

UNIDADES FUNCIONALES DEL COMPUTADOR

Tema 5

Principios de Computadoras

Temas

Introducción

1. Bus
 2. Memoria
 3. Entrada/Salida
 4. Unidad aritmética-lógica
 5. Unidad de Control
-

4. UNIDAD ARITMÉTICO- LÓGICA (ALU)

Tema 5

Principios de Computadoras

4. Unidad Aritmético-Lógica (ALU)

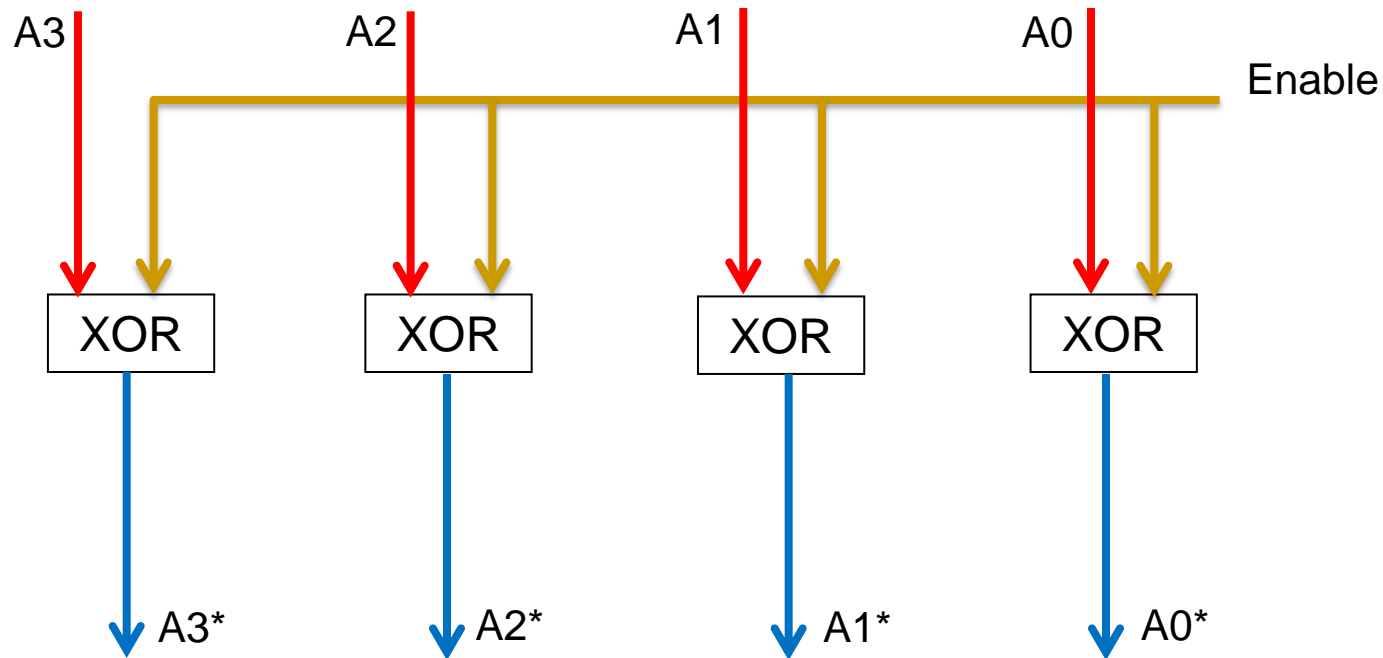
- Introducción
 - Complementadores
 - Suma: Semisumador y Sumador completo
 - Resta: Semirestador y Restador completo
 - Multiplicación: multiplicación en punto fijo
-

Introducción

- Realiza las operaciones aritméticas y lógicas con los datos buscados por el procesador.
 - Operaciones básicas: suma, resta, multiplicación, división. Op. Básica: la **suma**
-
- Resta: Sumando el sustraendo complementado a 2 al minuendo.
 - Multiplicación: repetición de sumas.
 - División: repetición de restas.
-

Complementadores

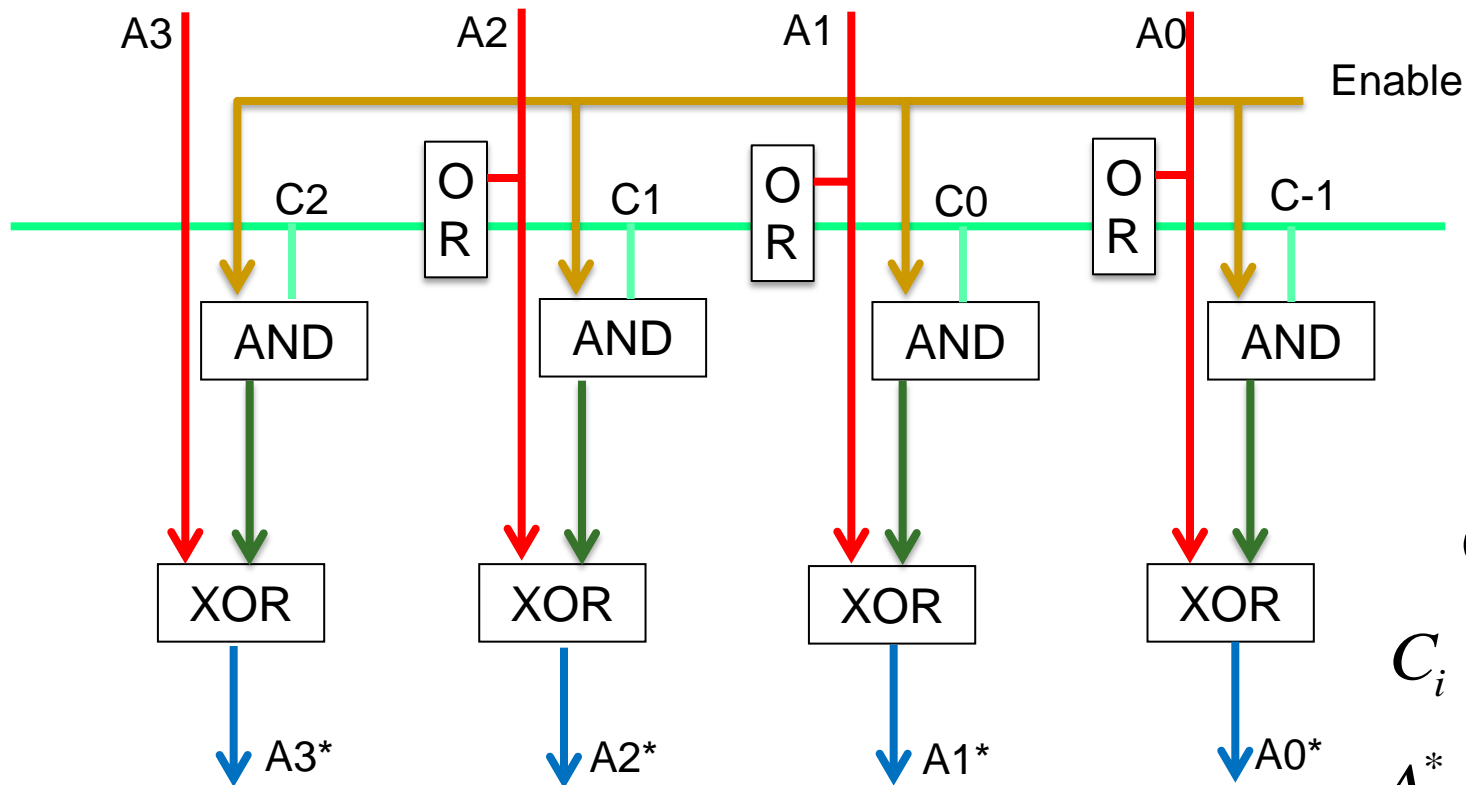
1) Complemento a uno:



$$A_i^* = A_i \bar{E} + \bar{A}_i E$$

Complementadores

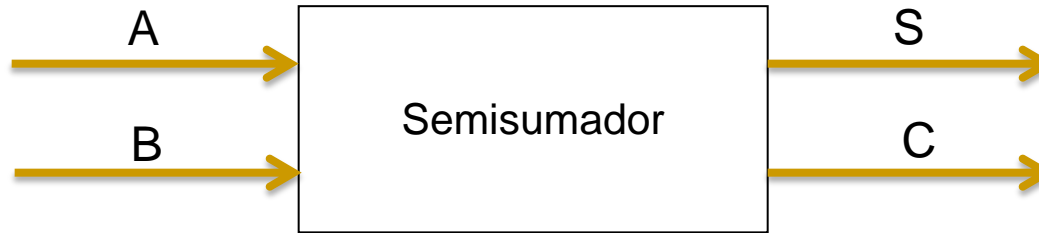
2) Complemento a dos:



$$C_{-1} = 0$$
$$C_i = A_i + C_{i-1}$$
$$A_i^* = A_i \oplus EC_{i-1}$$

Suma: Semisumador

- El semisumador suma dos dígitos binarios, produciendo una salida de suma y un acarreo



$$S = A \oplus B$$

$$C = AB$$

Suma: Sumador completo

- El sumador completo debe tener en cuenta el arrastre del dígito previo

$$S_i = \overline{A_i} \overline{B_i} C_{i-1} + \overline{A_i} B_i \overline{C_{i-1}} + A_i \overline{B_i} \overline{C_{i-1}} + A_i B_i C_{i-1}$$

$$C_i = A_i C_{i-1} + A_i B_i + B_i C_{i-1}$$

Sumador completo: Ec. Lógicas (1)

- Las fórmulas para el sumador completo son:

$$S_i = \overline{A_i} \overline{B_i} C_{i-1} + \overline{A_i} B_i \overline{C_{i-1}} + A_i \overline{B_i} \overline{C_{i-1}} + A_i B_i C_{i-1}$$

$$C_i = \overline{A_i} B_i C_{i-1} + A_i \overline{B_i} C_{i-1} + A_i B_i \overline{C_{i-1}} + A_i B_i C_{i-1} = A_i C_{i-1} + A_i B_i + B_i C_{i-1}$$

- Estas funciones pueden agruparse para dar

$$S_i = (\overline{A_i} \overline{B_i} + A_i B_i) C_{i-1} + (\overline{A_i} B_i + A_i \overline{B_i}) \overline{C_{i-1}}$$

Sumador completo: Ec. Lógicas (2)

- Estas funciones pueden agruparse para dar

$$S_i = (\bar{A}_i\bar{B}_i + A_iB_i)C_{i-1} + (\bar{A}_iB_i + A_i\bar{B}_i)\bar{C}_{i-1}$$

- Teniendo en cuenta que:

$$(\overline{\bar{A}B + A\bar{B}}) = (A + \bar{B})(\bar{A} + B) = AB + \bar{B}\bar{A}$$

- Llamando

$$Z = (\bar{A}B + A\bar{B})$$

$$\bar{Z} = (\bar{A}\bar{B} + AB)$$

- Entonces

$$S_i = Z_i\bar{C}_{i-1} + \bar{Z}_iC_{i-1} = Z_i \oplus C_{i-1}$$

$$C_i = Z_iC_{i-1} + A_iB_i$$

- Por tanto, la suma completa puede obtenerse mediante dos semisumadores en cascada.

Resta: Semirestador

- El semirestador resta dos dígitos binarios, produciendo una salida de diferencia y un préstamo



$$D = A \oplus B$$

$$P = \overline{A}B$$

Resta: Restador completo

- El restador completo debe tener en cuenta el préstamo del dígito previo

$$D_i = \overline{A_i} \overline{B_i} P_{i-1} + \overline{A_i} B_i \overline{P_{i-1}} + A_i \overline{B_i} \overline{P_{i-1}} + A_i B_i P_{i-1}$$

$$P_i = \overline{A_i} P_{i-1} + \overline{A_i} B_i + B_i P_{i-1}$$

- Igualmente, el restador completo se puede obtener mediante dos semirestadores en cascada

$$D_i = Z_i \overline{P_{i-1}} + \bar{Z}_i P_{i-1} = Z_i \oplus P_{i-1}$$

$$P_i = \bar{Z}_i P_{i-1} + \bar{A}_i B_i$$

Comparación

- Las fórmulas para el sumador completo son:

$$S_i = Z_i \overline{C_{i-1}} + \overline{Z_i} C_{i-1} = Z_i \oplus C_{i-1}$$

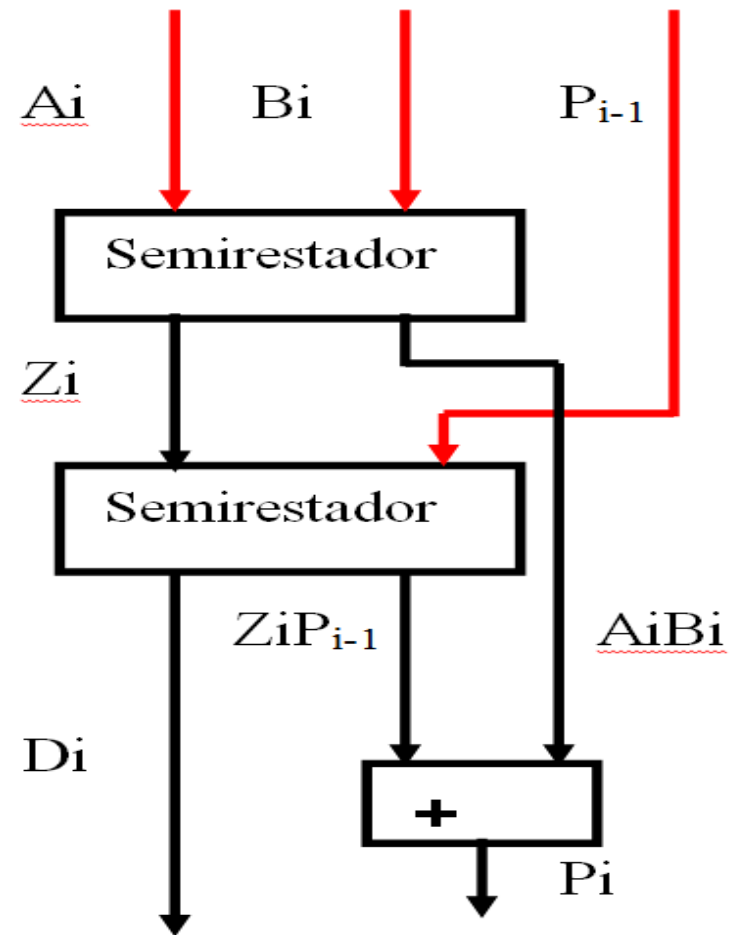
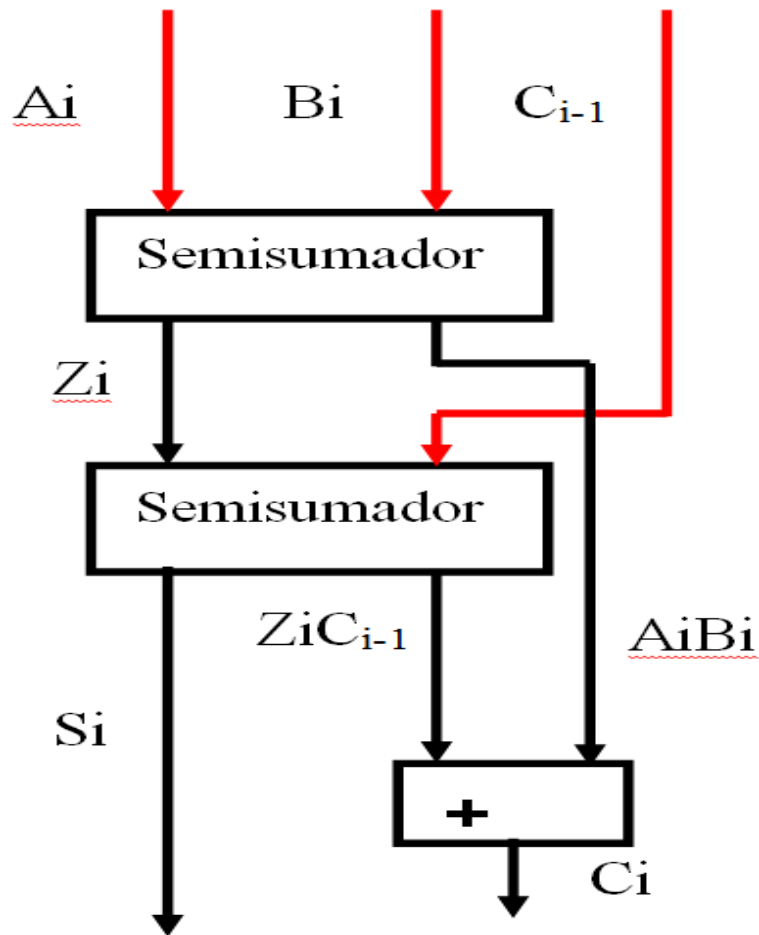
$$C_i = Z_i C_{i-1} + A_i B_i$$

- Igualmente, el restador completo se puede obtener mediante dos semirestadores en cascada

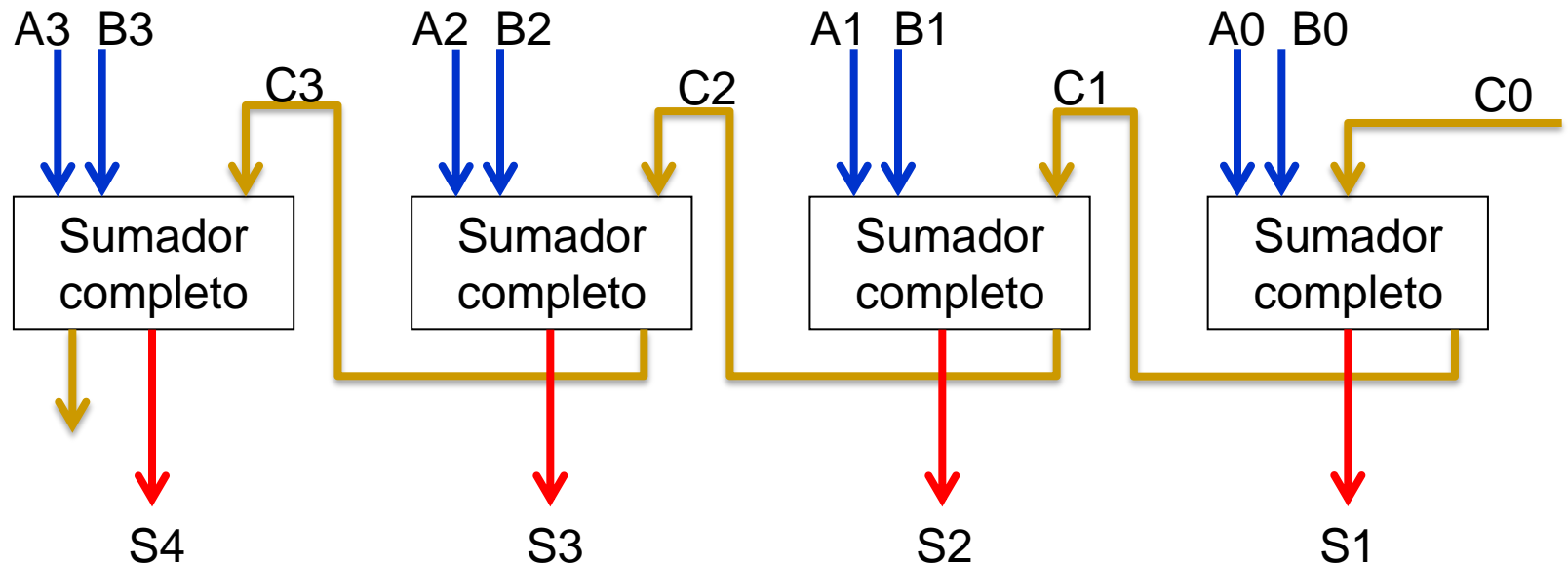
$$D_i = Z_i \overline{P_{i-1}} + \overline{Z_i} P_{i-1} = Z_i \oplus P_{i-1}$$

$$P_i = \overline{Z_i} P_{i-1} + \overline{A_i} B_i$$

Sumador y Restador completo



Sumar binarios de varios dígitos



- Es un circuito sumador paralelo: *n bit ripple carry adder*. Es el esquema más sencillo, pero conlleva retardos. Hay otros esquemas más complicados con menos retardos.

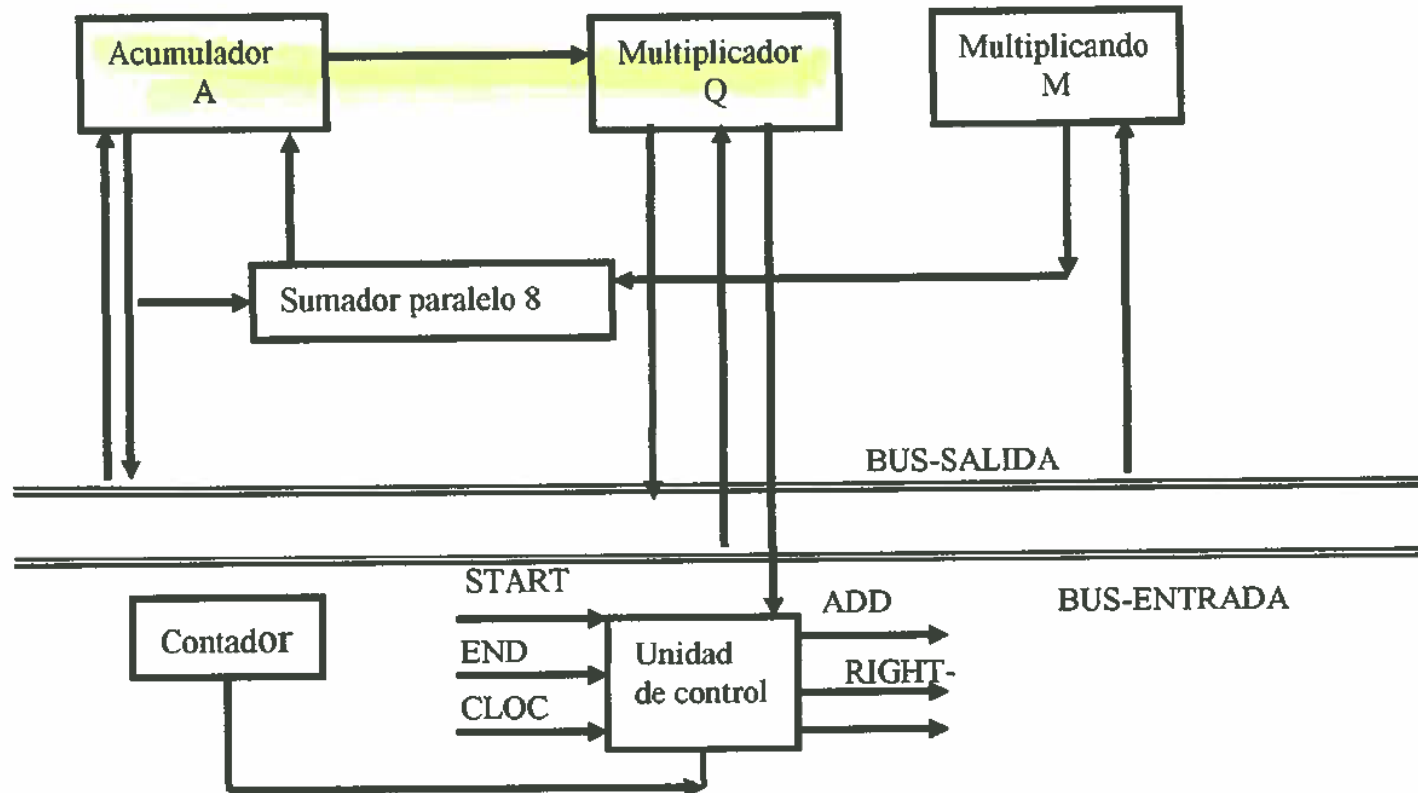
Multiplicación

- Requiere más hardware que la suma, por esto a veces no puede ser incluida en el repertorio de instrucciones de algunos ordenadores pequeños.
- Se implementa con **sumas sucesivas**:
 - 1) Sumar el multiplicando M consigo mismo Q veces, siendo Q el multiplicador: simple, pero demasiado lento.
 - 2) Multiplicar M por los k bits de Q cada vez y sumar los términos resultantes: se usan **sumas** y **desplazamientos**. Poco eficiente (hay que almacenar todos los productos hasta que se haya hecho la última suma).
 - 3) Sumar cada término a los anteriores para formar un **producto parcial**.

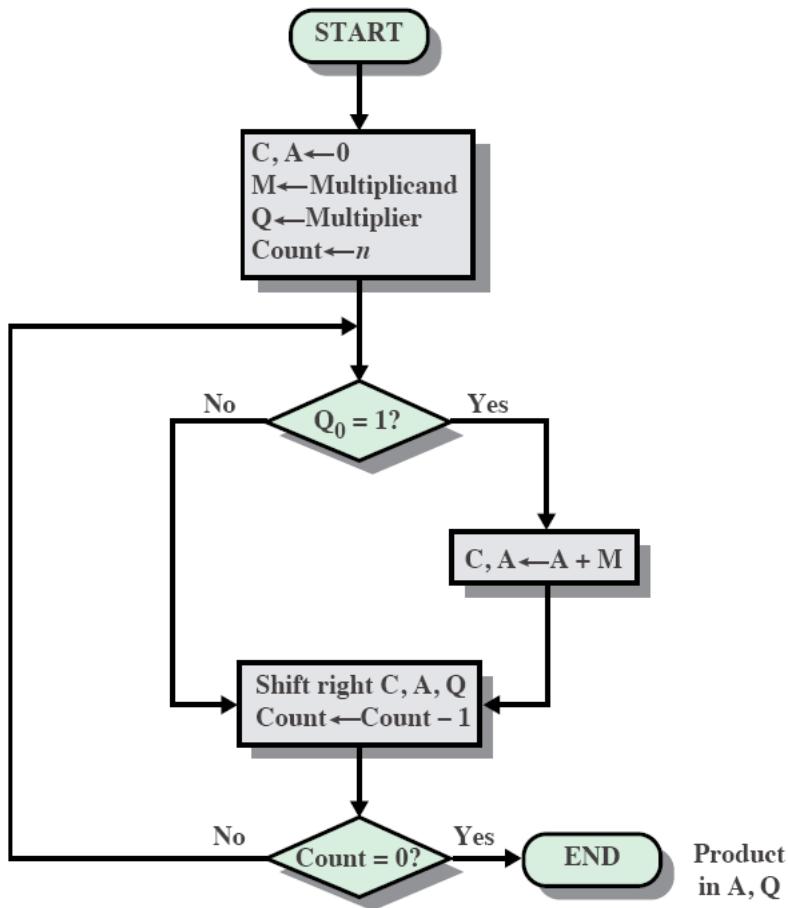
Multiplicación en punto fijo

Posible circuito: para números sin signo

Q y A están conectados físicamente. El desplazamiento a la derecha afecta a los dos registros. El resultado final queda almacenado en estos registros



Multiplicación en punto fijo



- 1) Registro A se inicializa a 0.
- 2) A y Q van a funcionar como un registro doble.
- 3) Registro M contiene multiplicando
- 4) Se necesita el bit de carry
- 5) En cada etapa
 - 1) Se comprueba el bit LSB
 - 2) Si es cero solo hay que desplazar A y Q 1 bit a derecha
 - 3) Si es uno, primero se suma M al registro A y después se desplazan A y Q 1 bit a la derecha
- 6) El bit que entra por la izquierda es copia del bit de carry