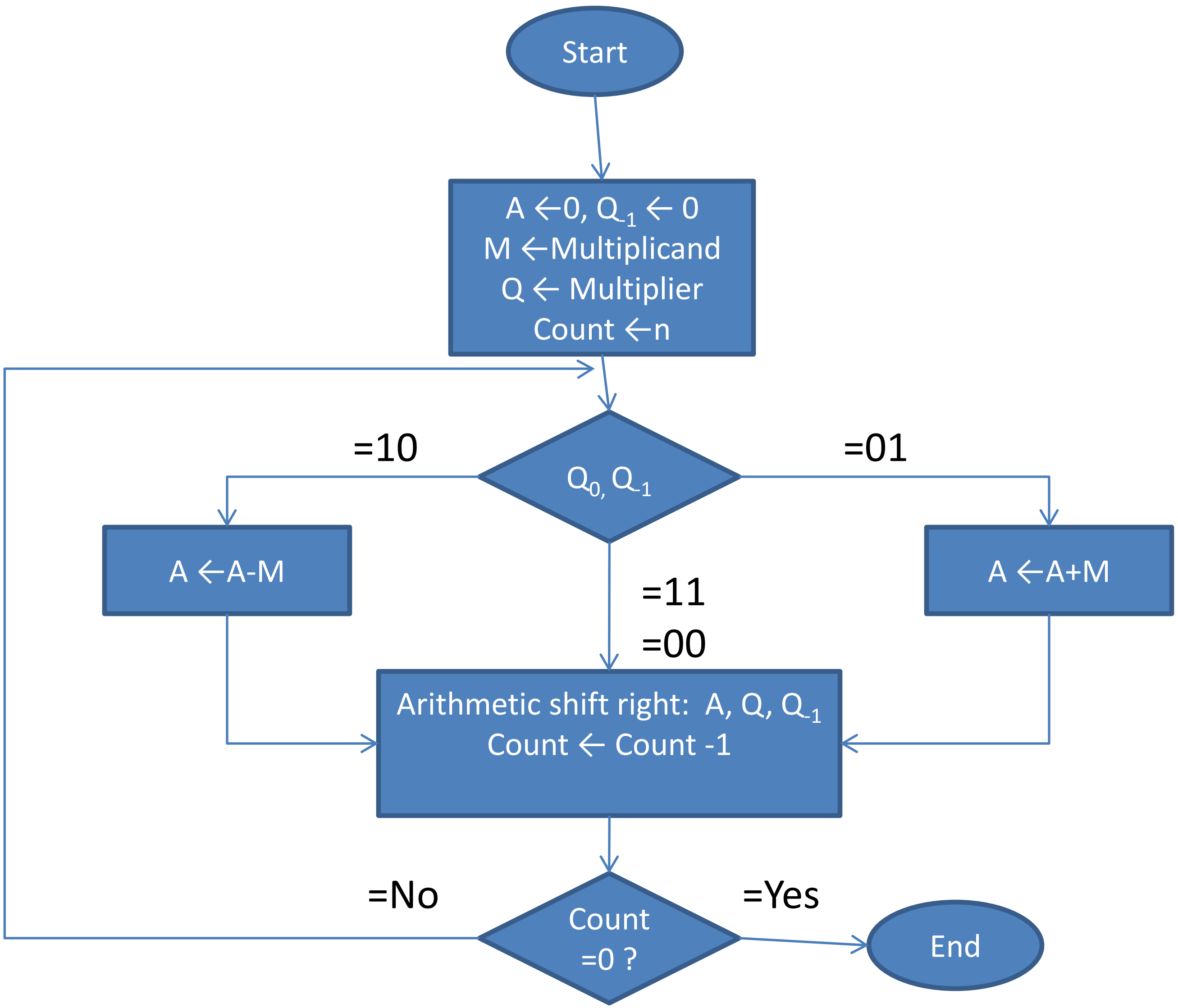
Booth’s Algorithm

Booth's Algorithm



Why Booth's Algorithm works?

- **Case 1: Positive multiplier**
  - Consider a positive multiplier consisting of one block of 1s surrounded by 0s.

    00011110

    $$M \times (00011110) = M \times (2^4 + 2^3 + 2^2 + 2^1)$$
    $$= M \times (16 + 8 + 4 + 2)$$
    $$= M \times 30$$

  - The number of Such operations can be reduced to two if we observe that

    $$2^n + 2^{n-1} + \dots + 2^{n-k} = 2^{n+1} - 2^{n-k}$$

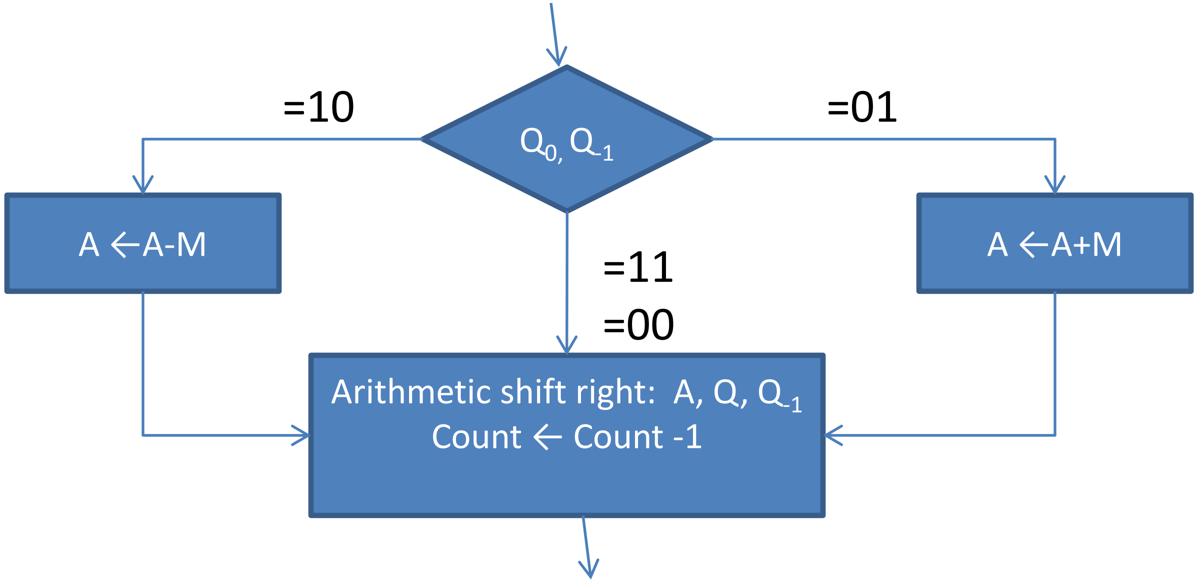
$$M \times (00011110) = M \times (2^5 - 2^1) = M \times (32 - 2) = M \times 30$$

Why Booth's Algorithm works? (contd.)

- Another Example

$$M \times (01111010) = M \times (2^6 + 2^5 + 2^4 + 2^3 + 2^1)$$
$$= M \times (2^7 - 2^3 + 2^2 - 2^1)$$

- Booth's algorithm confirms to this scheme by performing
  - a subtraction when the first 1 of the block is encountered (1-0) and
  - an addition when the end of the block is encountered (0-1).



Why Booth's Algorithm works? (contd.)

- **Case 2: Negative Multiplier**
  - Let X be a negative number in twos complement notation.

    Representation of X = $\{1\ x_{n-2}\ x_{n-3} \ldots x_1\ x_0\}$

  - Then the value of X can be expressed as follows:

$$-2^{n-1} + (x_{n-2} \times 2^{n-2}) + (x_{n-3} \times 2^{n-3}) + \ldots \ldots + (x_1 \times 2^1) + (x_0 \times 2^0) \ldots \ldots (1)$$

  - Assume that the leftmost 0 is in k^{th} position. Thus X is of the form,

    Representation of X = $\{111\ldots10\ x_{k-1}\ x_{k-2} \ldots x_1\ x_0\}$

  - Then the value of X can be expressed as follows:

$$\boldsymbol{-2^{n-1}+2^{n-2} + \cdots + 2^{k+1} + (x_{k-1} \times 2^{k-1}) + \cdots + (x_0 \times 2^0) \ldots \ldots (2)}$$

  - From previous equation, we can write,

$$\boldsymbol{2^{n-2} + \cdots + 2^{k+1} = 2^{n-1} - 2^{k+1}}$$

Why Booth's Algorithm works? (contd.)

- Then the value of X can be expressed as follows:

$$-\mathbf{2^{n-1}}+\mathbf{2^{n-2}}+\cdots+\mathbf{2^{k+1}}+(\boldsymbol{x_{k-1}}\times\mathbf{2^{k-1}})+\cdots+(\boldsymbol{x_0}\times\mathbf{2^0})\ \ldots\ldots(\mathbf{2})$$

- From previous equation, we can write,

$$\mathbf{2^{n-2}}+\cdots+\mathbf{2^{k+1}}=\mathbf{2^{n-1}}-\mathbf{2^{k+1}}$$

- Rearranging,

$$-\mathbf{2^{n-1}}+\mathbf{2^{n-2}}+\cdots+\mathbf{2^{k+1}}=-\mathbf{2^{k+1}}\ \ldots\ldots(\mathbf{3})$$

- Substituting equation (3) into (2),

$$\mathbf{X}=-\mathbf{2^{k+1}}+(\boldsymbol{x_{k-1}}\times\mathbf{2^{k-1}})+\cdots+(\boldsymbol{x_0}\times\mathbf{2^0})$$

- As the algorithm scans over the leftmost 0 and encounters the next 1, a 1-0 transition occurs and a subtraction takes place.

Why Booth's Algorithm works? (contd.)

- Example

$$M \times (11111010) = M \times (-2^7+2^6+2^5+2^4+2^3+2^1)$$

$$= M \times (-2^3+2^2-2^1)$$

$$= M \times (-6)$$