Module 2
Data Representation And Computer Arithmetic

Dr. B. Bhanu Chander
Sr. Assistant Professor, SCOPE, VIT Chennai

Outline:

- Fixed Point Division Operation:
    - Restoring Division
    - Non-Restoring Division

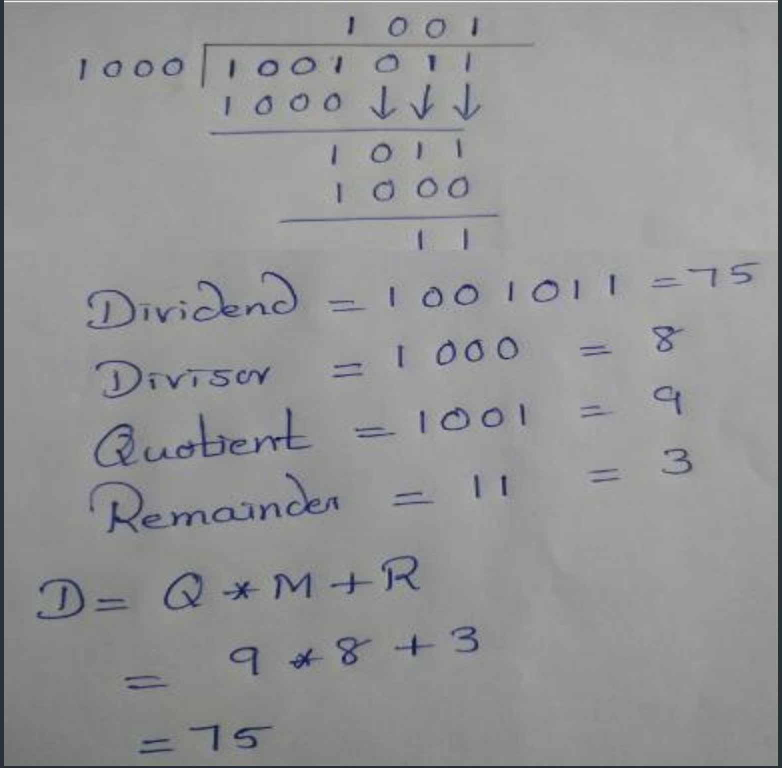
Integer Division Operation

- Division is more complex than multiplication
- It involves repetitive shifting and addition or subtraction
- In integer division operation, a divisor M and a dividend D are given.
- We need to find quotient Q and remainder R.
- The relationship between the components is given by,

$$D=Q*M+R$$

Simple Integer Division: Pencil-Paper Method

Let,
Dividend, D=1000011
Divisor, M=1001



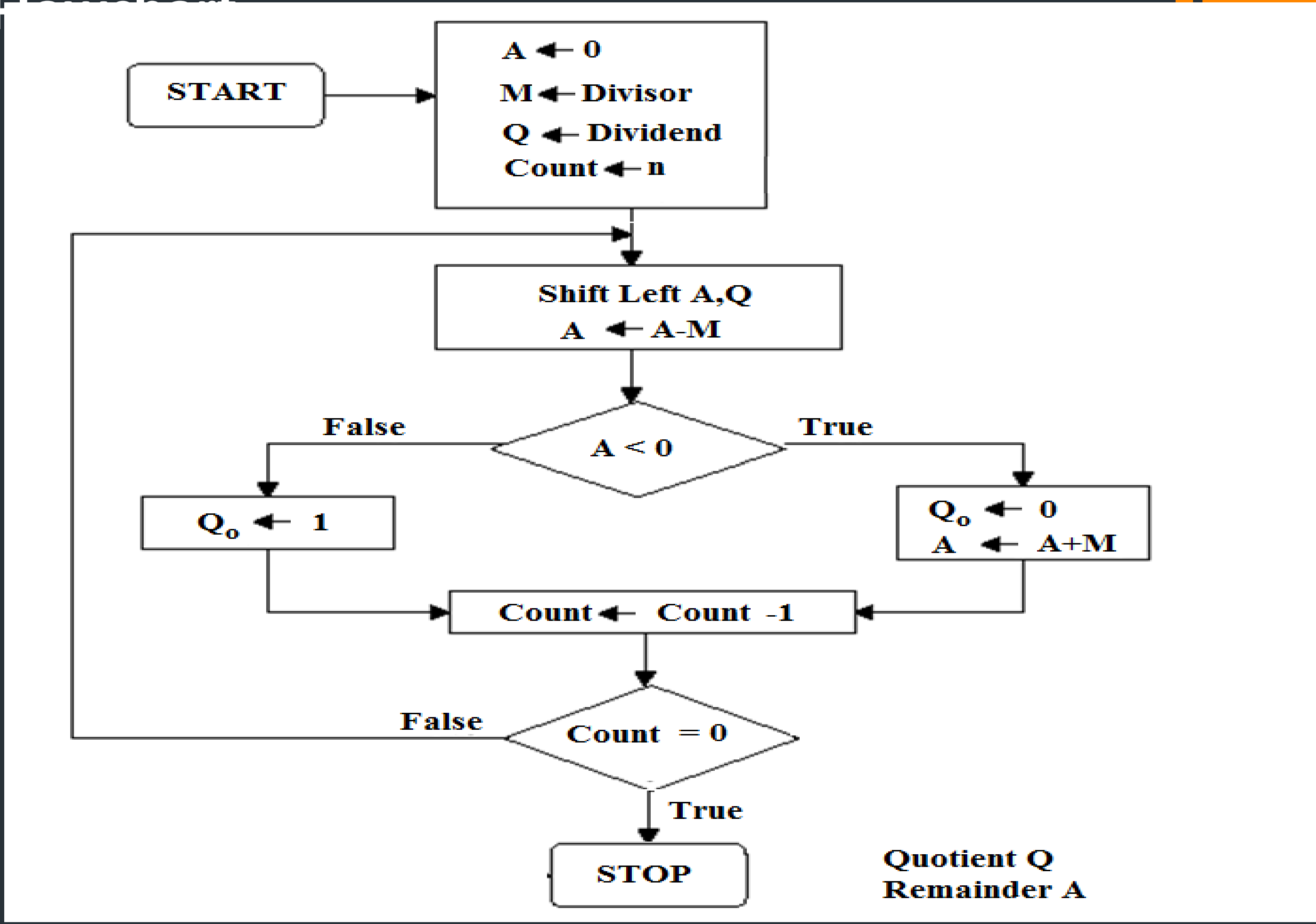
Fixed Point Division

- Restoring Division
- Non-Restoring Division

Restoring Division Algorithm

- Shift A and Q left by one bit binary position
- Subtract M from A, and place the answer back in A
- If the resultant A is negative, set Q_0 to 0 and add M back to A(restore A); Otherwise set Q_0 to 1
- Repeat the above steps for n times

Restoring Division Flowchart



Example 7 ÷ 3

Q=7=0111
M=3=0011

Solution:
Quotient =0010=2
Remainder=0001=1

A	Q		
0000	0111	Initial Values	
0000	111□	Shift left A,Q	①
1101		$A = A - M = 0000 - 0011 = 1101$ $A<0$	
0000	1110	$A = A + M = 1101 + 0011 = 0000$ Restore A	
0001	110□	Shift Left A,Q	②
1110		$A = A - M$	
0001	1100	$A = A + M$	
0011	100□	Shift left A,Q	③
0000		$A = A - M$ $A>0$	
0000	1001	$Q_0 = 1$	
0001	001□	Shift left A,Q	④
1110		$A = A - M$	
0001	0010	$A = A + M$	
Remainder = 1	Quotient = 2		

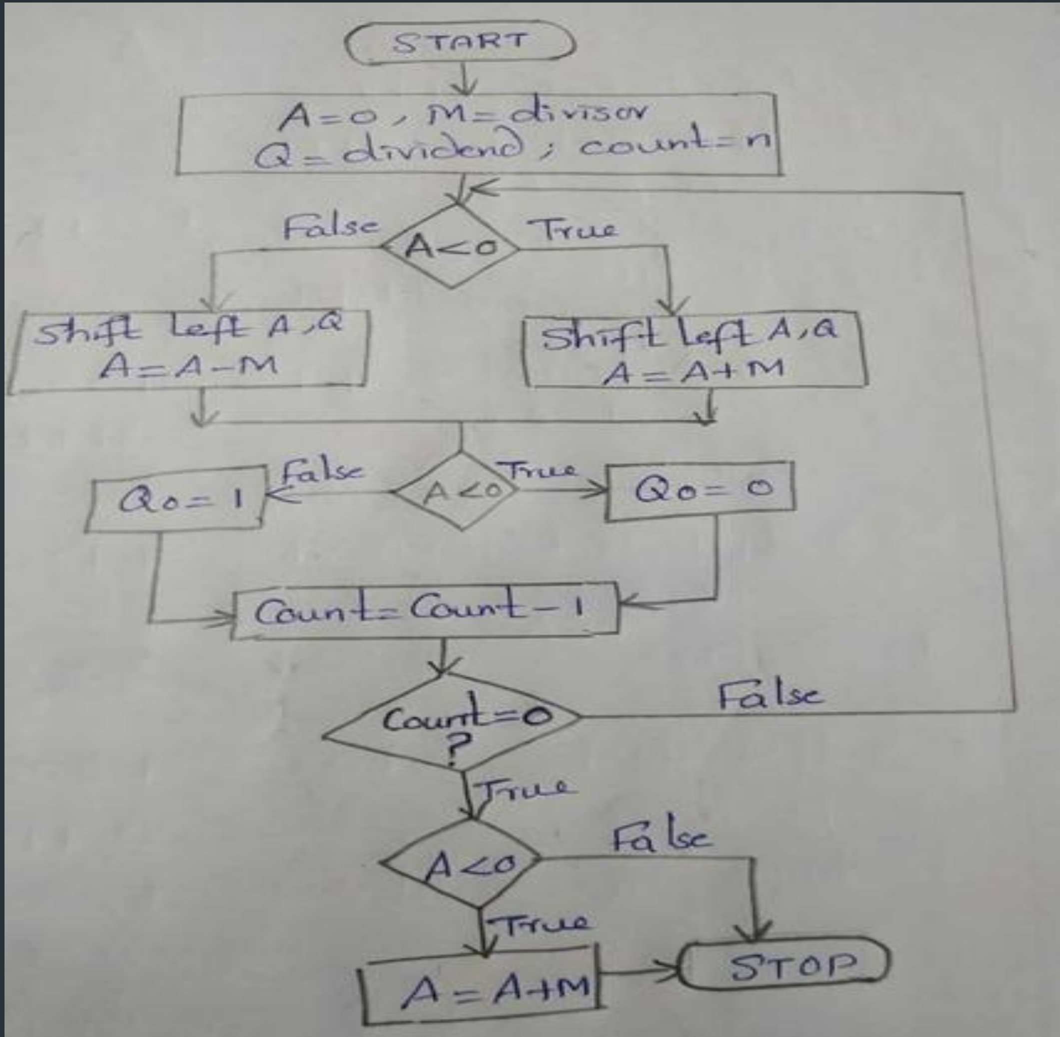
Non-restoring Division

Modifying the basic division algorithm by eliminating restoring step is non-restoring division

Algorithm

1. Start by initializing register A to 0 and repeat steps (2-4) n times
2. If A is positive,
   2.1 Shift A and Q left by one bit position
   2.2 Subtract M from A
3. If A is negative
   3.1 Shift A and Q left by one bit position
   3.2 Add M to A
4. If A is positive, set Q_0 to 1, else Q_0 to 0
5. If A is negative, add M to A as a final corrective step

Non-Restoring Division Flowchart



Example 7 ÷ 3

Q=7=0111
M=3=0011

Solution:
Quotient =0010=2
Remainder=0001=1

Restoring Division	Non-Restoring Division
Needs restoring of register A if resultant of subtraction is negative	Doesn't need restoring
In each cycle content of register A is first shifted left and then divisor is subtracted from it.	In each cycle content of register A is first shifted left and then divisor is added or subtracted with A depending on the sign of A
Does not need restoring of remainder	Needs restoring of remainder if it is negative
Slower algorithm	Faster than restoring method

Booth’s Algorithm

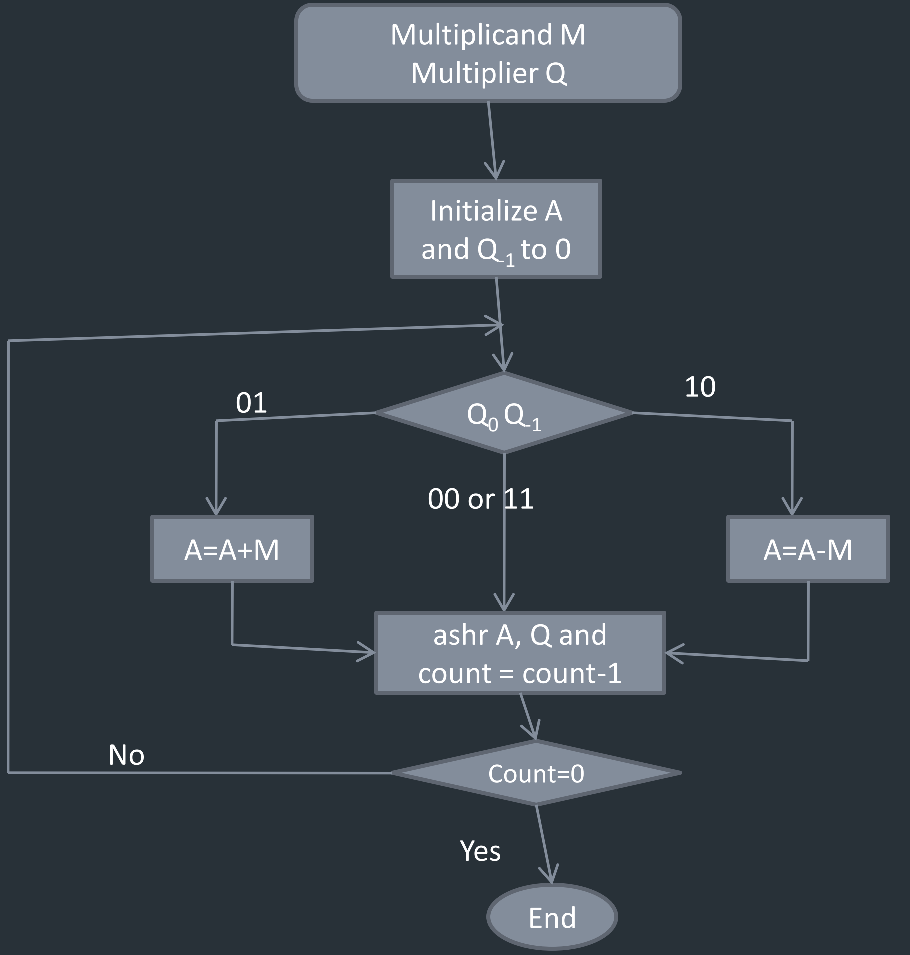
- Booth's algorithm is a multiplication algorithm that multiplies two signed binary numbers in 2's compliment notation.

Procedure:

1. **Let M be the multiplicand and Q be the multiplier**
2. **Consider a 1-bit register Q_{-1} and initialize it to 0**
3. **Consider a register A and initialize it to 0**

Steps for Booth's Algorithm

- If Q_0 and Q_{-1} are same i.e. 00 or 11 then perform right shift by 1 bit
- **If** Q_0 and Q_{-1} is 10, then perform A=A-M and perform right shift by 1 bit
- **If** Q_0 and Q_{-1} is 01, then perform A=A+M and perform right shift by 1 bit


Multiplicand M
Multiplier Q
Initialize A
and Q-1 to 0
Q0 Q-1
01
10
00 or 11
A=A+M
A=A-M
ashr A, Q and
count = count-1
Count=0
No
Yes
End

Multiplication of -12 and -5 using Booth's Algorithm

Multiplicand (M) = -12 $\qquad -12 \times -5 = \underline{60}$

Multiplier (Q) = -5

M = -12 = 11100

In 2's complement → 10100

∴ M = 10100

$\bar{M}+1$ = 10100 → In 2's complement
01011
+1
01100

∴ $\bar{M}+1$ = 01100

Q = -5 = 1101

In 2's complement → 11011

∴ Q = 11011

A	Q	Q_{-1}	Operation	Count
00000	11011	0	Initially	
01100	11011	0	A = A − M	5
00110	01101	1	RS	
00011	00110	1	RS	4
10111	00110	1	A = A + M 00011 10100 10111	3
11011	10011	0	RS	
00111	10011	0	A = A − M 11011 01100 00111	2
00011	11001	1	RS	
00001	11100	1		1
				0

∴ A Q
00001 11100 = $\underline{60}$

Modified Booth’s Algorithm

Multiplication of -12 and -5 Using Modified Booth's alg

-12 is Multiplicand ∴ $Q = -12$

-5 is Multiplier $M = -5$

$Q = -12 \rightarrow 11100$

in 2's Complement form → 10011

Adding Extra one to make total 6 digits → 010100

$\bar{Q}+1 = 001011$

+1

001100

$M = -5 \rightarrow 1101$

in 2's Complement form → 1010

Adding 2 Extra 1's to make total 6 digits → 111011

Step 1: Represent Multiplier in bit-coded pair format

i.e 1110110 ← Adding 0 before LSB of multiplier

$\underbrace{00}_{0}\ \underbrace{-1\,1}_{-1}\ \underbrace{0\,-1}_{-1}$

Step 2: Multiply with Multiplicand.

001100

110000

000000

111100

$(-1)(-12)(2^0) = 12$

$(-1)(-12)(2^2) = 48$

$(0)(-12)(2^4) = 0$

60