Lab Sheet-5

DIVISION OF TWO UNSIGNED INTEGER BINARY NUMBERS

Objective:
To implement non-restoring division algorithm in digital computer.

In the non-restoring algorithm, B is not added if the difference is negative but instead, the negative difference is shifted left and then B is added. Here B is subtracted if the previous value of $Q_{LSB\ was}$ 1, but B is added if the previous value of $Q_{LSB\ was}$ 0 and no restoring of partial remainder is required. The first time the dividend is shifted, B must be subtracted. Also, if the last bit of the quotient is 0, the partial remainder must be restored to obtain the correct final remainder.
Consider two binary numbers A and B. A is the dividend, B the divisor and Q = A / B the quotient. We assume that A>B and B!=0. The flow chart of the algorithm used, is given in figure 5.1.
Let us take examples:

1)

Dividend (Register - A) = 12 Equivalent binary representation is 1100 and
divisor (Register - B) = 4 Equivalent binary representation is 0100.

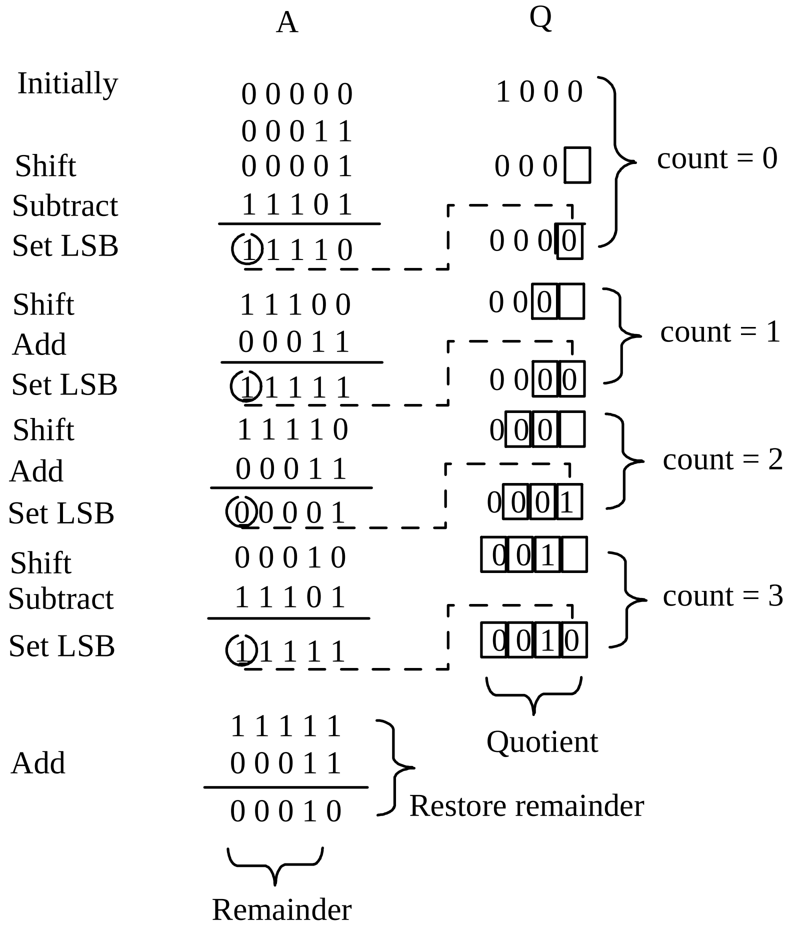
Subtraction may be achieved by adding 2's complement of B as we have done in lab 3 and here it is 1100. Double length dividend is stored in registers AQ.

	A	Q	
Initially	0 0 0 0	1 1 0 0	
	0 1 0 0		
Shift	0 0 0 1	1 0 0 □	count = 0
Subtract	1 1 0 0		
Set LSB	1 1 0 1	1 0 0 0	
Shift	1 0 1 1	0 0 0 □	
Add	0 1 0 0		count = 1
Set LSB	1 1 1 1	0 0 0 0	
Shift	1 1 1 0	0 0 0 □	
Add	0 1 0 0		count = 2
Set LSB	0 0 1 0	0 0 0 1	
Shift	0 1 0 0	0 0 1 □	
Subtract	1 1 0 0		count = 3
	0 0 0 0	0 0 1 1	
	Remainder	Quotient	

2)

Dividend (Register - A) = 8 Equivalent binary representation is 1000 and divisor (Register - B) = 3 Equivalent binary representation is 00011.

Subtraction may be achieved by adding 2's complement of B as we have done in lab 3 and here it is 11101. Dividend is stored in registers AQ.



	A	Q	
Initially	0 0 0 0 0	1 0 0 0	count = 0
	0 0 0 1 1		
Shift	0 0 0 0 1	0 0 0 □	
Subtract	1 1 1 0 1		
Set LSB	1 1 1 1 0	0 0 0 0	
Shift	1 1 1 0 0	0 0 0 □	count = 1
Add	0 0 0 1 1		
Set LSB	1 1 1 1 1	0 0 0 0	
Shift	1 1 1 1 0	0 0 0 □	count = 2
Add	0 0 0 1 1		
Set LSB	0 0 0 0 1	0 0 0 1	
Shift	0 0 0 1 0	0 0 1 □	count = 3
Subtract	1 1 1 0 1		
Set LSB	1 1 1 1 1	0 0 1 0	
		Quotient	
	1 1 1 1 1		Restore remainder
Add	0 0 0 1 1		
	0 0 0 1 0		
	Remainder		

Let the number of bits stored in register Q is n Registers AQ is now shifted to the left with zero insertion into $Q_{LSB.}$ Initialize the counter to zero value. And divisor is subtracted by adding 2's complement value. If A_{MSB} =1, set Q_{LSB} with value 0 and then increment the counter value by 1. The partial remainder is shifted to the left and then B is added to the partial remainder. . If A_{MSB} =0, set Q_{LSB} with value 1 and then increment the counter value by 1. Process is repeated until count = n-1 i.e. all quotient bits are formed. If the last bit of the quotient is 0, the partial remainder must be restored to obtain the correct final remainder. Finally result, Quotient is in Q and the final remainder is in A, is obtained.

The flow chart for non-restoring division is shown below.

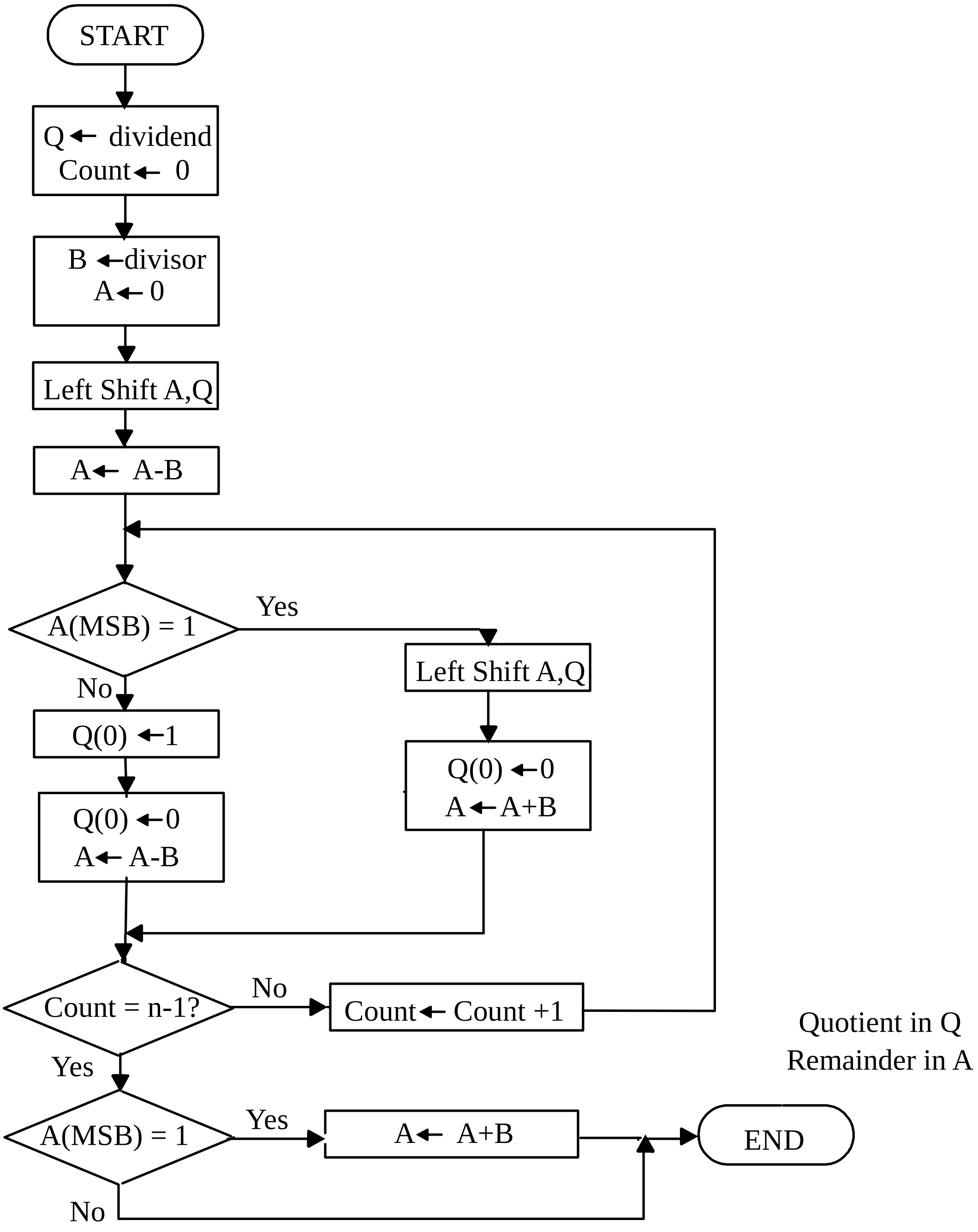


Figure 5.1