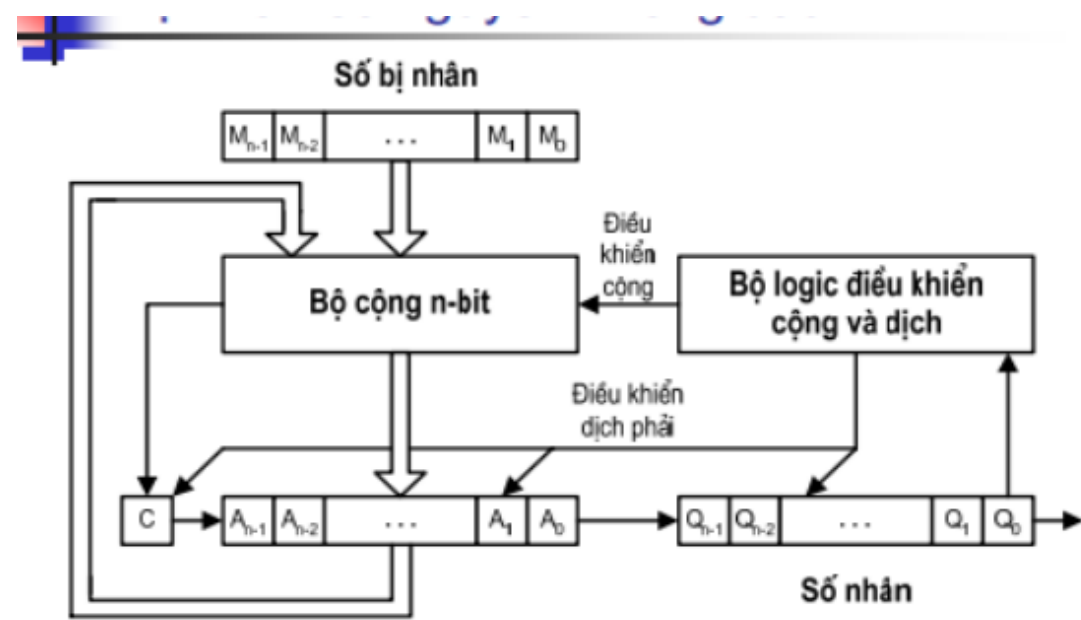
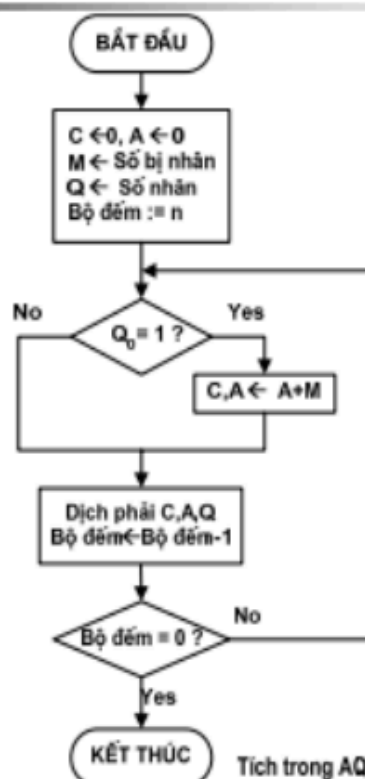


Nhân số nguyên không dấu

- Các **tích riêng phần** được xác định như sau:
 - Nếu bit của số nhân bằng 0 → tích riêng phần bằng 0.
 - Nếu bit của số nhân bằng 1 → tích riêng phần bằng số bị nhân.
 - Tích riêng phần tiếp theo được dịch trái một bit so với tích riêng phần trước đó.
- Tích bằng tổng các **tích riêng phần**
- Nhân hai số nguyên n-bit, tích có độ dài 2n bit (không bao giờ tràn).



Lưu đồ nhân số nguyên không dấu



Ví dụ

- Số bị nhân M = 1011 (11)
- Số nhân Q = 1101 (13)
- Tích = 1000 1111 (143)

	C	A	Q	
■	0	0000	1101	Các giá trị khởi đầu
		+ 1011		
■	0	1011	1101	A ← A + M
■	0	0101	1110	Dịch phải
■	0	0010	1111	Dịch phải
		+ 1011		
■	0	1101	1111	A ← A + M
■	0	0110	1111	Dịch phải
		+ 1011		
■	1	0001	1111	A ← A + M
■	0	1000	1111	Dịch phải

- Số bị nhân M = 0010 (2)
- Số nhân Q = 0011 (3)
- Tích = 0000 0110 (6)

	C	A	Q	
■	0	0000	0011	Các giá trị khởi đầu
		+ 0010		
■	0	0010	0011	A ← A + M
■	0	0001	0001	Dịch phải
		+ 0010		
■	0	0011	0001	
■	0	0001	1000	Dịch phải
■	0	0000	1100	Dịch phải
■	0	0000	0110	Dịch phải

Nhân số nguyên có dấu

- Dùng giải thuật nhân số nguyên không dấu
- + B1: Chuyển đổi số bị nhân và số nhân thành số dương tương ứng
- + B2: Nhân 2 số dương bằng phương pháp nhân số nguyên không dấu, được tích của 2 số dương
- + B3: Hiệu chỉnh dấu của tích: Nếu 2 thừa số ban đầu cùng dấu thì giữ nguyên kết

- Dùng giải thuật Booth

qua. Khác dấu thì đảo dấu kết quả (bằng cách lấy bù 2)

Ví dụ

Ví dụ nhân 10x13 Số bị nhân $10_{10} = 1010_2$ và Số nhân $13_{10} = 1101_2$

Bước	Bộ đếm	Số nhân	Nhớ	Tích thành phần	Vòng lặp	Ghi chú
(a) và (b)	4	1101	0	0000 0000		Khởi trị, LSB=1
(c)	4	1101	0	1010 0000	1	Cộng
(d)	4	0110	0	0101 0000	1	Dịch phải
(e)	3	0110	0	0101 0000	1	Giảm n
(c)	3	0110	0	0101 0000	2	LSB=0
(d)	3	0011	0	0010 1000	2	Dịch phải
(e)	2	0011	0	0010 1000	2	Giảm n, LSB=1
(c)	2	0011	0	1100 1000	3	Cộng
(d)	2	0001	0	0110 0100	3	Dịch phải
(e)	1	0001	0	0110 0100	3	Giảm n, LSB=1
(c)	1	0001	1	0000 0100	4	Cộng
(d)	1	0000	0	1000 0010	4	Dịch phải
(e)	0	0000	0	1000 0010	4	Giảm n. Dừng

Giải thuật Booth

Lưu ý: Trong các trường hợp, sự dịch phải các bit trái nhất của A - bit A_{n-1} không chỉ dịch vào A_{n-2} nhưng cũng còn lại trong A_{n-1} , điều này được đòi hỏi để bảo tồn dấu của số trong A và Q

Example Consider the multiplication of the two positive numbers $M = 0111$ (7) and $Q = 0011$ (3) and assuming that $n = 4$. The steps needed are tabulated below.

M	A	Q	Q(-1)			
0111	0000	0011	0	Initial value		c=4
0111	1001	0011	0	$A = A - M$		
0111	1100	1001	1	ASR	End cycle #1	c=3

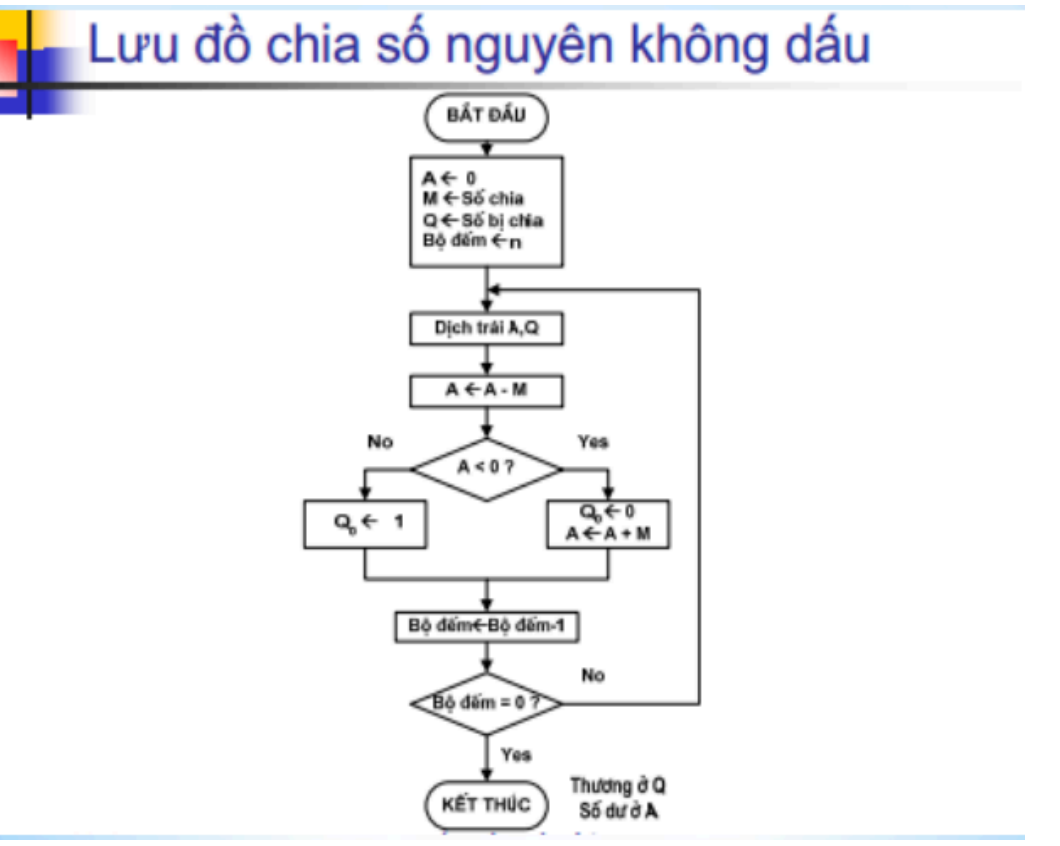
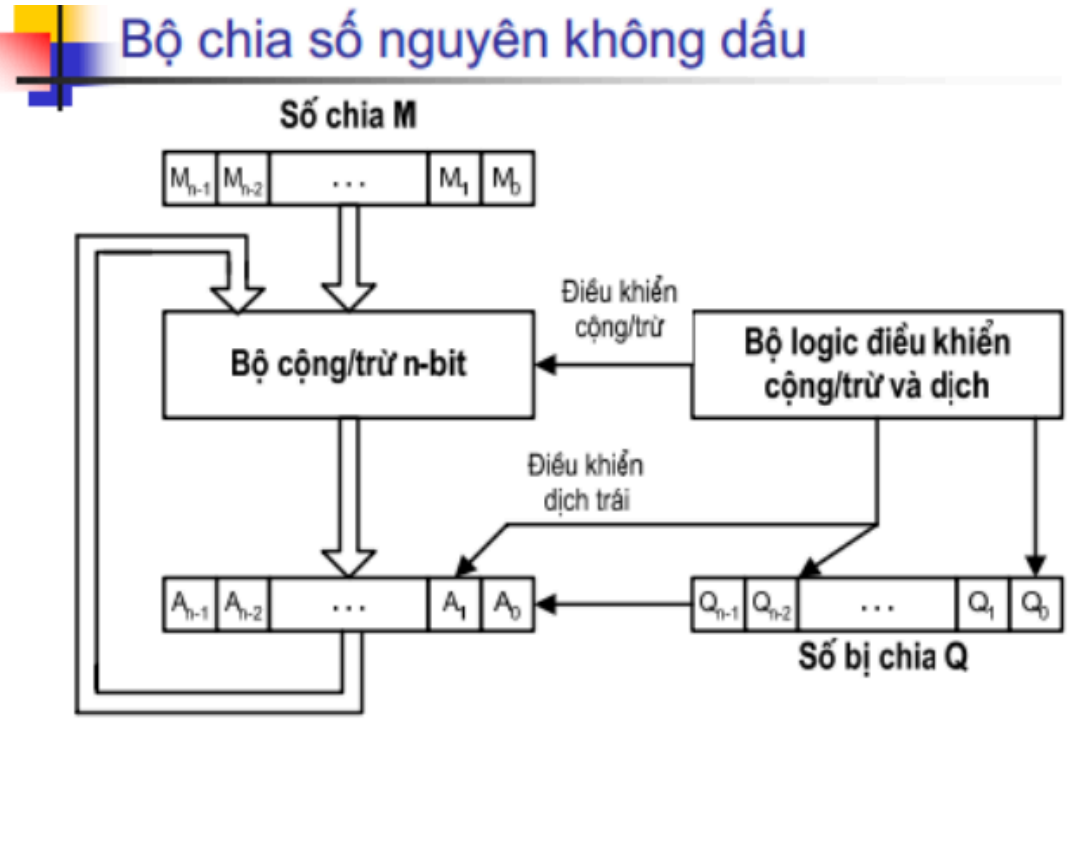
0111	1100	0100	1	ASR	End cycle #2	c=2

0111	0101	0100	1	$A = A + M$		
0111	0010	1010	0	ASR	End cycle #3	c=1

0111	0001	0101	1	ASR	End cycle #4	c=0

				+21 (correct result)		

Chia số nguyên không dấu



Ví dụ

* Ví dụ phép chia 7/3 (0111₂/0011₂)

A	Q	M = 0011
0000	0111	Khởi trị. c=4
0000	1110	Dịch trái A, Q (vòng 1)
1101		Trừ (A←A-M), không thành công do (A<0)
0000	1110	A=0, Q ₀ ←0, khôi phục A (A←A+M), c=3
0001	1100	Dịch trái (vòng 2)
1110		Trừ (A←A-M); không thành công (A<0)
0001	1100	Khôi phục A, c=2
0011	1000	Dịch trái (vòng 3)
0000		Trừ, thành công (A≥0)
0000	1001	Đặt Q ₀ ←1, c=1
0001	0010	Dịch trái (vòng 4)
1110		Trừ
0001	0010	Khôi phục A, c=0. Dừng.

Vậy kết quả thương Q = 0010₂=2₁₀ và dư A = 0001₂=1₁₀

Chia số nguyên có dấu

- Bước 1. Chuyển đổi số bị chia và số chia về thành số dương tương ứng.
- Bước 2. Sử dụng thuật giải chia số nguyên không dấu để chia hai số dương, kết quả nhận được là thương Q và phần dư R đều là dương
- Bước 3. Hiệu chỉnh dấu của kết quả như sau:
(Lưu ý: phép đảo dấu thực chất là thực hiện phép lấy bù hai)

Số bị chia	Số chia	Thương	Số dư
dương	dương	giữ nguyên	giữ nguyên
dương	âm	đảo dấu	giữ nguyên
âm	dương	đảo dấu	đảo dấu
âm	âm	giữ nguyên	đảo dấu

Thuật toán chia 2 số nguyên tổng quát

- Nạp số chia vào thanh ghi M và số bị chia vào các thanh ghi A, Q. Số bị chia phải ở dạng 2n bit, vì vậy, ví dụ số 4 bit 0111 trở thành 00000111 và 1001 trở thành 11111001. Count=n
- Dịch trái A, Q đi 1 bit
- Nếu M và A có cùng dấu, thực hiện gán A←A-M, ngược lại gán A ← A+M
- Thao tác trên thành công nếu dấu của A là như nhau trước và sau thao tác
 - Nếu thao tác thành công hoặc (A=0 hoặc Q=0) thì đặt Q₀← 1
 - Nếu thao tác không thành công và (A≠0 hoặc Q ≠ 0) thì đặt Q₀← 0 và khôi phục giá trị trước đó của A. Count ← Count-1
- Lặp lại bước 2 đến bước 4 số lần lặp bằng số bit của Q (Count=0)
- Phần dư là A. Nếu dấu của số chia và số bị chia giống nhau thì thương là Q, ngược lại thương là số bù 2 của Q.