COA
Division Algorithm

Contents

1. Restoring Division Algorithm
2. Non Restoring Division Algorithm
3. Examples

Restoring Division Algorithm

Division of unsigned integers

Division is more complex operation than multiplication.

We know:

If Dividend = D
Divisor = V
Quotient = Q
& Remainder = R

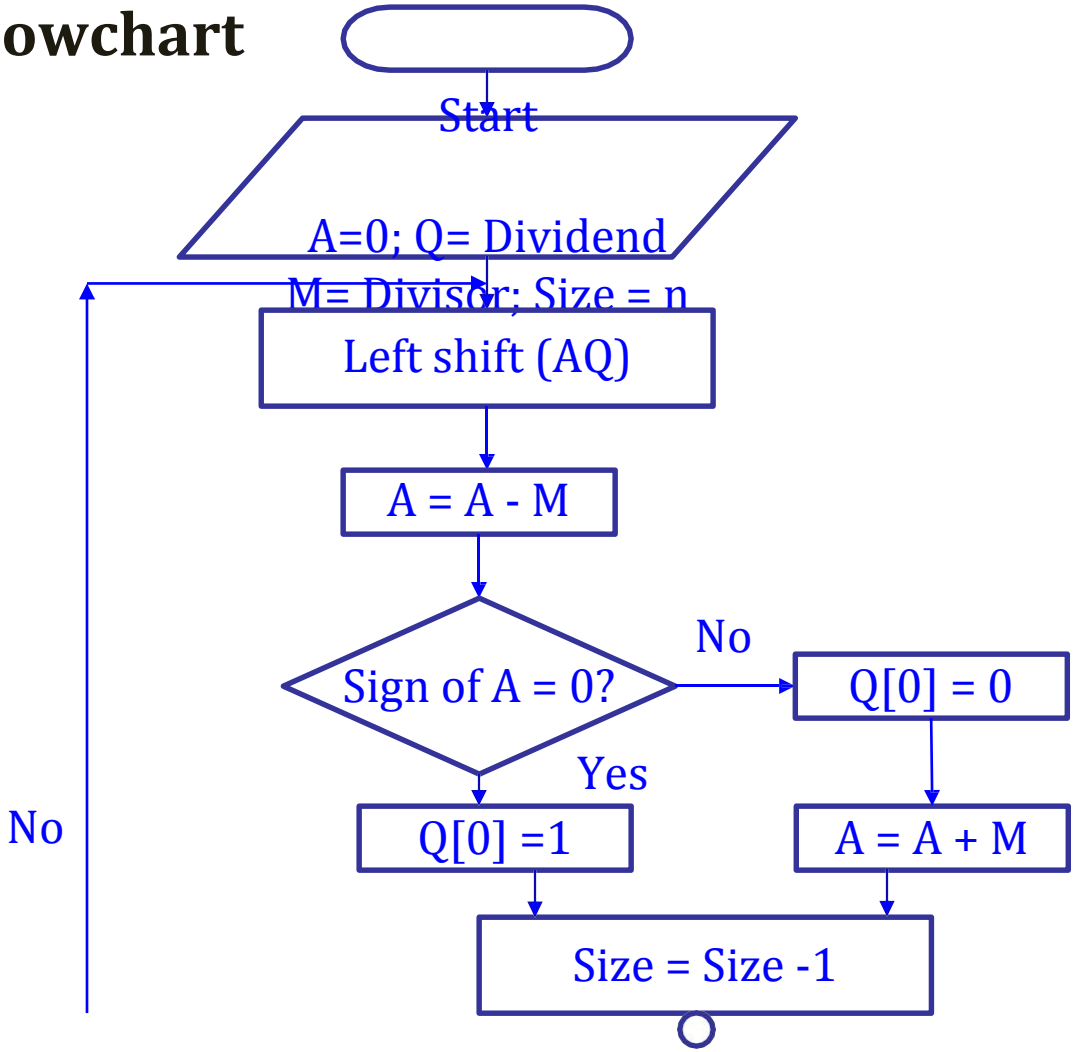
Then, D= QV + R where 0 ≤ R ≤ V

In case of Restoring Division method, three n bit registers: A, M, & Q are used
for dividing two n bit
numbers. Here, M = Holding
the Divisor

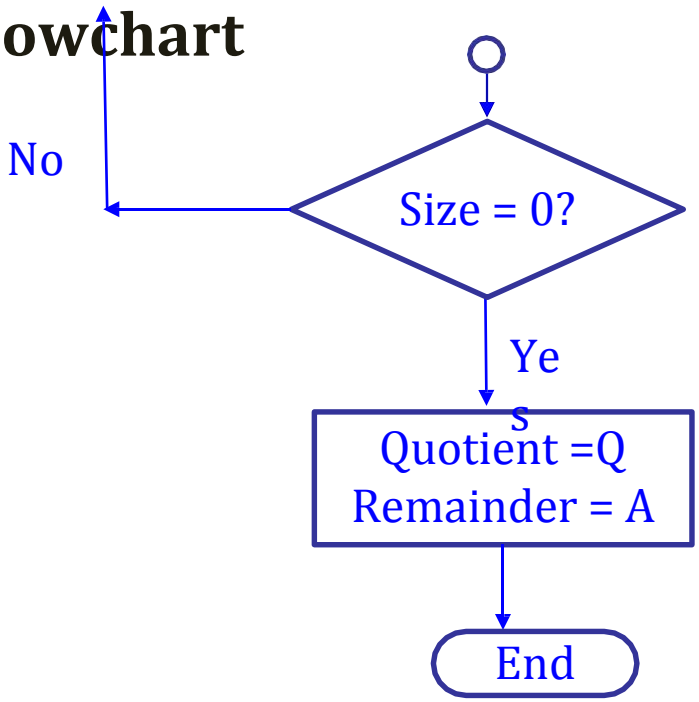
When the algorithm terminates, register A contains the remainder and Q = Holding the quotient result.

- At the beginning, initialize register A = 0
- Register Q = Holding the Dividend

Restoring Division Algorithm Flowchart



Restoring Division Algorithm Flowchart



Restoring Division Algorithm Example

Let us consider, Dividend Q = 7 = 0111 Divisor M = 3 = 0011

Initial Configuration:	**M**	**A**	**Q**		**Size**
	00011	00000	0111		4
Step 1:					
LS (AQ)	00011	00000	111		
A=A-M	00011	11101	111		
As Sign of A = -ve Set Q[0] = 0 & Restore A	00011	00000	1110		3
Step 2:					
LS (AQ)	00011	00001	110		
A=A-M	00011	11110	110		
As Sign of A = -ve Set Q[0] = 0 & Restore A	00011	00001	1100		2

Restoring Division Algorithm Example

Previous Configuration:	M	A	Q	Size
	00011	00001	1100	2
Step 3:				
LS (AQ)	00011	00011	100	
A=A-M	00011	00000	100	
As Sign of A = +ve				
Set Q[0] = 1	00011	00000	1001	1
Step 4:				
LS (AQ)	00011	00001	001	
A=A-M	00011	11110	001	
As Sign of A = -ve				
Set Q[0] = 0				
& Restore A	00011	00001	0010	0

Therefore, Quotient Q = 0010 = 2

& Remainder A = 00001 = 1

Non Restoring Division Algorithm

In case of Restoring Division method, some extra additions are required to restore the number, when A is –ve. Proper restructuring of the Restoring Division algorithm can eliminate that restoration step. This is known as Non restoring Division algorithm.

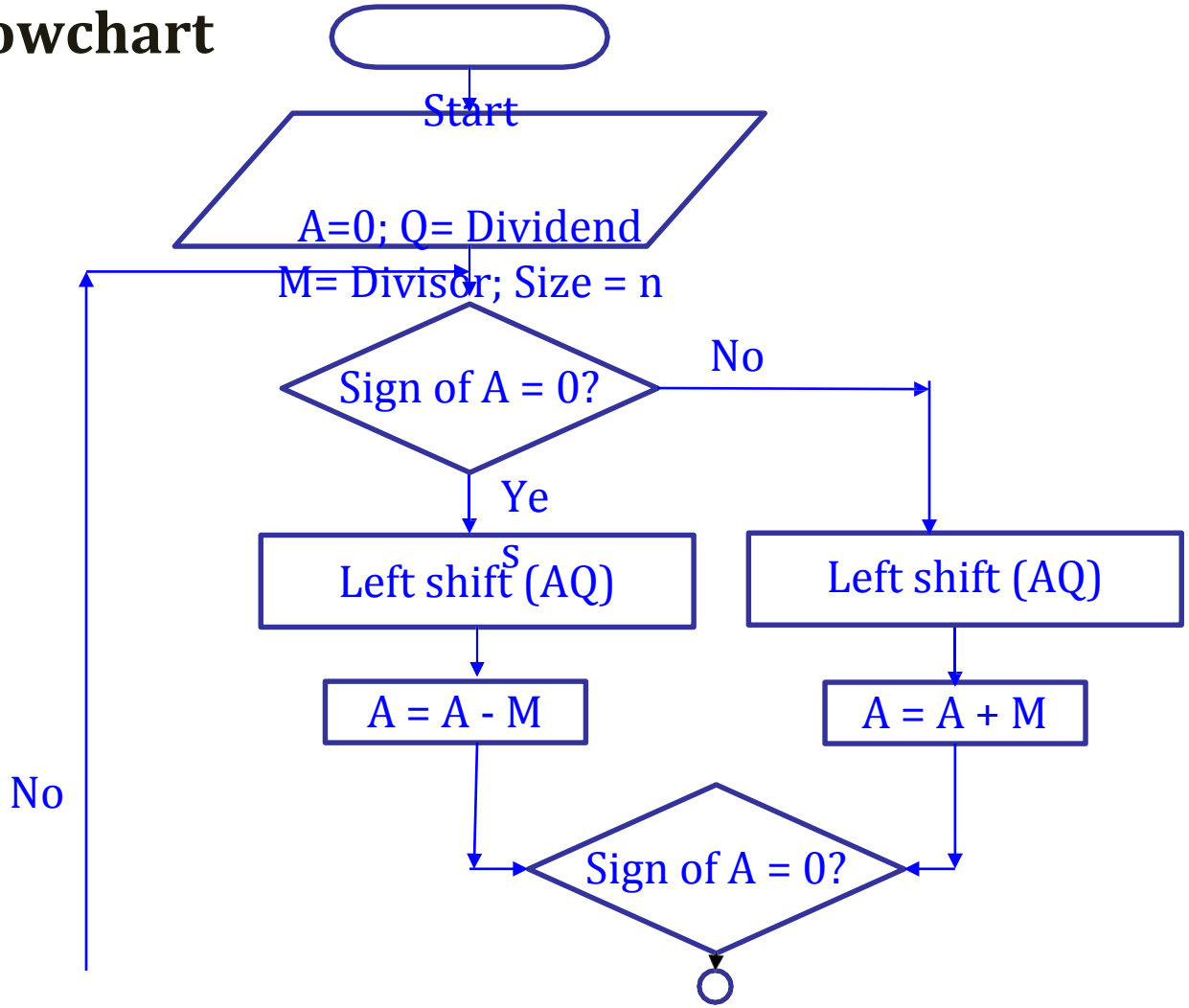
The three steps of Restoring Division algorithm were:

1. **Shift content of AQ register pair to the left one position.**
2. **A = A – M**
3. **If the sign of A is +ve after step 2, set Q[0] = 1; otherwise set Q[0] = 0 and restore A.**

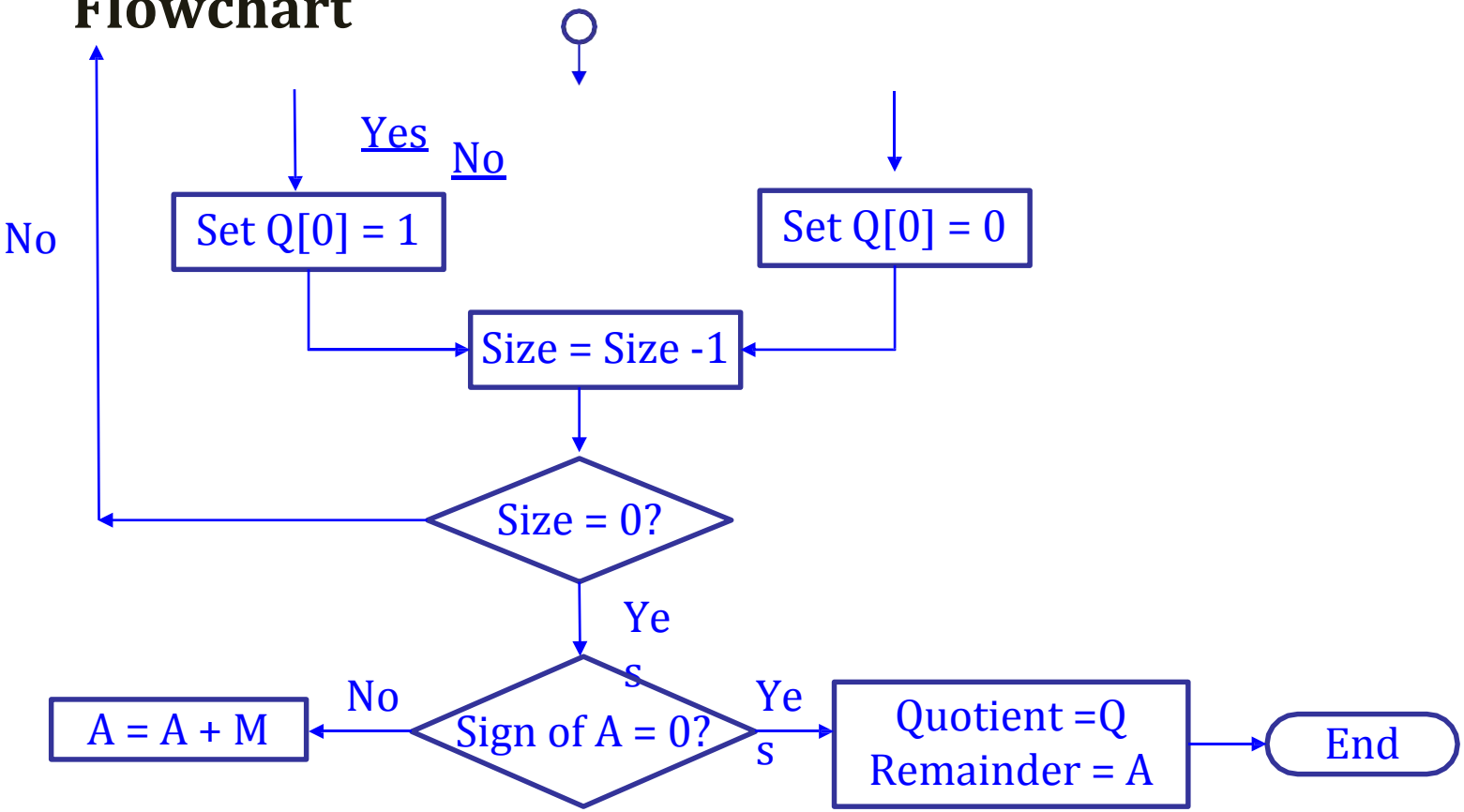
Now, if step 3 is performed first and then step 1 followed by step 2, we may get two cases:

1. If A is +ve, shifting A register to the left one position ≡ 2A – M
2. When A is –ve, first restore A by adding the content of M and then shift A to the left one position. After that A will be subtracted from; i.e. ≡ 2A + M

Non Restoring Division Algorithm Flowchart



Restoring Division Algorithm Flowchart



Non Restoring Division Algorithm Example

Let us consider, Dividend Q = 7 = 0111 Divisor M = 3 = 0011

Initial Configuration:	**M**	**A**	**Q**		**Size**
	00011	00000	0111		4
Step 1:					
As Sign of A = +ve					
Left Shift (AQ)	00011	00000		111	
A= A- M	00011	11101		111	
As Sign of A = -ve					
Set Q[0] = 0	00011	11101	1110		3
Step 2:					
As Sign of A = -ve					
Left Shift (AQ)	00011	11011		110	
A = A + M	00011	11110	110		
As Sign of A = -ve					
Set Q[0] = 0	00011	11110	1100	2	

Non Restoring Division Algorithm Example

	M	A	Q		Size
Previous Configuration:		00011	11110	1100	2
Step 3:					
As Sign of A = -ve					
Left Shift (AQ)		00011	11101	100	
A= A + M		00011	00000	100	
As Sign of A = +ve					
Set Q[0] = 1		00011	00000	1001	1
Step 4:					
As Sign of A = +ve					
Left Shift (AQ)		00011	00001	001	
A = A - M		00011	11110	001	
As Sign of A = -ve					
Set Q[0] = 0		00011	11110	0010	0

Therefore, Quotient Q = 0010 = 2

& Remainder A = 00001 = 1 (A = A + M)