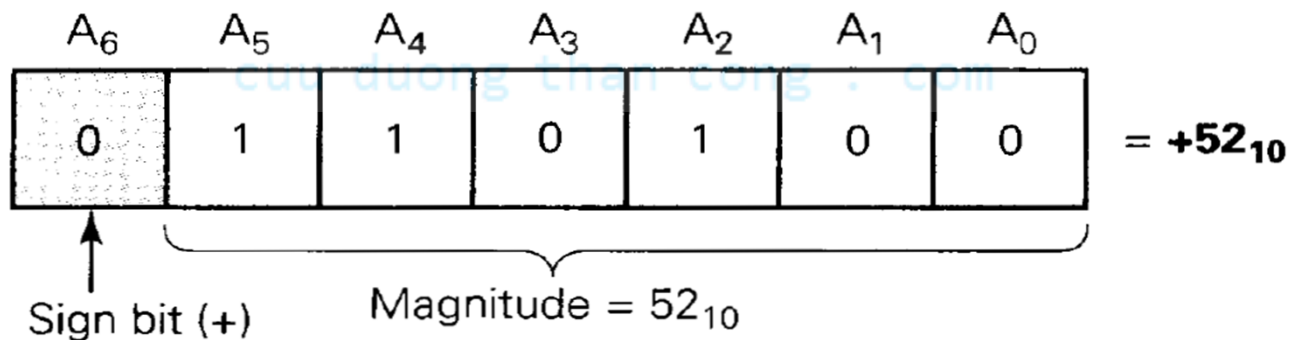
$$\begin{array}{r} 0 \ 1 \ 1 \ (3) \\ + \ 1 \ 1 \ 0 \ (6) \\ \hline 1 \ 0 \ 0 \ 1 \ (9) \end{array}$$

$$\begin{array}{r} 1 \ 0 \ 0 \ 1 \ (9) \\ + \ 1 \ 1 \ 1 \ 1 \ (15) \\ \hline 1 \ 1 \ 0 \ 0 \ 0 \ (24) \end{array}$$

$$\begin{array}{r} 1 \ 1.0 \ 1 \ 1 \ (3.375) \\ + \ 1 \ 0.1 \ 1 \ 0 \ (2.750) \\ \hline 1 \ 1 \ 0.0 \ 0 \ 1 \ (6.125) \end{array}$$

Biểu diễn số có dấu (1)

- Bit dấu (**sign bit**)
0: dương (positive) 1: âm (negative)
- Lượng số (**magnitude**)
- Hệ thống **sign-magnitude**



Biểu diễn số có dấu (2)

- Hệ thống *sign-magnitude* tuy đơn giản nhưng thông thường không được sử dụng do việc hiện thực mạch phức tạp hơn các hệ thống khác
- Dạng bù-1 (**1's-Complement Form**)
 - Chuyển mỗi bit của số nhị phân sang dạng bù
 - Ví dụ: $101101_2 \rightarrow 010010$ (số bù-1)
- Dạng bù-2 (**2's-Complement Form**)
 - Cộng **1** vào vị trí bit **LSB** (trọng số nhỏ nhất) của số **bù-1**
 - Ví dụ: $45_{10} = 101101_2$

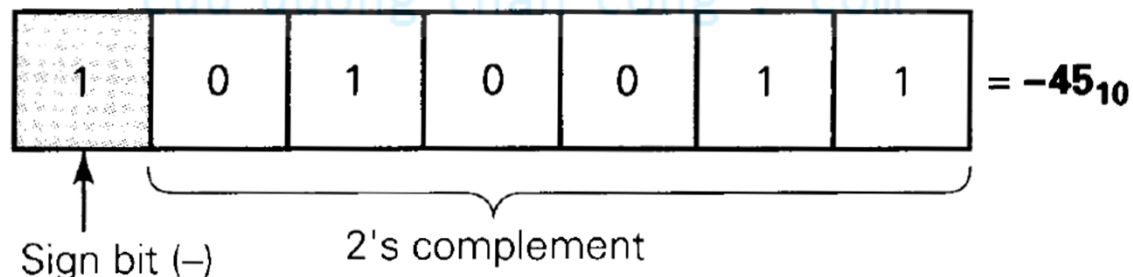
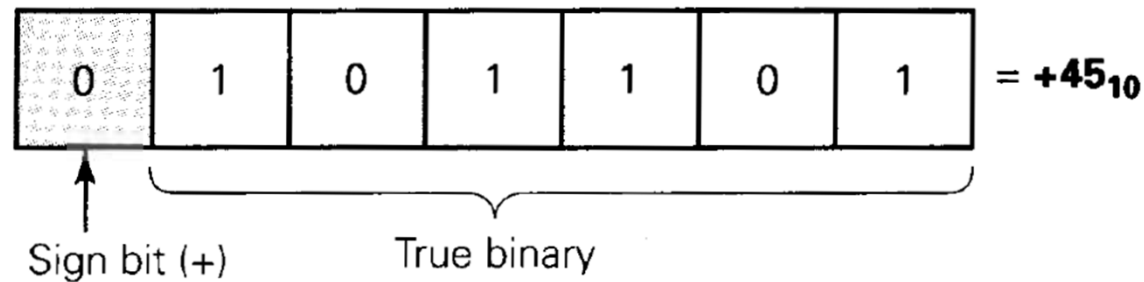
Số bù-1 **010010**

Cộng 1 + **1**

Số bù-2 **010011**

Biểu diễn số có dấu sử dụng bù-2

- Quy tắc
 - **Số dương** (positive): lượng số (magnitude) biểu diễn dưới dạng **số nhị phân đúng**, **bit dấu** bằng **0** (bit trọng số cao nhất - MSB)
 - **Số âm** (negative): lượng số biểu diễn dưới dạng số **bù-2**, **bit dấu** bằng **1** (bit MSB)



Biểu diễn số có dấu sử dụng bù-2

- Hệ thống bù-2 được sử dụng để biểu diễn số có dấu vì nó cho phép thực hiện phép toán trừ bằng cách sử dụng phép toán cộng
 - Các máy tính số sử dụng cùng một mạch điện cho cộng và trừ → tiết kiệm phần cứng
- **Phủ định** (negation): đổi từ số dương sang số âm hoặc từ số âm sang số dương
 - **Phủ định của 1 số nhị phân có dấu là bù-2 của số đó**
 - Ví dụ:

+9	01001	số có dấu
- 9	10111	phủ định (bù-2)
+9	01001	phủ định lần 2 (bù-2)

Trường hợp đặc biệt của bù-2

- Bit dấu bằng 1, N bit lượng số bằng 0: số thập phân tương đương là -2^N
 - Ví dụ:
$$1000 = -2^3 = -8$$
$$10000 = -2^4 = -16$$
$$100000 = -2^5 = -32$$
- Bit dấu bằng 0, N bit lượng số bằng 1: số thập phân tương đương là $+(2^N - 1)$
 - Ví dụ: $0111 = +(2^3 - 1) = +7$
- Khoảng giá trị có thể biểu diễn bằng hệ thống bù-2 với N bit lượng số là
 -2^N đến $+(2^N - 1)$

Phép cộng trong hệ thống bù-2 (1)

- Luật cộng
 - Cộng 2 số **bù-2** theo luật cộng cơ bản (cộng cả bit dấu)
 - Loại bỏ **bit nhớ** (carry) ở **vị trí cuối cùng** của phép cộng (sinh ra bởi phép cộng 2 bit dấu)

Trường hợp 1

bit dấu

+9	→	0	1001
+4	→	0	0100
<hr/>			
+13		0	1101

Trường hợp 2

bit dấu

+9	→	0	1001
-4	→	1	1100
<hr/>			
+5		0	0101

1

↑

carry

Phép cộng trong hệ thống bù-2 (2)

Trường hợp 3

		bit đầu	
-9	→	<div style="border: 1px dashed black; padding: 2px;">1</div>	0111
+4	→	<div style="border: 1px dashed black; padding: 2px;">0</div>	0100
<hr/>			
-5		<div style="border: 1px dashed black; padding: 2px;">1</div>	1011

Trường hợp 4

		bit đầu	
-9	→	<div style="border: 1px dashed black; padding: 2px;">1</div>	0111
-4	→	<div style="border: 1px dashed black; padding: 2px;">1</div>	1100
<hr/>			
-13		<div style="border: 1px dashed black; padding: 2px;">1</div>	0011
		<div style="border: 1px dashed black; padding: 2px;">1</div>	
		carry	

		bit đầu	
-9	→	<div style="border: 1px dashed black; padding: 2px;">1</div>	0111
+9	→	<div style="border: 1px dashed black; padding: 2px;">0</div>	1001
<hr/>			
0		<div style="border: 1px dashed black; padding: 2px;">0</div>	0000
carry		<div style="border: 1px dashed black; padding: 2px;">1</div>	
		bit đầu	

Trường hợp 5

Phép trừ trong hệ thống bù-2

- Phép toán trừ trong hệ thống bù-2 được thực hiện thông qua phép toán cộng
- Trình tự thực hiện
 - Phủ định số trừ
 - Cộng giá trị thu được vào số bị trừ
- Ví dụ

$$\begin{aligned} +9 - 4 &= +9 + (-4) = 01001 + 11100 \\ &= \cancel{1}00101 = +5 \end{aligned}$$

$$\begin{aligned} -9 - 4 &= -9 + (-4) = 10111 + 11100 \\ &= \cancel{1}10011 = -13 \end{aligned}$$

$$\begin{aligned} +9 - 9 &= +9 + (-9) = 01001 + 10111 \\ &= \cancel{1}00000 = 0 \end{aligned}$$

Tràn số học (Arithmetic Overflow)

+9	→	<div style="border: 1px dashed black; padding: 2px; display: inline-block;">0</div>	1001
+8	→	<div style="border: 1px dashed black; padding: 2px; display: inline-block;">0</div>	1000
<hr/>			
+17		<div style="border: 1px dashed black; padding: 2px; display: inline-block;">1</div>	0001

sai bit dấu sai lượng số

cuu duong than cong . com

- Điều kiện tràn: cộng **2 số dương** hoặc **2 số âm**
- Phát hiện tràn
 - Hiện tượng tràn được phát hiện bằng cách kiểm tra bit dấu của kết quả phép cộng so với các bit dấu của các toán hạng
 - **Phép trừ**: tràn chỉ có thể xảy ra khi số trừ và số bị trừ có **bit dấu khác nhau**

Phép toán nhân (multiplication)

- Thao tác nhân 2 số nhị phân được thực hiện theo cách tương tự nhân 2 số thập phân

$$\begin{array}{r} 1001 \\ 1011 \\ \hline 1001 \\ 1001 \\ 0000 \\ 1001 \\ \hline 1100011 \end{array}$$

← Số bị nhân = 9_{10}
← Số nhân = 11_{10}
Tích thành phần
(lần lượt dịch trái)
← Kết quả = 99_{10}

Phép nhân trong hệ thống bù-2

- Nếu số nhân và số bị nhân đều **dương**
 - Nhân bình thường
- Nếu số nhân và số bị nhân là các số **âm**
 - Chuyển 2 số sang số dương sử dụng bù-2
 - Nhân bình thường
 - Kết quả là 1 số dương với **bit dấu bằng 0**
- Nếu **1 trong 2** số là số **âm**
 - Chuyển số âm sang số dương sử dụng bù-2
 - Nhân bình thường
 - Kết quả được **chuyển sang dạng bù-2, bit dấu bằng 1**

Phép toán chia (Division)

- Phép chia 2 số nhị phân được thực hiện theo cách tương tự chia 2 số thập phân

$$\begin{array}{r} 0011 \\ 11 \overline{) 1001} \\ \underline{011} \\ 0011 \\ \underline{11} \\ 0 \end{array}$$

$$9 \div 3 = 3$$

$$\begin{array}{r} 11 \\ \underline{11} \\ 0 \end{array}$$

$$\begin{array}{r} 0010.1 \\ 100 \overline{) 1010.0} \\ \underline{100} \\ 100 \\ \underline{100} \\ 0 \end{array}$$

$$10 \div 4 = 2.5$$

$$\begin{array}{r} 100 \\ \underline{100} \\ 0 \end{array}$$

- Phép chia 2 số có dấu được xử lý theo cách tương tự phép nhân 2 số có dấu

Phép cộng BCD (1)

- Trình tự cộng 2 số BCD
 - Sử dụng phép cộng nhị phân thông thường để cộng các nhóm mã BCD cho từng vị trí ký số BCD
 - Ứng với mỗi vị trí, nếu tổng ≤ 9 , kết quả không cần sửa lỗi
 - Nếu tổng của 2 ký số > 9 , kết quả được cộng thêm 6 (0110) để sửa lỗi, thao tác này **luôn tạo bit nhớ** (carry) cho vị trí ký số kế tiếp

45	0100	0101	← BCD for 45
+ 33	+ 0011	0011	← BCD for 33
78	0111	1000	← BCD for 78

Phép cộng BCD (2)

$$\begin{array}{r}
 0110 \quad \leftarrow \text{BCD for 6} \\
 + 0111 \quad \leftarrow \text{BCD for 7} \\
 \hline
 1101 \quad \leftarrow \text{invalid sum} \\
 + 0110 \quad \leftarrow \text{add 6 for correction} \\
 \hline
 0001 \quad 0011 \quad \leftarrow \text{BCD for 13} \\
 \underbrace{}_1 \quad \underbrace{}_3
 \end{array}$$

$$\begin{array}{r}
 47 \quad 0100 \quad 0111 \quad \leftarrow \text{BCD for 47} \\
 + 35 \quad + 0011 \quad 0101 \quad \leftarrow \text{BCD for 35} \\
 \hline
 82 \quad 0111 \quad 1100 \quad \leftarrow \text{invalid sum in first digit} \\
 \quad \quad 1 \leftarrow \quad 0110 \quad \leftarrow \text{add 6 to correct} \\
 \quad \quad \hline
 \quad \quad 1000 \quad 0010 \quad \leftarrow \text{correct BCD sum} \\
 \quad \quad \underbrace{}_8 \quad \underbrace{}_2
 \end{array}$$

Số học thập lục phân (1)

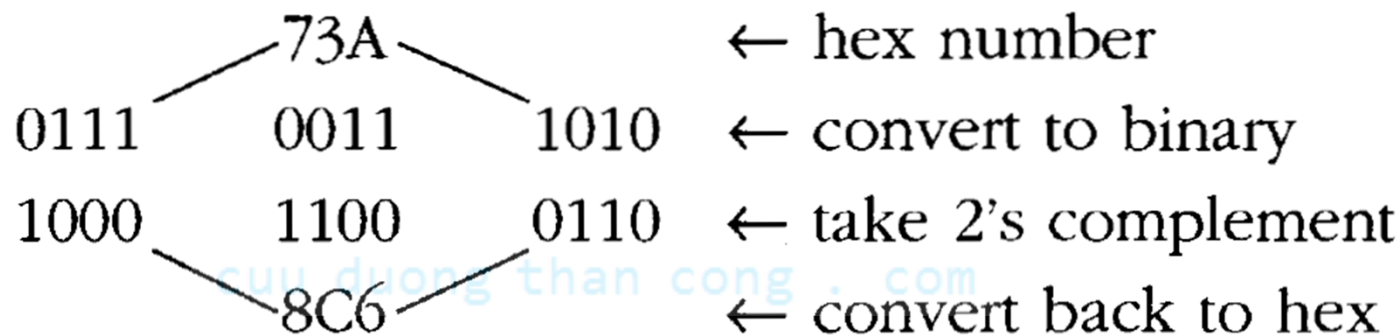
- Phép cộng 2 số thập lục phân được thực hiện theo cách tương tự phép cộng 2 số thập phân
 - Cộng 2 ký số hex dưới dạng thập phân
 - Nếu tổng ≤ 15 , biểu diễn trực tiếp bằng ký số hex
 - Nếu tổng ≥ 16 , trừ cho 16 và nhớ 1 vào vị trí ký số tiếp theo

$$\begin{array}{r} 58 \\ + 24 \\ \hline 7C \end{array} \qquad \begin{array}{r} 3AF \\ + 23C \\ \hline 5EB \end{array}$$

- Phép trừ 2 số thập lục phân
 - Chuyển số trừ sang dạng bù-2 và đem cộng vào số bị trừ
 - Loại bỏ bit nhớ sinh ra do phép cộng 2 ký số ở vị trí cuối cùng (nếu có)

Số học thập lục phân (2)

- Chuyển số hex sang dạng bù-2
 - Số hex \rightarrow số nhị phân \rightarrow dạng bù-2 \rightarrow số hex



- Trừ mỗi ký số hex, lấy kết quả cộng thêm 1

$$\begin{array}{r}
 F \quad F \quad F \\
 -7 \quad -3 \quad -A \\
 \hline
 8 \quad C \quad 5 \\
 \hline
 +1 \\
 \hline
 8 \quad C \quad 6
 \end{array}$$

$$592_{16} - 3A5_{16}$$

$$\begin{array}{r}
 592 \\
 + \quad C5B \\
 \hline
 \end{array}$$

$$11ED$$

\uparrow Disregard carry

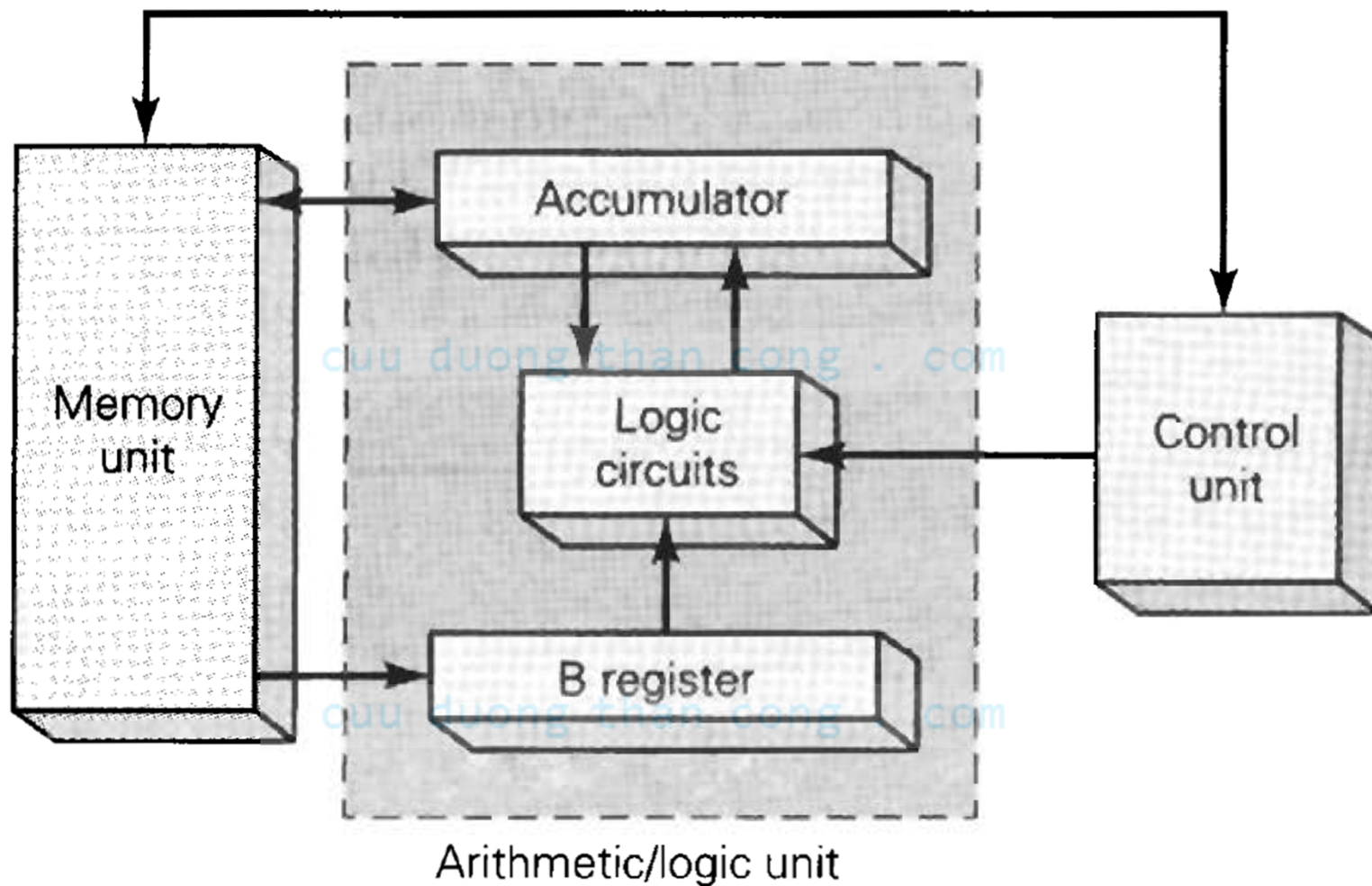
Số học thập lục phân (3)

- Dạng biểu diễn thập lục phân của các số có dấu

Hex Address	Stored Binary Data	Hex Value	Decimal Value
4000	00111010	3A	+58
4001	11100101	E5	-29
4002	01010111	57	+87
4003	10000000	80	-128

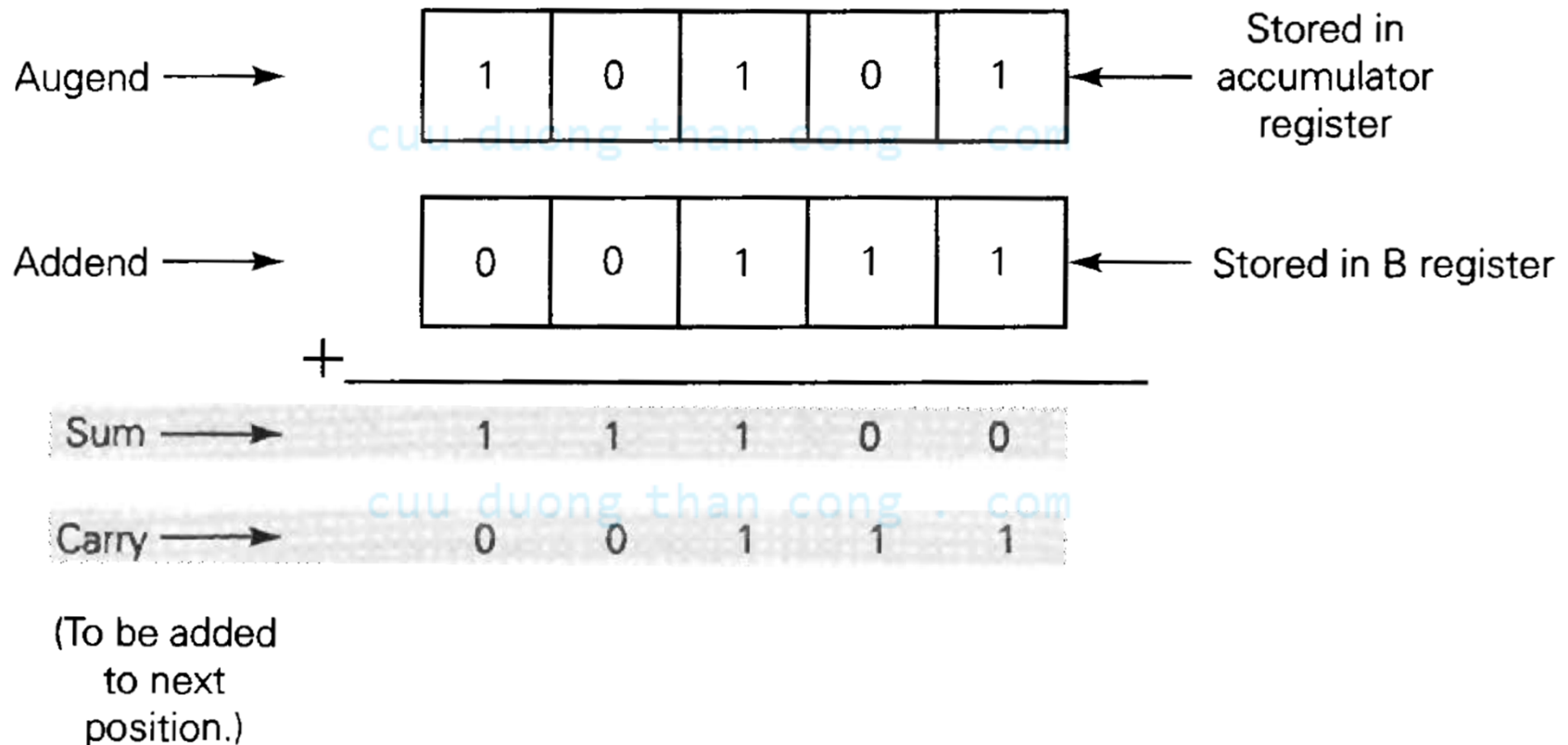
- Số có trọng số cao nhất (**MSD** – most significant digit) ≥ 8 , số được biểu diễn là số âm
- Nếu **MSD** ≤ 7 , số được biểu diễn là số dương

Đơn vị số học và luận lý (ALU)



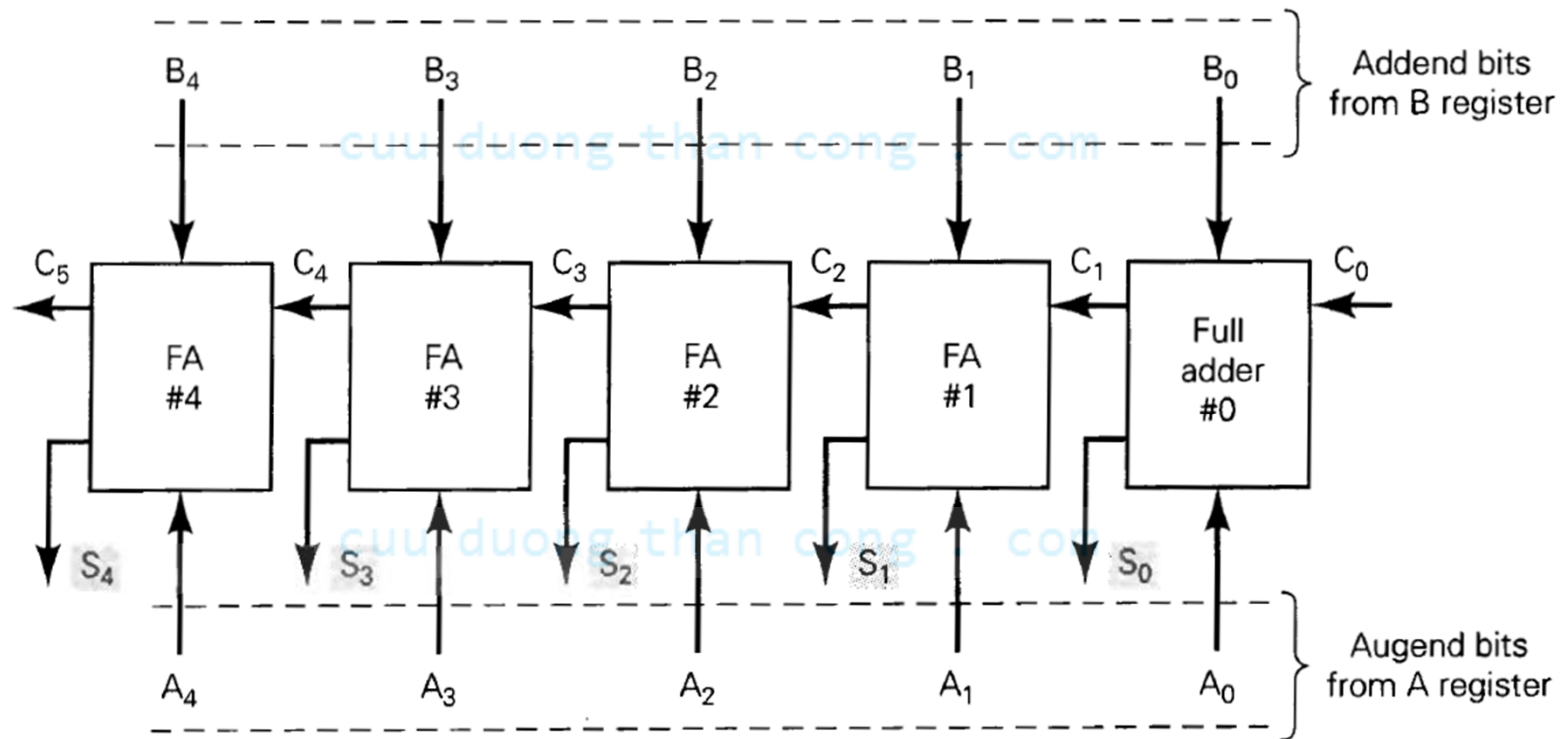
Mạch cộng nhị phân song song

- Toán hạng 1 (số bị cộng): lưu trong thanh ghi tích lũy (accumulator – A)
- Toán hạng 2 (số cộng): lưu trong thanh ghi B



Mạch cộng nhị phân song song

- Toán hạng 1 và 2 được đưa vào mạch cộng toàn phần (**full adder**)
- Thao tác cộng trên mỗi bit được thực hiện đồng thời

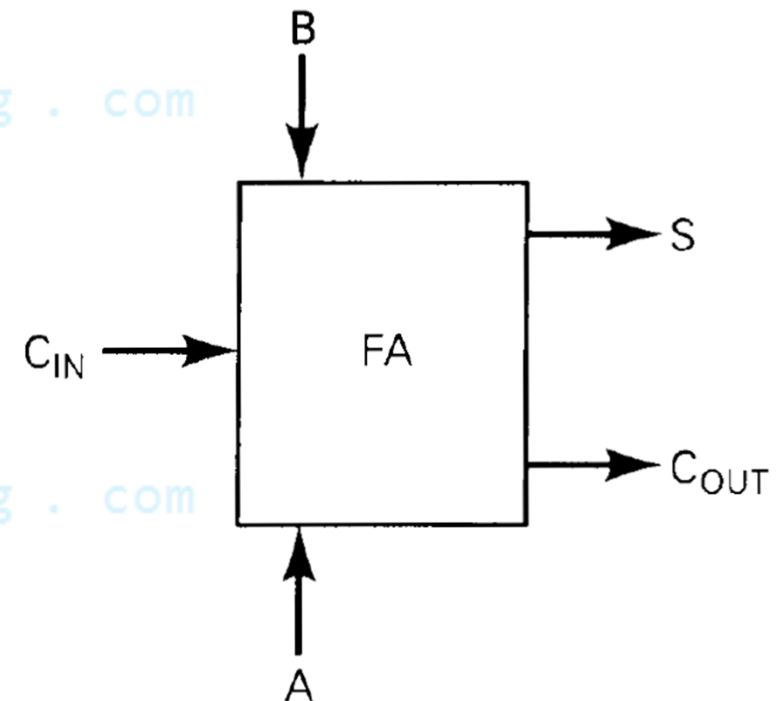


Sum appears at S_4, S_3, S_2, S_1, S_0 outputs.

Thiết kế mạch cộng toàn phần

- Lập bảng sự thật cho mạch cộng toàn phần
- Rút gọn biểu thức mạch cộng toàn phần bằng phương pháp đại số hoặc bìa Karnaugh
- Mạch hoàn chỉnh

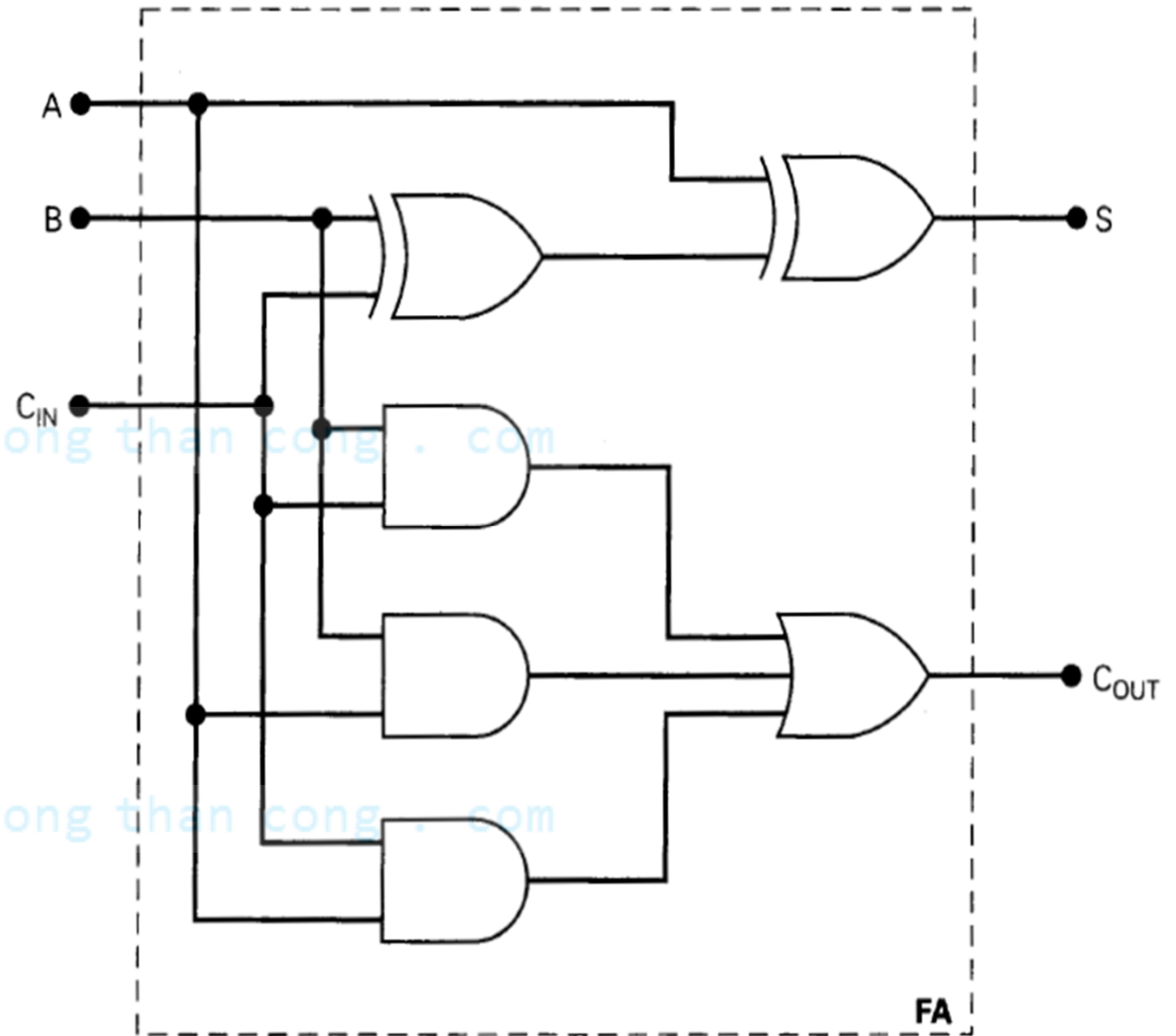
A	B	C_{IN}	S	C_{OUT}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



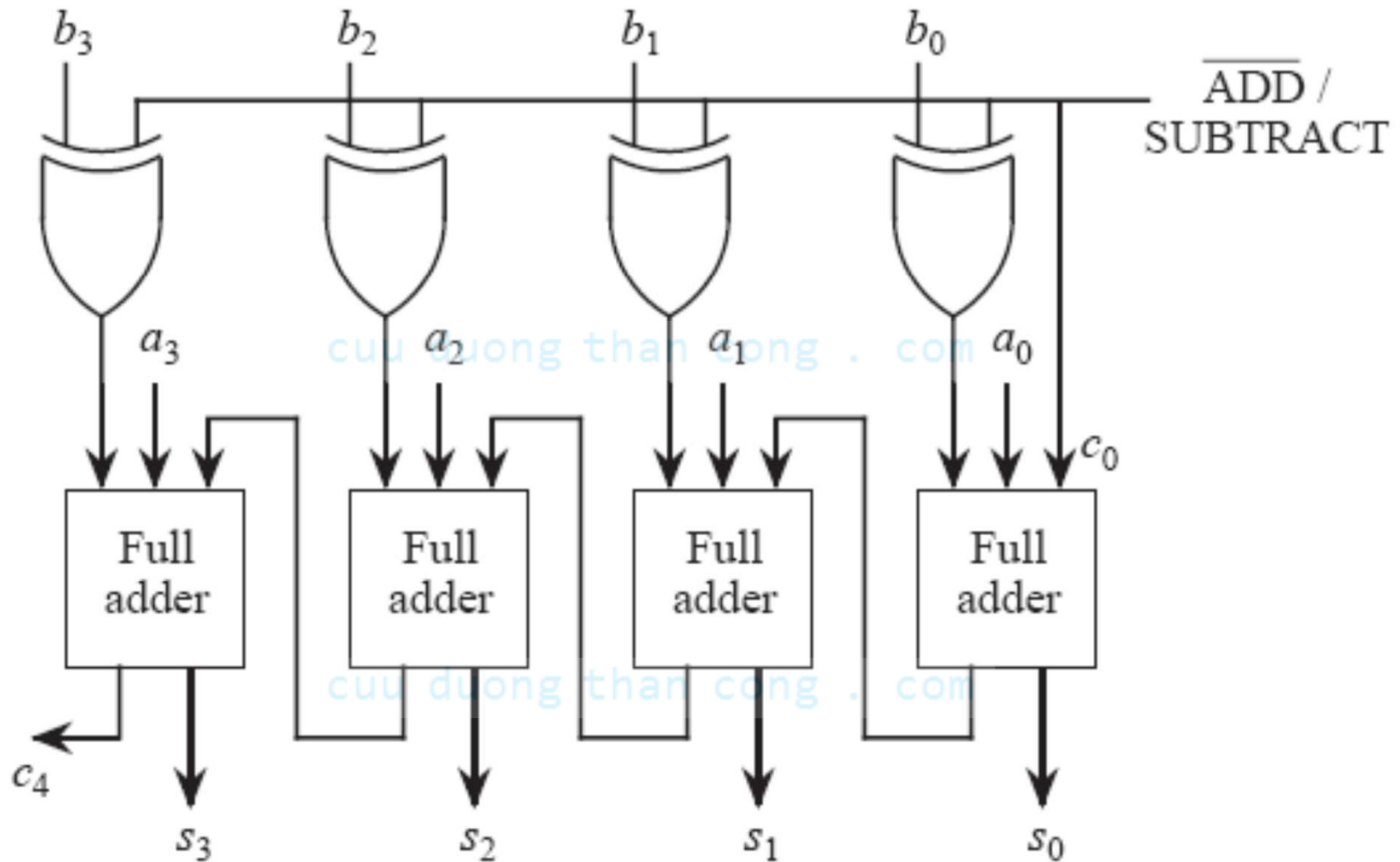
Thiết kế mạch cộng toàn phần

$$S = A \oplus (B \oplus C_{IN})$$

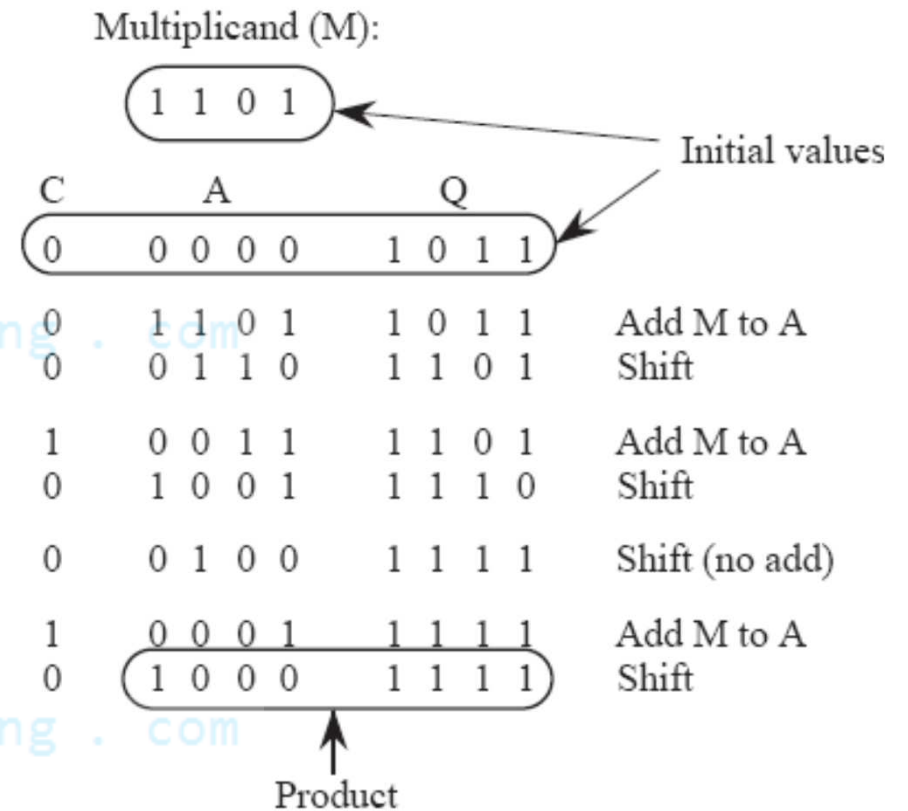
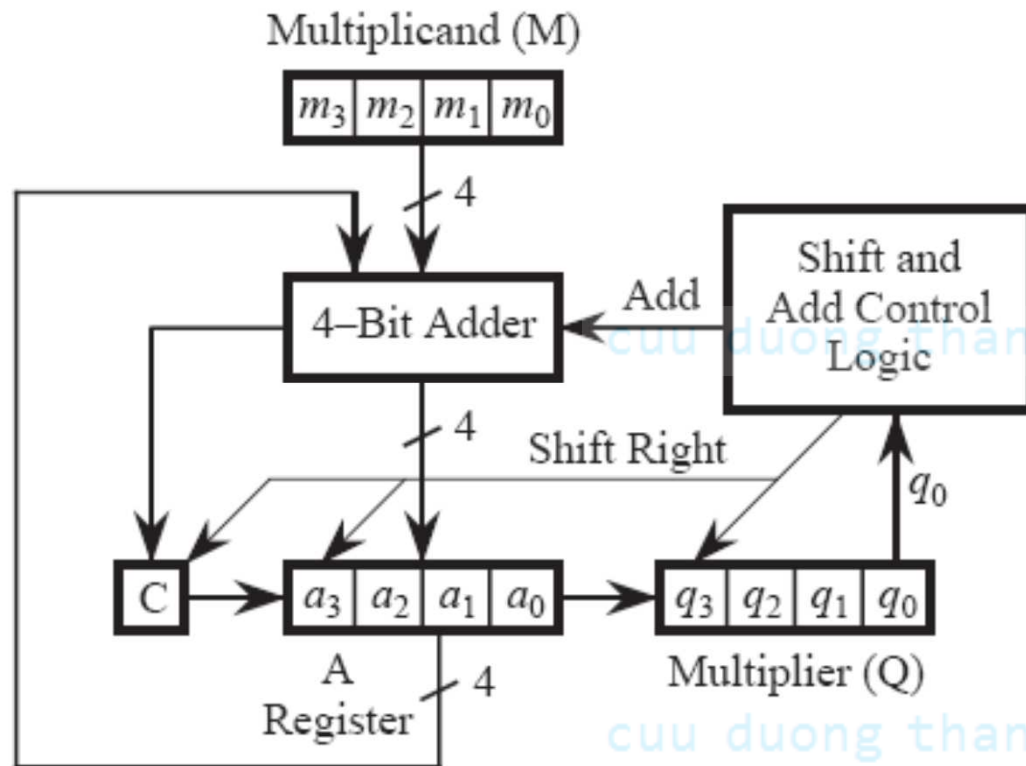
$$C_{OUT} = BC_{IN} + AC_{IN} + AB$$



Mạch cộng/trừ toàn phần



Mạch nhân nhị phân



Mạch chia nhị phân

