Computer Organization & Architecture

Chapter 9 – Integer Division

Zhang Yang

cszyang@scut.edu.cn

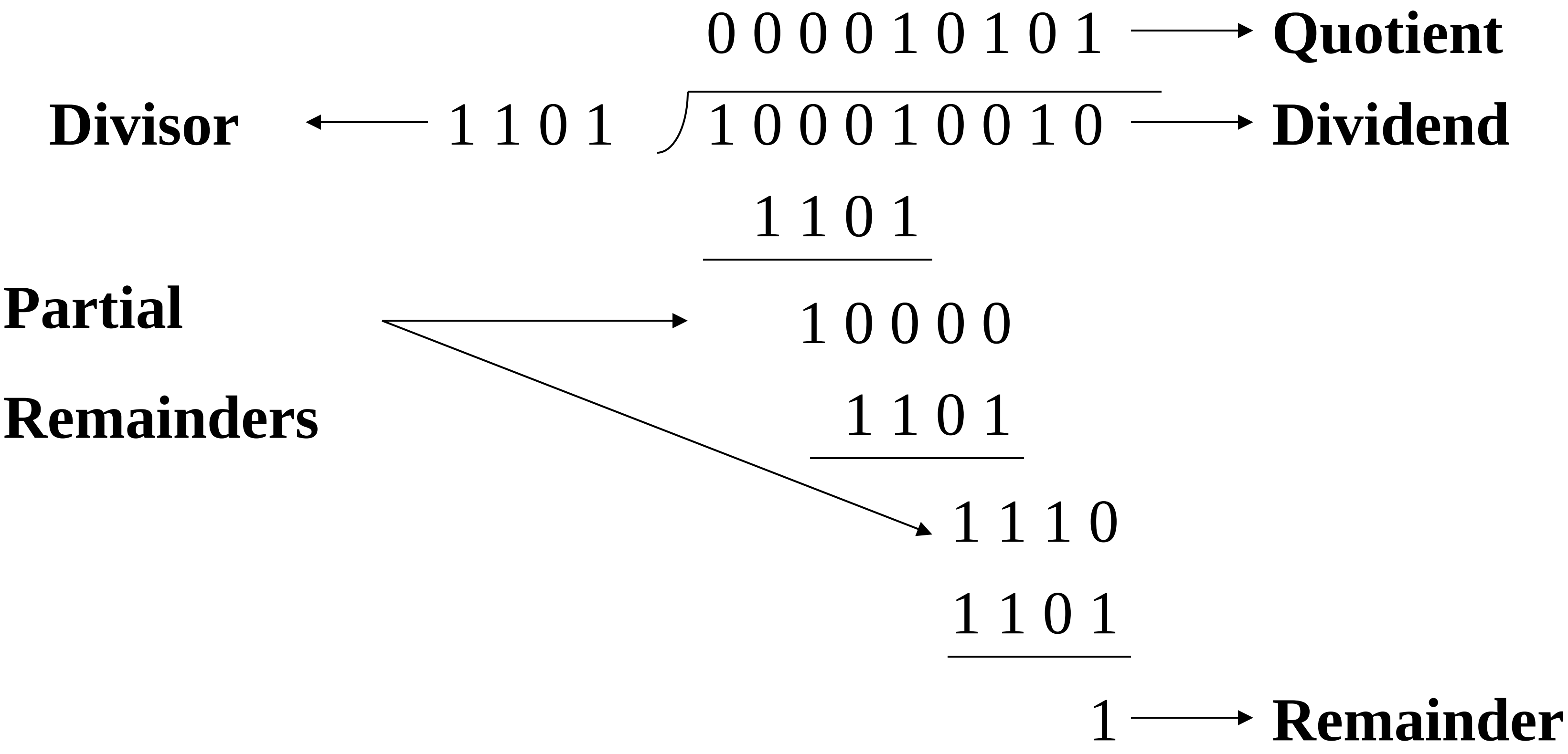
Autumn 2021

Content of this lecture

- 9.6 Integer Division
  - Manual Division
  - Restoring Division
  - Non-Restoring Division
  - Summary

Manual Division (1)

- Example



```
                              0 0 0 0 1 0 1 0 1  ──→  Quotient
                           ___________________
Divisor  ←──  1 1 0 1     ) 1 0 0 0 1 0 0 1 0    ──→  Dividend
                              1 1 0 1
                            _________
Partial                           1 0 0 0 0
                                    1 1 0 1
Remainders                        _________
                                      1 1 1 0
                                      1 1 0 1
                                      _______
                                            1    ──→  Remainder
```

Manual Division (2)

- Process Description
  - The bits of the dividend are examined from left to right, until the set of bits examined represents a number greater than or equal to the divisor. Until this event occurs, 0s are placed in the quotient from left to right.
  - If the event occurs, a 1 is placed in the quotient and the divisor is subtracted from the partial dividend.
  - From this point on, the division follows a cyclic pattern. At each cycle, additional bits from the dividend are appended to the partial remainder until the result is greater than or equal to the divisor. The divisor is subtracted from this number to produce a new partial remainder.
  - Continue until all the bits of the dividend are exhausted.

Content of this lecture

- 9.6 Integer Division
  - Manual Division
  - Restoring Division
  - Non-Restoring Division
  - Summary

Restoring Division (1)

We assume that both the dividend and divisor are positive and hence the quotient and the remainder are positive or zero.

- Hardware

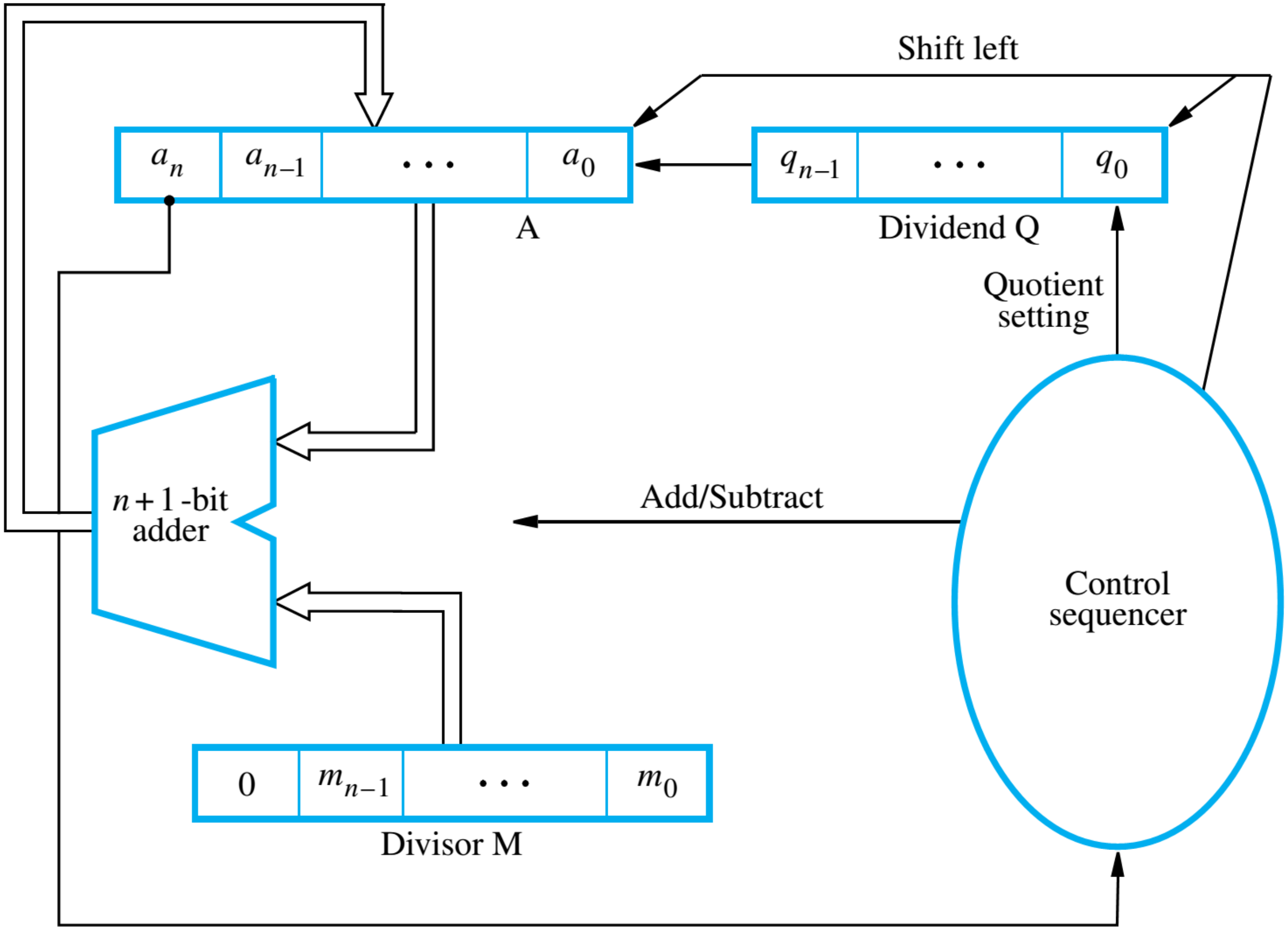
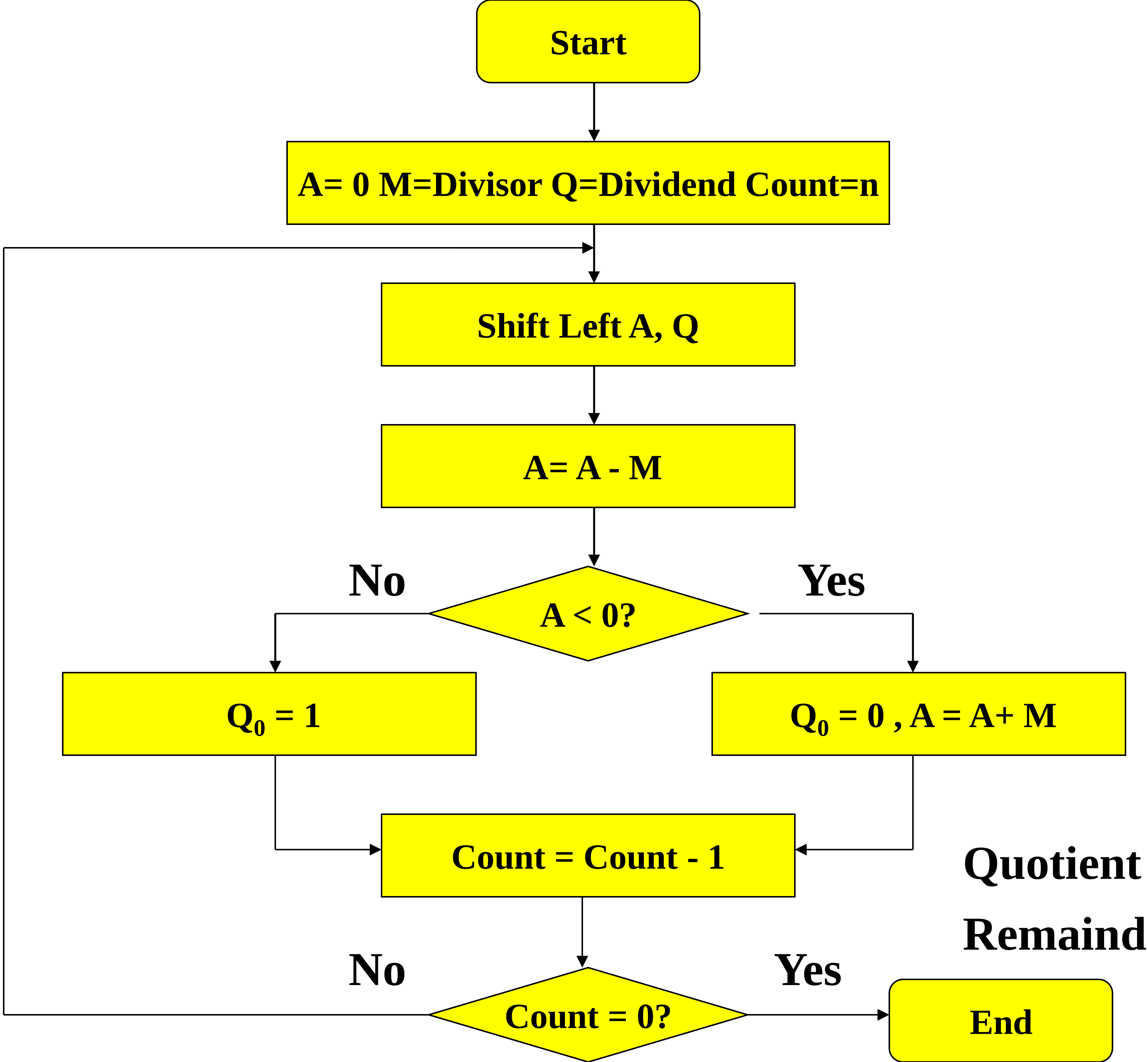


Figure 9.23 Circuit arrangement for binary division.



Start

A= 0 M=Divisor Q=Dividend Count=n

Shift Left A, Q

A= A - M

A < 0?

No

Yes

Q_0 = 1

Q_0 = 0 , A = A+ M

Count = Count - 1

Count = 0?

No

Yes

End

Quotient in Q

Remainder in A

Restoring Division (2)

- Example

```
           1 0
1 1 ⟌ 1 0 0 0
           1 1
        ------
           1 0
```

	M		
Initially	0 0 0 1 1		
	0 0 0 0 0	1 0 0 0	
	A	Q	
Shift	0 0 0 0 1	0 0 0 □	First Cycle
Subtract	1 1 1 0 1		
Set q_0	1 1 1 1 0		
Restore	0 0 0 1 1		
	0 0 0 0 1	0 0 0 0	

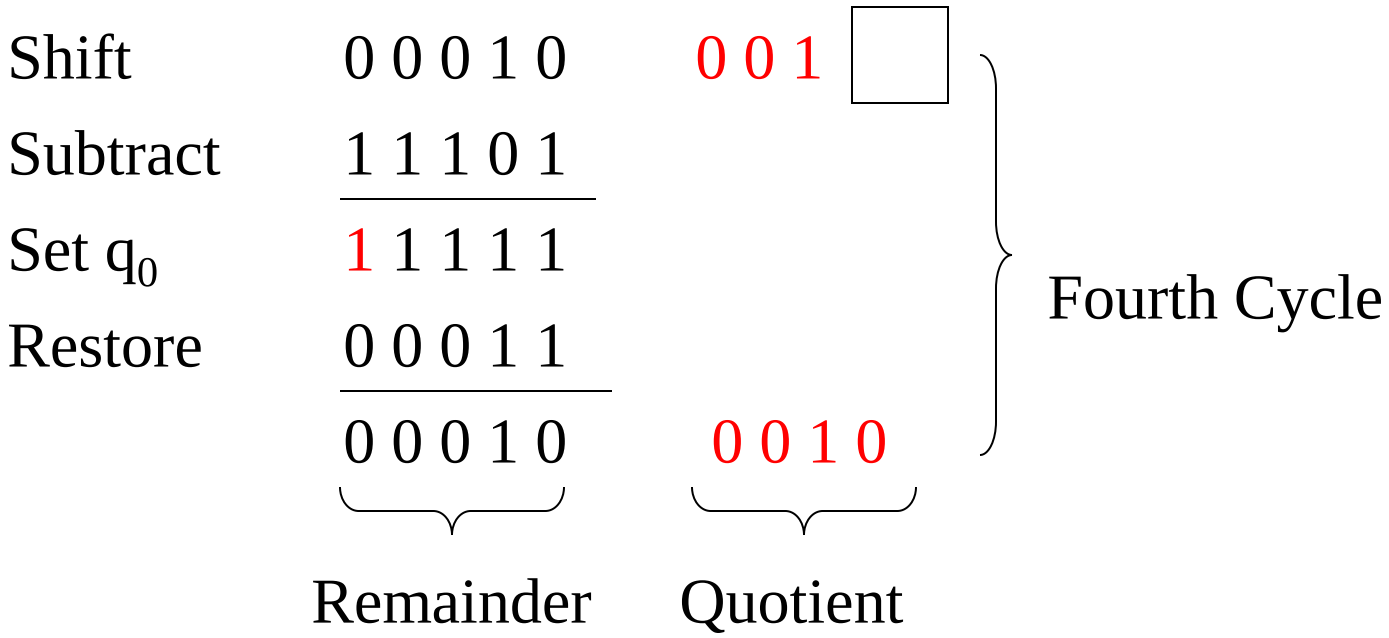
Restoring Division (3)

- Example (ctd.)

Shift	0 0 0 1 0	0 0 0 □	Second Cycle
Subtract	1 1 1 0 1		
Set q_0	1 1 1 1 1		
Restore	0 0 0 1 1		
	0 0 0 1 0	0 0 0 0	
Shift	0 0 1 0 0	0 0 0 □	Third Cycle
Subtract	1 1 1 0 1		
Set q_0	0 0 0 0 1	0 0 0 1	

Restoring Division (4)

- Example (ctd.)



Shift	0 0 0 1 0	0 0 1 □
Subtract	1 1 1 0 1	
Set q_0	1 1 1 1 1	
Restore	0 0 0 1 1	
	0 0 0 1 0	0 0 1 0
	Remainder	Quotient

Fourth Cycle

Content of this lecture

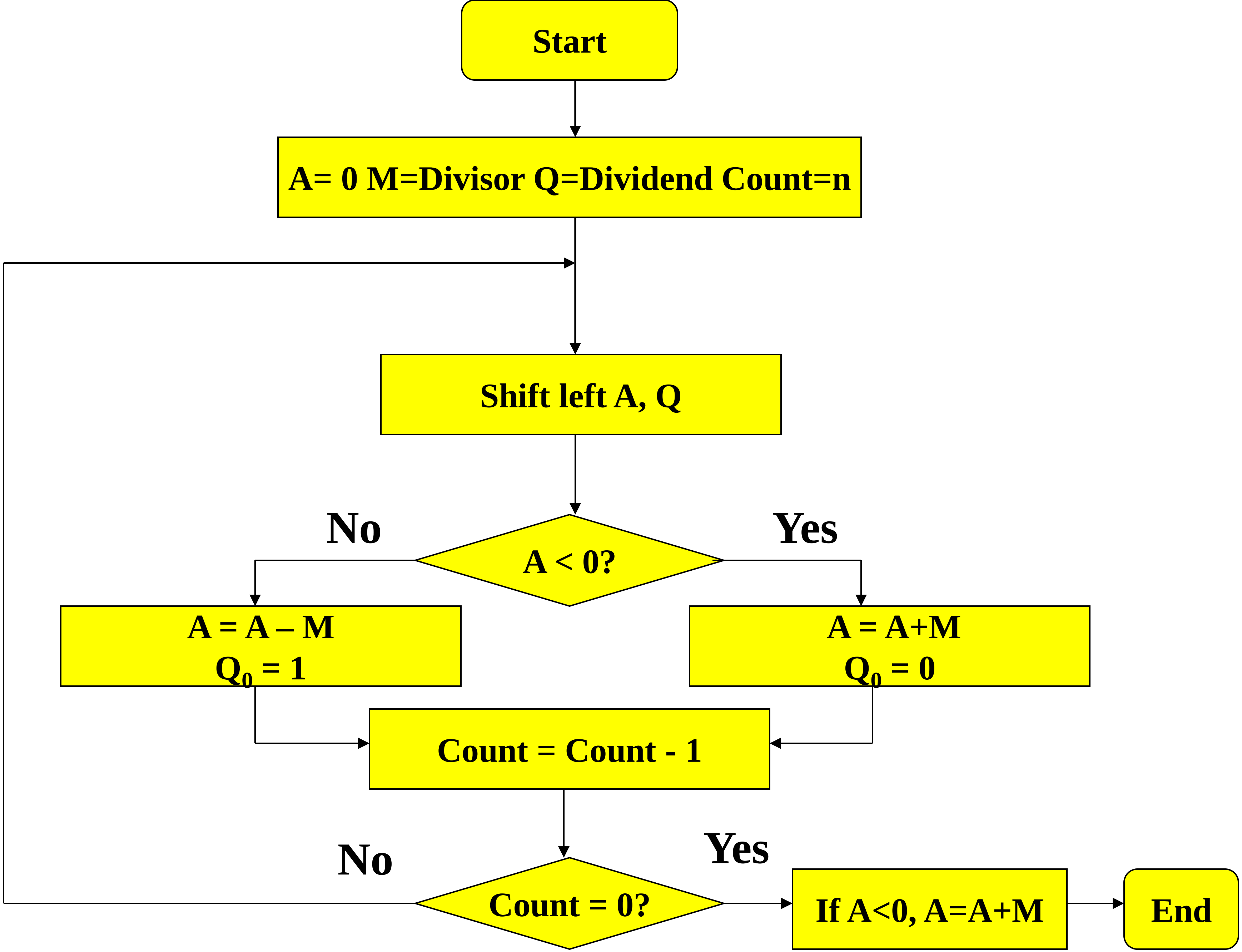
- 9.6 Integer Division
    - Manual Division
    - Restoring Division
    - Non-Restoring Division
    - Summary

Non-Restoring Division (1)

- Disadvantage of Restoring Division
  - Restoring the partial remainder increases the execution time of the division operation, since on average the restoring is executed in 50% of the cases.
- Non-Restoring Division
  - Avoid the need for restoring the partial remainder after an unsuccessful subtraction.

Non-Restoring Division (2)

- In restoring division
  - □ *(i – 1)* step, $R_{i\ -1}$ (A= $R_{i\ -1}$)
  - □ *i* step, A = $2R_{i\ -1} – M$
    - $A > 0, Q_0 = 1, R_i = A$
    - $A < 0, Q_0 = 0, R_i = A + M$
  - □ *(i + 1)* step, A = $2R_i – M$
    - R_i = A, A = 2Ri– M = 2A – M
    - R_i = A + M, A = 2Ri– M = 2(A + M) – M= 2A + M



Start

A= 0 M=Divisor Q=Dividend Count=n

Shift left A, Q

A < 0?

No

Yes

A = A – M
Q_0 = 1

A = A+M
Q_0 = 0

Count = Count - 1

Count = 0?

No

Yes

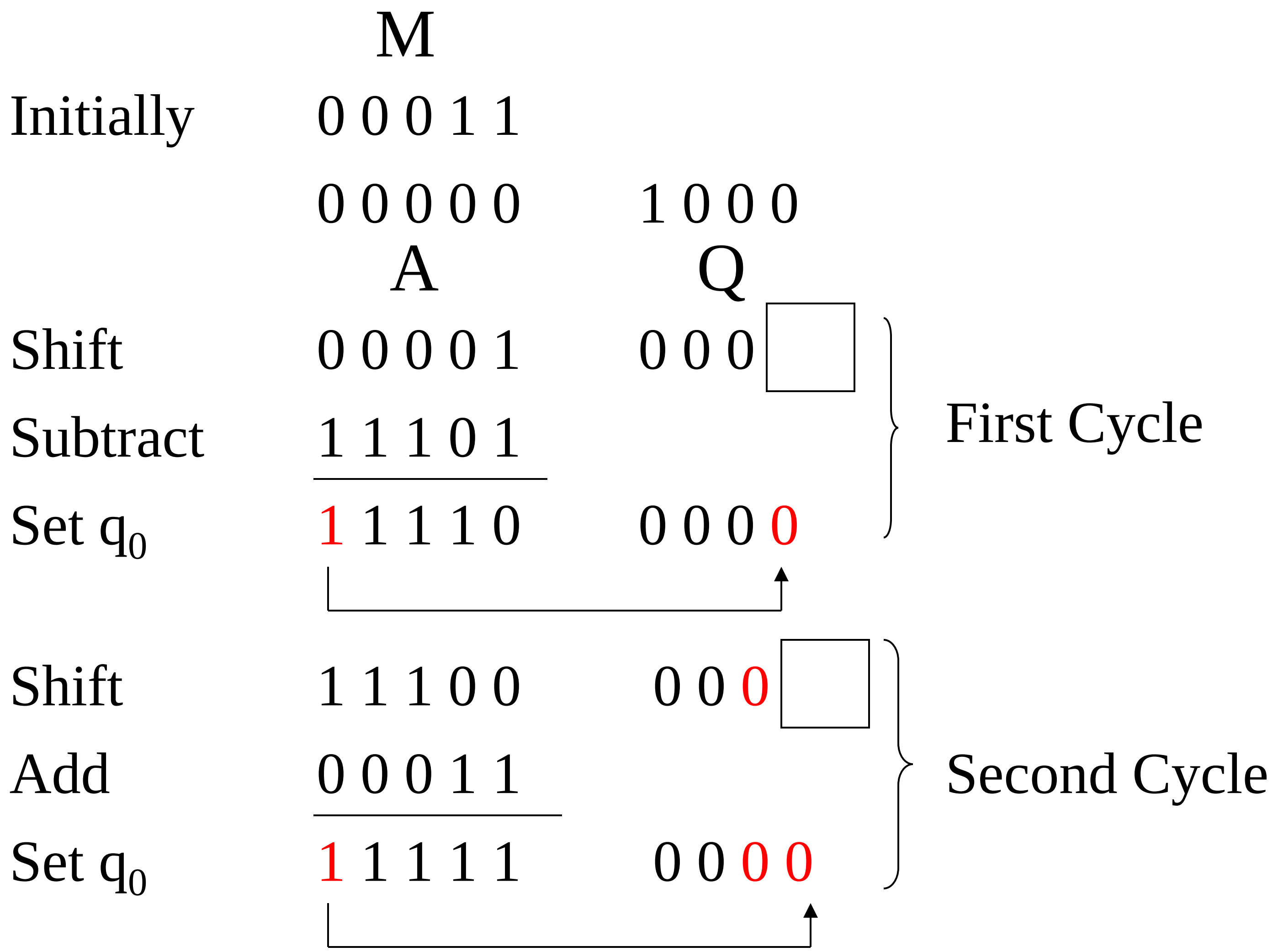
If A<0, A=A+M

End

Non-Restoring Division (3)

- Example

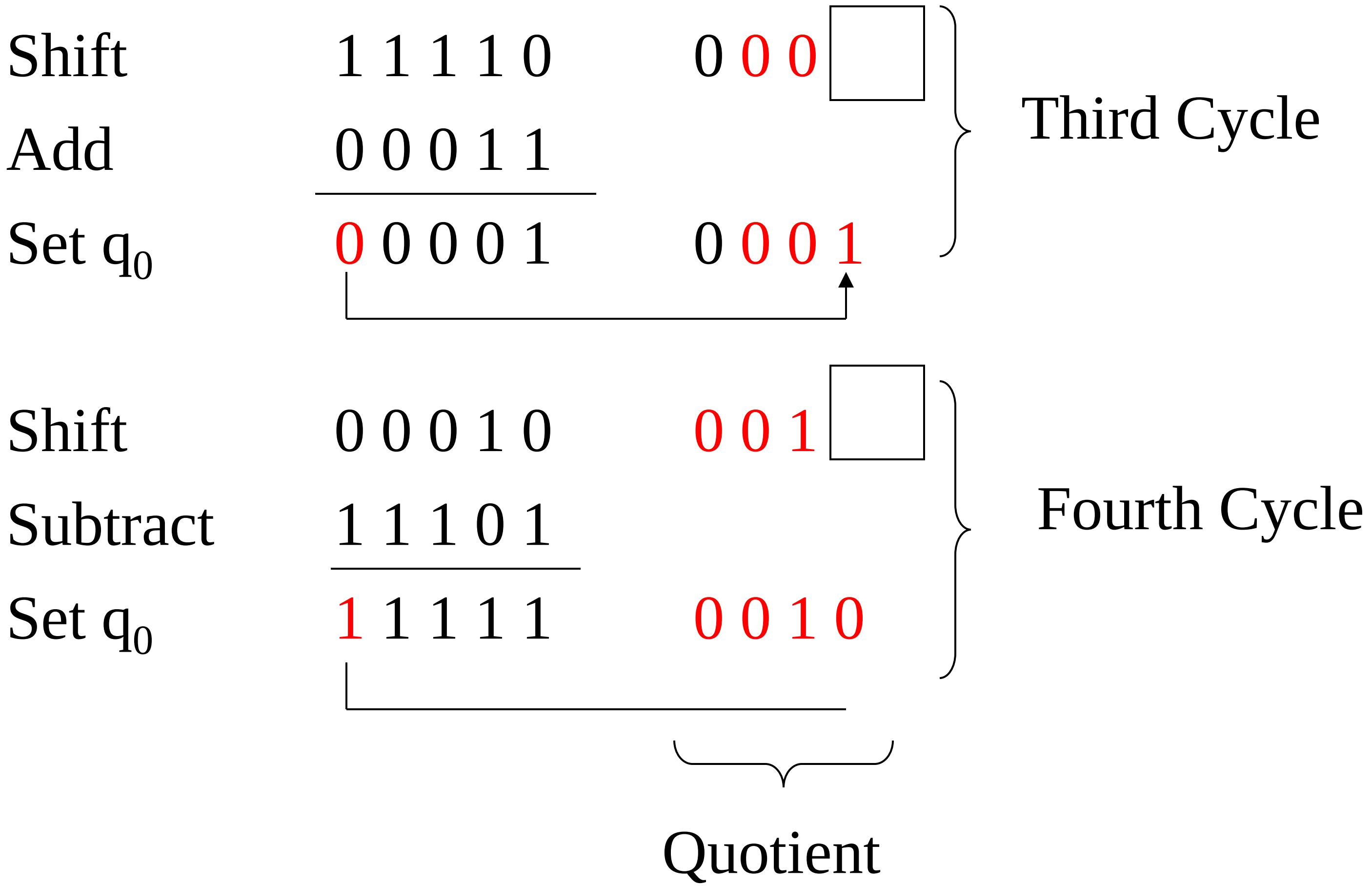
```
          1 0
  1 1 ) 1 0 0 0
          1 1
        -----
            1 0
```



	M		
Initially	0 0 0 1 1		
	0 0 0 0 0	1 0 0 0	
	A	Q	
Shift	0 0 0 0 1	0 0 0 □	
Subtract	1 1 1 0 1		First Cycle
Set q_0	1 1 1 1 0	0 0 0 0	
Shift	1 1 1 0 0	0 0 0 □	
Add	0 0 0 1 1		Second Cycle
Set q_0	1 1 1 1 1	0 0 0 0	

Non-Restoring Division (4)

- Example (ctd.)



Shift	1 1 1 1 0	0 0 0	Third Cycle
Add	0 0 0 1 1		
Set q_0	0 0 0 0 1	0 0 0 1	
Shift	0 0 0 1 0	0 0 1	Fourth Cycle
Subtract	1 1 1 0 1		
Set q_0	1 1 1 1 1	0 0 1 0	

Quotient

Non-Restoring Division (5)

- Example (ctd.)

$$\left.\begin{array}{lr} \text{Add} & 1\ 1\ 1\ 1\ 1 \\ & 0\ 0\ 0\ 1\ 1 \\ \hline & \underbrace{0\ 0\ 0\ 1\ 0}_{\text{Remainder}} \end{array}\right\} \begin{array}{l}\text{Restoring}\\ \text{Remainder}\end{array}$$

Non-Restoring Division (6)

- Note
  - There are no simple algorithms for directly performing division on signed operands that are comparable to the algorithms for signed multiplication.
  - In division, the operands can be processed to transform them into positive values.
  - After using one of the algorithms (restoring division or nonrestoring division), the results are transformed to the correct signed values, as necessary.

Summary

- 知识点 Integer Division
  - Restoring division
  - Nonrestoring division
- 掌握程度
  - 给定被除数和除数，使用不恢复余数除法计算出商和余数。

Homework

- P377 9.1,
- P380 9.20
- P379 9.9 (a) (b)