Multiplication algorithms

Multiplication algorithms

- Multiplication of two fixed point binary numbers in signed magnitude representation is done with paper and pencil of successive shift and add operation.

```
 23        10111     Multiplicand
 19      × 10011     Multiplier
 --     ---------
           10111
          10111
         00000    +
        00000
       10111
       ---------
437    110110101     Product
```

Multiplication algorithms

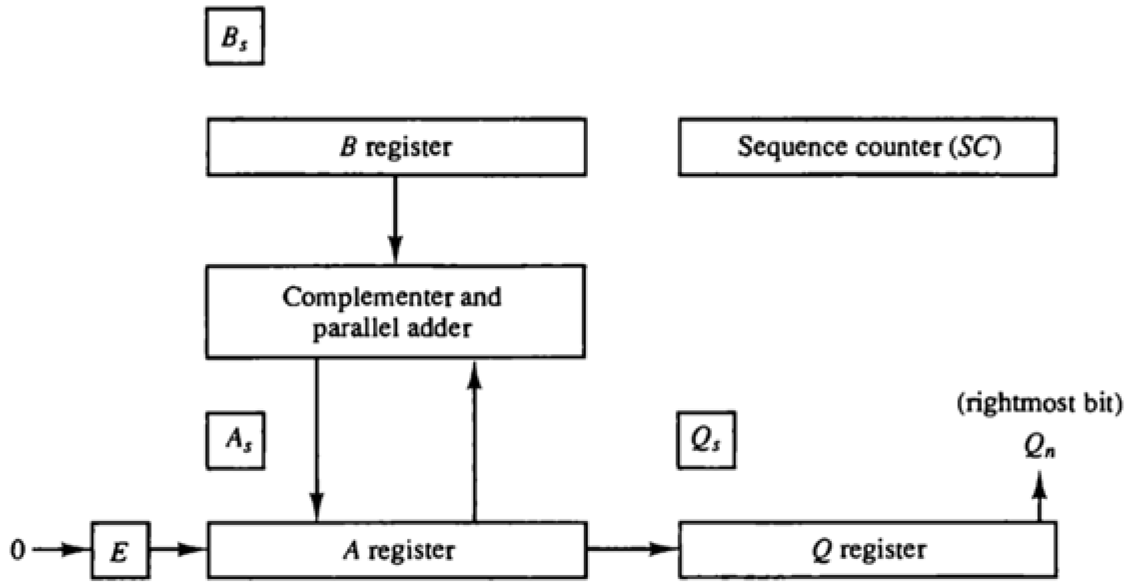
- If the multiplier bit is a 1,the multiplicand is copied down; otherwise zero are copied down.
- Finally all the partial products are added to get the desired product.
- The sign of the product is determined from the sign of the multiplicand and multiplier.
- If they are same sign then product is positive and if they are of different sign, the sign of the product is negative.

Hardware Implementation for Signed-Magnitude data

- First instead of providing register to store and add simultaneously as many binary numbers as there are bits in the multiplier , it is convenient to provide an adder for the summation of only two binary numbers and successively accumulate the partial products in a register.
- Second instead of shifting the multiplicand to the left , the partial product is shifted to the right .
- Third when the corresponding bits of the multiplier is 0 there is no need al add all zero’s to the partial product.

Hardware Implementation for Signed-Magnitude data

- The hardware for multiplication consists of the equipment shown in following fig.

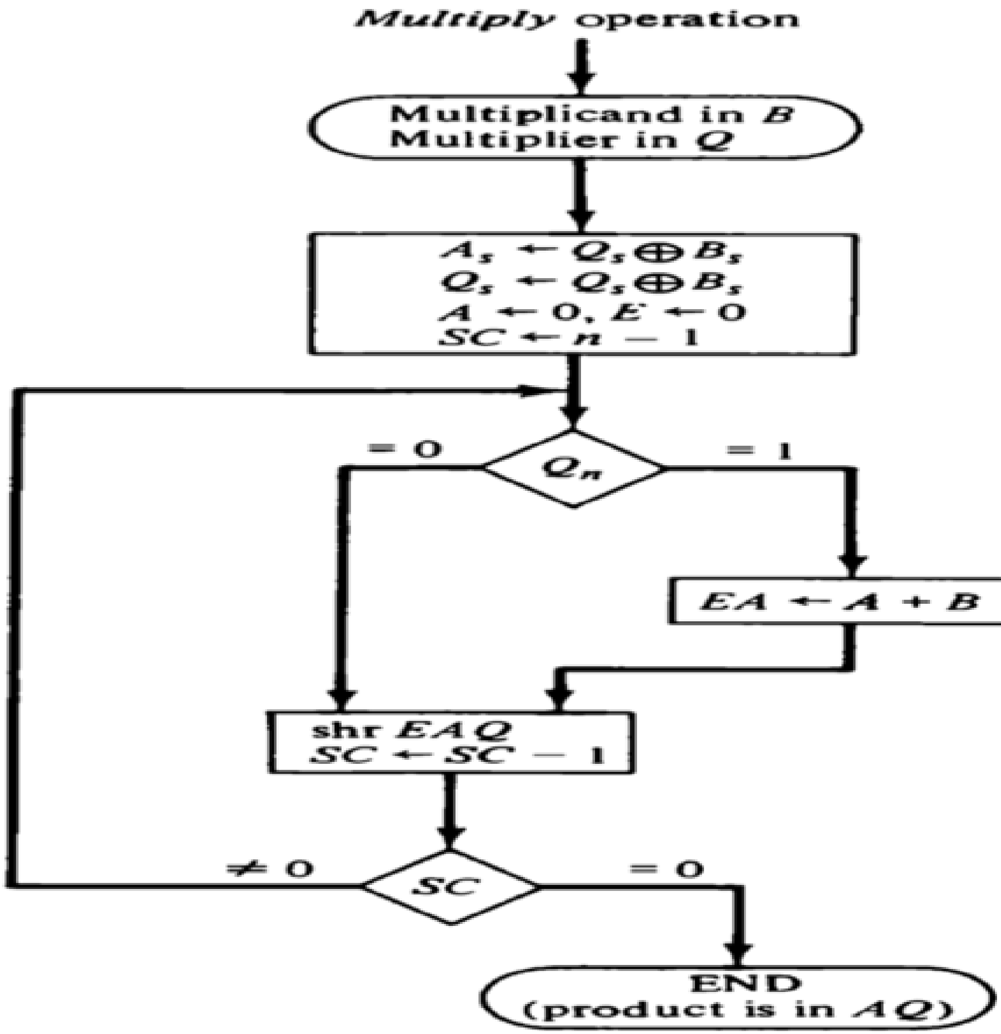


Hardware Implementation for Signed-Magnitude data

- The multiplier stored in the Q register and its sign in Q_s The sequence counter SC is initially set to a number equal to the number of bits in the multiplier.
- The counter is decremented by 1 after forming each partial product. When the counter reaches to zero, the product is formed and process stops.
- Initially the multiplicand is in B register and multiplier in Q register.
- The sum of A and B forms a partial product which is transferred to the EA register .
- The shift will be denoted by the statement shr EAQ to designate the right shift.
- The least significant bit of A is shifted into the most significant position of Q.

Hardware Algorithm

Flowchart for multiply operation.



Hardware Algorithm

- Initially the multiplicand is in B and the multiplier in Q and there corresponding signs are in B_s and Q_s respectively.
- Register A and E are cleared and the sequence counter SC is set to a number equal to the number of bits of the multiplier.
- After the initialization , the low order bit of the multiplier is in Qn is tested. If it is 1,the multiplicand in B is added to the present partial product in A . If it is 0, nothing is done.
- Register EAQ shifted once to the right to form the new partial product.
- The SC is decremented by one and its new value is checked. If it is not zero the process is repeated and it it is zero the process stops.
- The final product is available in both A and Q, with A holding the most significant bits and Q holding the least significant bits.

Example
Multiply 23 X 19 = 437

TABLE 10-2 Numerical Example for Binary Multiplier

Multiplicand B = 10111	E	A	Q	SC
Multiplier in Q	0	00000	10011	101
Q_n = 1; add B		10111		
First partial product	0	10111		
Shift right EAQ	0	01011	11001	100
Q_n = 1; add B		10111		
Second partial product	1	00010		
Shift right EAQ	0	10001	01100	011
Q_n = 0; shift right EAQ	0	01000	10110	010
Q_n = 0; shift right EAQ	0	00100	01011	001
Q_n = 1; add B		10111		
Fifth partial product	0	11011		
Shift right EAQ	0	01101	10101	000
Final product in AQ = 0110110101				