SNS COLLEGE OF ENGINEERING

Kurumbapalayam (PO), Coimbatore – 641 107
Accredited by NAAC-UGC with 'A' Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

DEPARTMENT OF INFORMATION TECHNOLOGY

COURSE NAME: 19IT301 COMPUTER ORGANIZATION AND ARCHITECTURE

II YEAR/ III SEM

Unit 2 : ARITHMETIC OPERATIONS

Topic 6: Integer Division

Manual Division

```
        21
     ------
13 ) 274
     26
    ----
     14
     13
    ----
      1
```

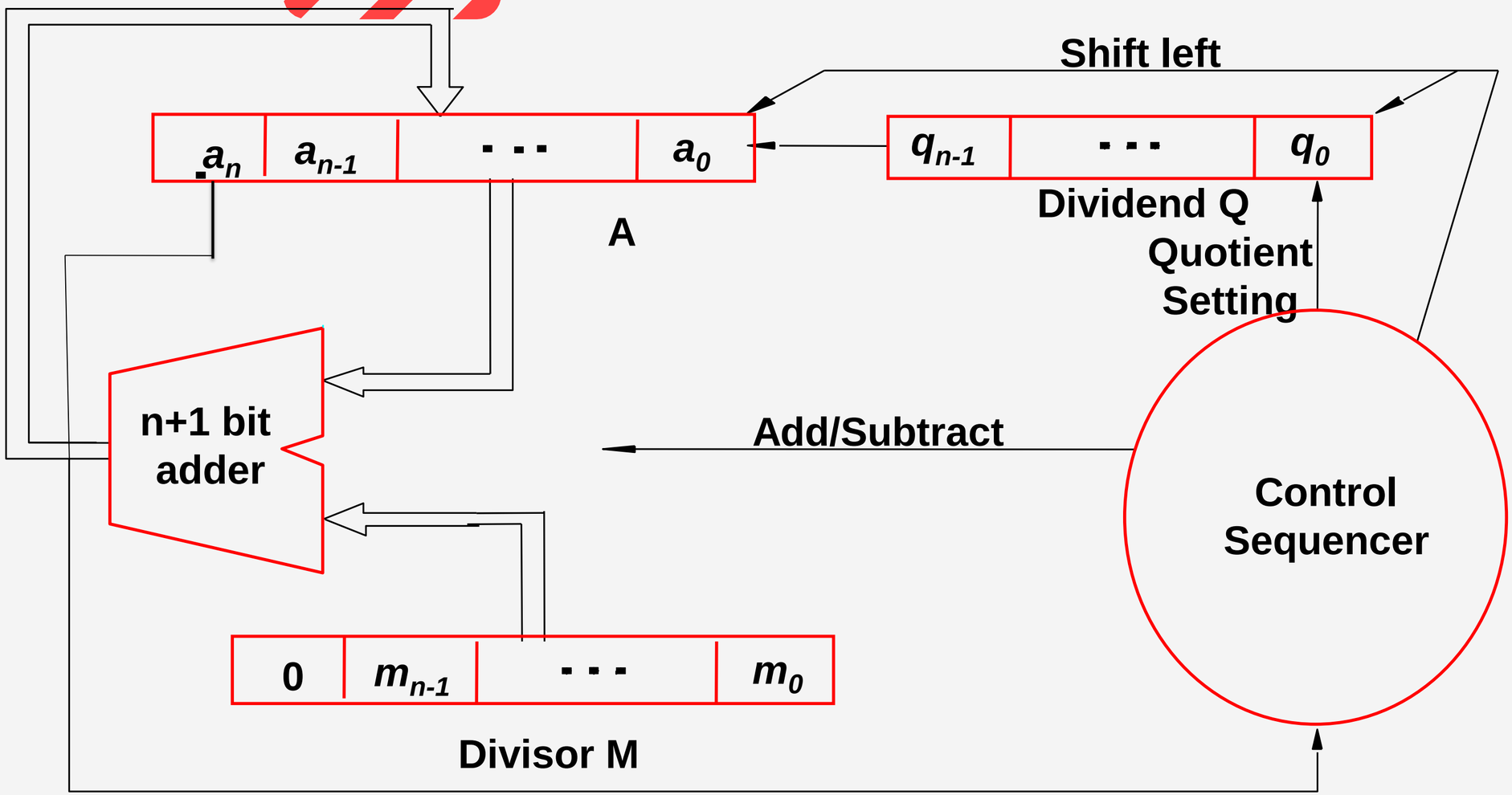
```
                10101   <- Quotient
            ------------
Divisor -> 1101 ) 100010010  <- Dividend
               1101
              -----
               10000
                1101
               -----
                1110
                1101
               -----
                   1   <- Remainder
```

Longhand division examples.

Longhand Division Steps

- Position the divisor appropriately with respect to the dividend and performs a subtraction.
- If the remainder is zero or positive, a quotient bit of 1 is determined, the remainder is extended by another bit of the dividend, the divisor is repositioned, and another subtraction is performed.
- If the remainder is negative, a quotient bit of 0 is determined, the dividend is restored by adding back the divisor, and the divisor is repositioned for another subtraction.

Circuit Arrangement



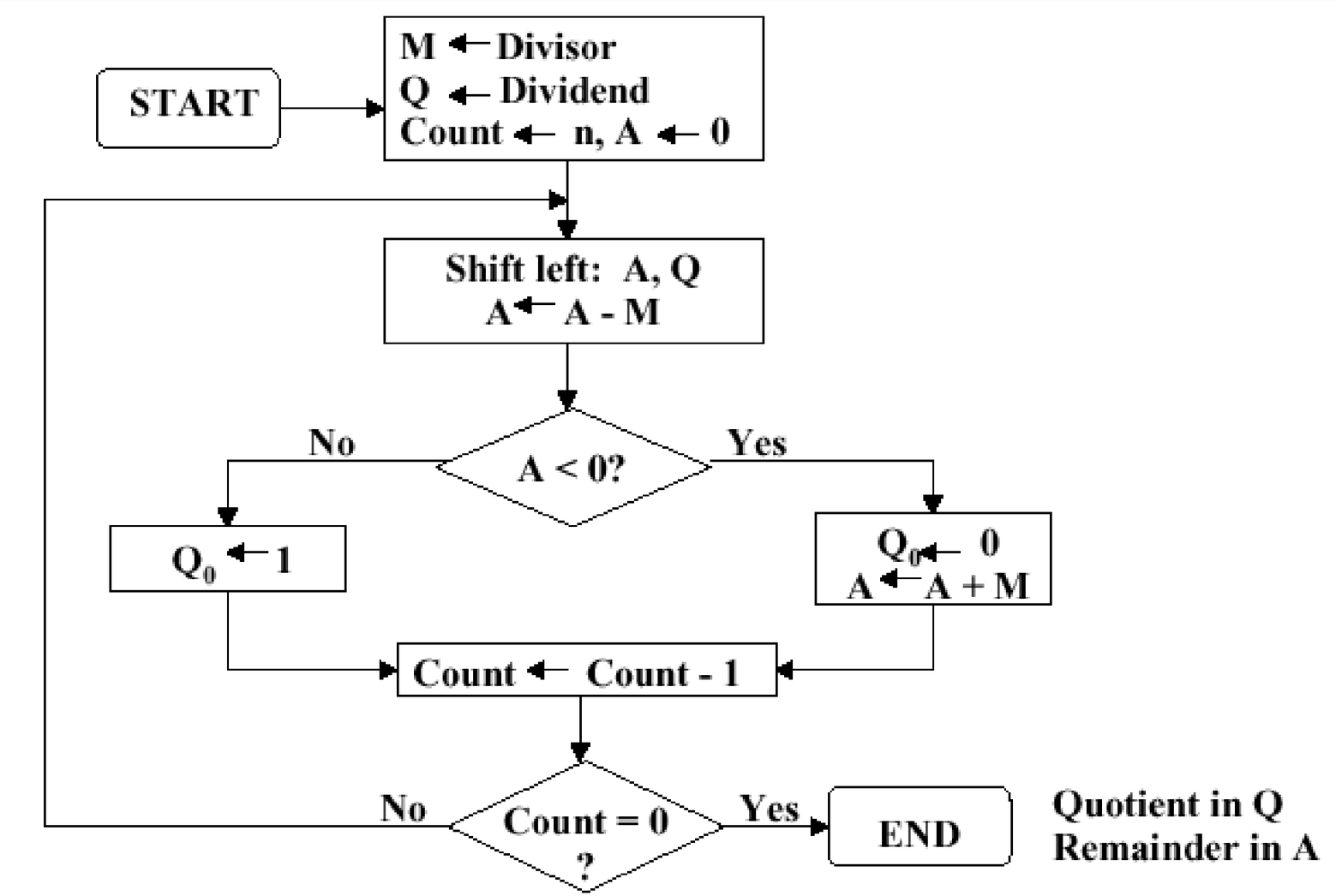
Logic circuit for restoring division

Restoring Division

Algorithm

- Shift A and Q left one binary position
- Perform A - M, and place the answer back in A
- If the sign of A is 1, set q_0 to 0 and add M back to A (restore A); otherwise, set q_0 to 1
- Repeat these steps n times

Restoring division Flowchart



Example: Restoring Division

```
          1 0
      ------------
1 1 ) 1 0 0 0
        1 1
        -----
          1 0
```

	A	Q(Dividend)	Count	
Initially	0 0 0 0 0	1 0 0 0	4	
M	0 0 0 1 1			
Shift	0 0 0 0 1	0 0 0		
Subtract	1 1 1 0 1			
Set q_0	1 1 1 1 0			First cycle
Restore	1 1			3
	0 0 0 0 1	0 0 0 0		
Shift	0 0 0 1 0	0 0 0		
Subtract	1 1 1 0 1			
Set q_0	1 1 1 1 1			Second cycle
Restore	1 1			2
	0 0 0 1 0	0 0 0 0		
Shift	0 0 1 0 0	0 0 0		
Subtract	1 1 1 0 1			
Set q_0	0 0 0 0 1			Third cycle
				1
Shift	0 0 0 1 0	0 0 0 1		
Subtract	1 1 1 0 1	0 0 1		
Set q_0	1 1 1 1 1			Fourth cycle
Restore	1 1			0
	0 0 0 1 0	0 0 1 0		
	Remainder	**Quotient**		

Nonrestoring Division

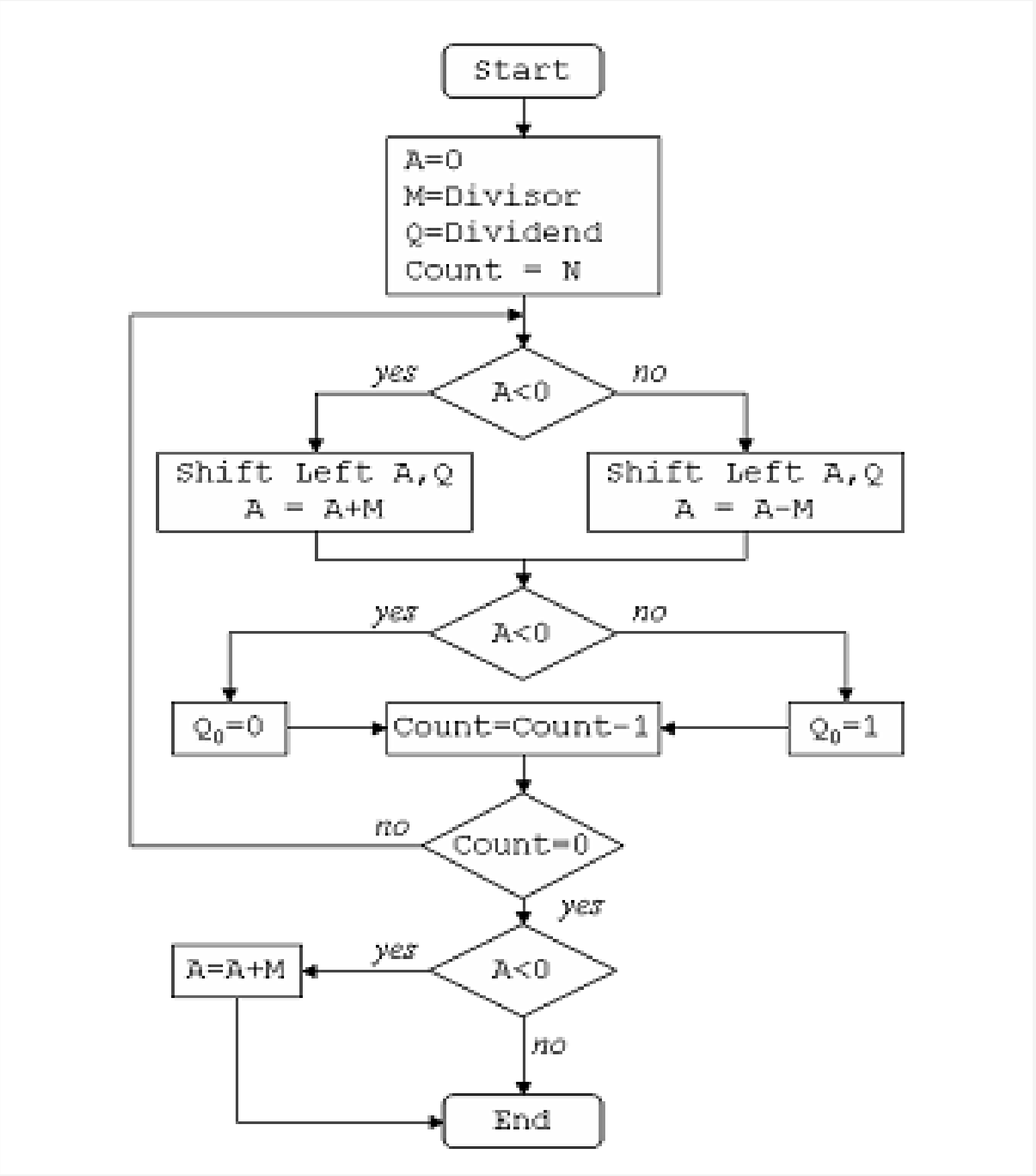
- Avoid the need for restoring A after an unsuccessful subtraction.
- Any idea?

Step 1: (Repeat *n* times)

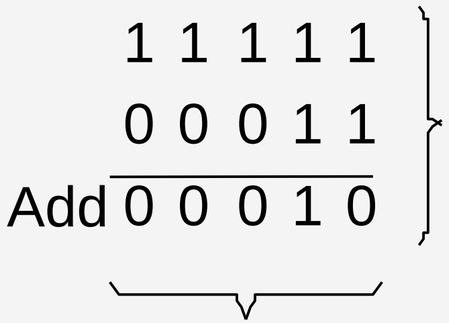
- If the sign of A is 0, shift A and Q left one bit position and subtract M from A; otherwise, shift A and Q left and add M to A.
- Now, if the sign of A is 0, set q_0 to 1; otherwise, set q_0 to 0.

Step2: If the sign of A is 1, add M to A

Nonrestoring division Flowchart



A nonrestoring-division example



Restore remainder

M = 0 0 0 1 1
2's complement of M = 1 1 1 0 1

	A	Q(Dividend)	Count
Initially	0 0 0 0 0	1 0 0 0	4
M	0 0 0 1 1		
Shift	0 0 0 0 1	0 0 0	
Subtract	1 1 1 0 1		
Set q_0	1 1 1 1 0	0 0 0 0	First cycle
Shift	1 1 1 0 0	0 0 0	
Add	0 0 0 1 1		
Set q_0	1 1 1 1 1	0 0 0 0	Second cycle
Shift	1 1 1 1 0	0 0 0	
Add	0 0 0 1 1		
Set q_0	0 0 0 0 1	0 0 0 1	Third cycle
Shift	0 0 0 1 0	0 0 1	
Subtract	1 1 1 0 1		
Set q_0	1 1 1 1 1	0 0 1 0	Fourth cycle

Quotient

Division of signed operands

- No simple algorithms for performing division of signed operands

Solution

- Transform the operands to positive values, use either restoring or non-restoring algorithm
- Transform the result to correct signed values

Exercise

Compute 27/11 using restoring and non-restoring algorithm

Thank You