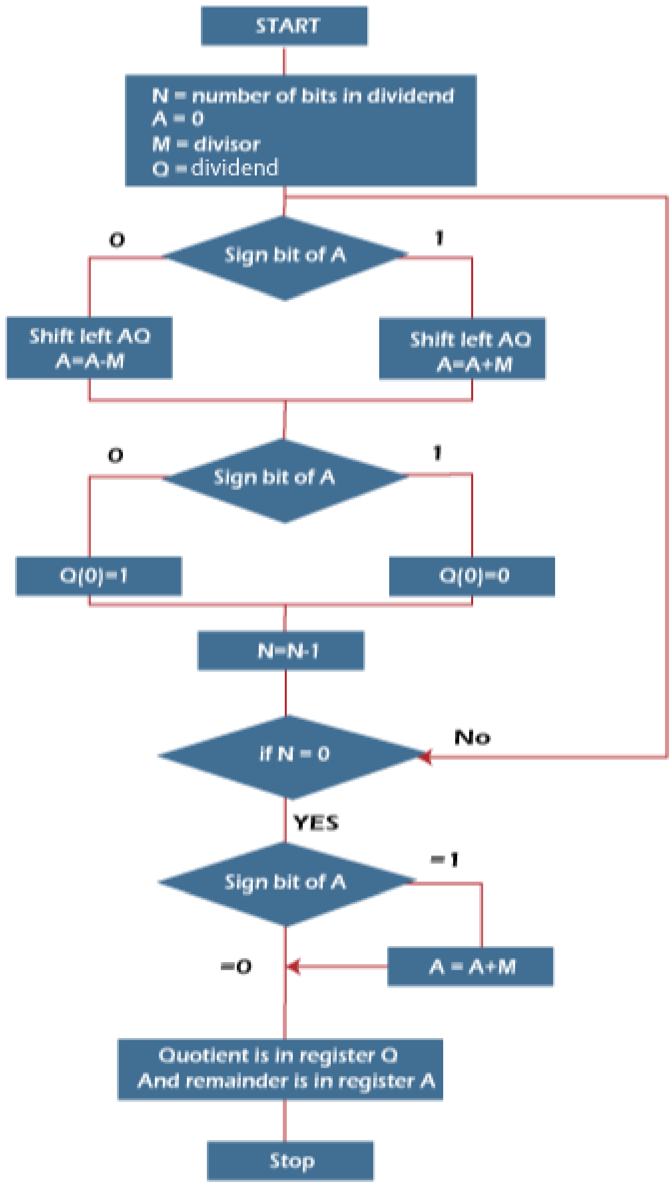
Non restoring Division Algorithm

Instead of the quotient digit set {0, 1}, the set {-1, 1} is used by the non-restoring division. The non-restoring division algorithm is more complex as compared to the restoring division algorithm. But when we implement this algorithm in hardware, it has an advantage, i.e., it contains only one decision and addition/subtraction per quotient bit. After performing the subtraction operation, there will not be any restoring steps. Due to this, the numbers of operations basically cut down up to half. Because of the less operation, the execution of this algorithm will be fast. This algorithm basically performs simple operations such as addition, subtraction. In this method, we will use the sign bit of register A. 0 is the starting value/bit of register A.



Now we will learn steps of the non-restoring division algorithm, which are described as follows:

Step 1: In this step, the corresponding value will be initialized to the registers, i.e., register A will contain value 0, register M will contain Divisor, register Q will contain Dividend, and N is used to specify the number of bits in dividend.

Step 2: In this step, we will check the sign bit of A.

Step 3: If this bit of register A is 1, then shift the value of AQ through left, and perform A = A + M. If this bit is 0, then shift the value of AQ into left and perform A = A - M. That means in case of 0, the 2's complement of M is added into register A, and the result is stored into A.

Step 4: Now, we will check the sign bit of A again.

Step 5: If this bit of register A is 1, then Q[0] will become 0. If this bit is 0, then Q[0] will become 1. Here Q[0] indicates the least significant bit of Q.

Step 6: After that, the value of N will be decremented. Here N is used as a counter.

Step 7: If the value of N = 0, then we will go to the next step. Otherwise, we have to again go to step 2.

Step 8: We will perform A = A + M if the sign bit of register A is 1.

Step 9: This is the last step. In this step, register A contains the remainder, and register Q contains the quotient.

For example:

In this example, we will perform a Non-Restoring Division algorithm with the help of an Unsigned integer.

1. Dividend = 11
2. Divisor = 3
3. -M = 11101

N	M	A	Q	Action
4	00011	00000	1011	Begin
	00011	00001	011_	Shift left AQ
	00011	11110	011_	A = A - M
3	00011	11110	0110	Q[0] = 0
	00011	11100	110_	Shift left AQ
	00011	11111	110_	A = A + M
2	00011	11111	1100	Q[0] = 0
	00011	11111	100_	Shift left AQ
	00011	00010	100_	A = A +M

1 00011 00010 1001 Q[0] = 1
00011 00101 001_ Shift left AQ
00011 00010 001_ A = A - M
0 00011 00010 0011 Q[0] = 1

So, register A contains the remainder 2, and register Q contains the quotient 3.