

Computer Organization & Architecture  
**Chapter 9 – Integer Division**

Zhang Yang

`cszyang@scut.edu.cn`

Autumn 2025



# Content of this lecture

## ■ 9.6 Integer Division

- Manual Division
- Restoring Division
- Non-Restoring Division
- Summary

# Manual Division (1)

## ■ Example

The diagram illustrates the manual division of the binary number 100010010 by 1101. The divisor is 1101, and the dividend is 100010010. The quotient is 000010101, and the remainder is 1. The process involves subtracting the divisor from the dividend to find the remainder.

**Divisor** ← 1 1 0 1

0 0 0 0 1 0 1 0 1 → **Quotient**

1 0 0 0 1 0 0 1 0 → **Dividend**

1 1 0 1

**Partial** → 1 0 0 0 0

**Remainders** → 1 1 0 1

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

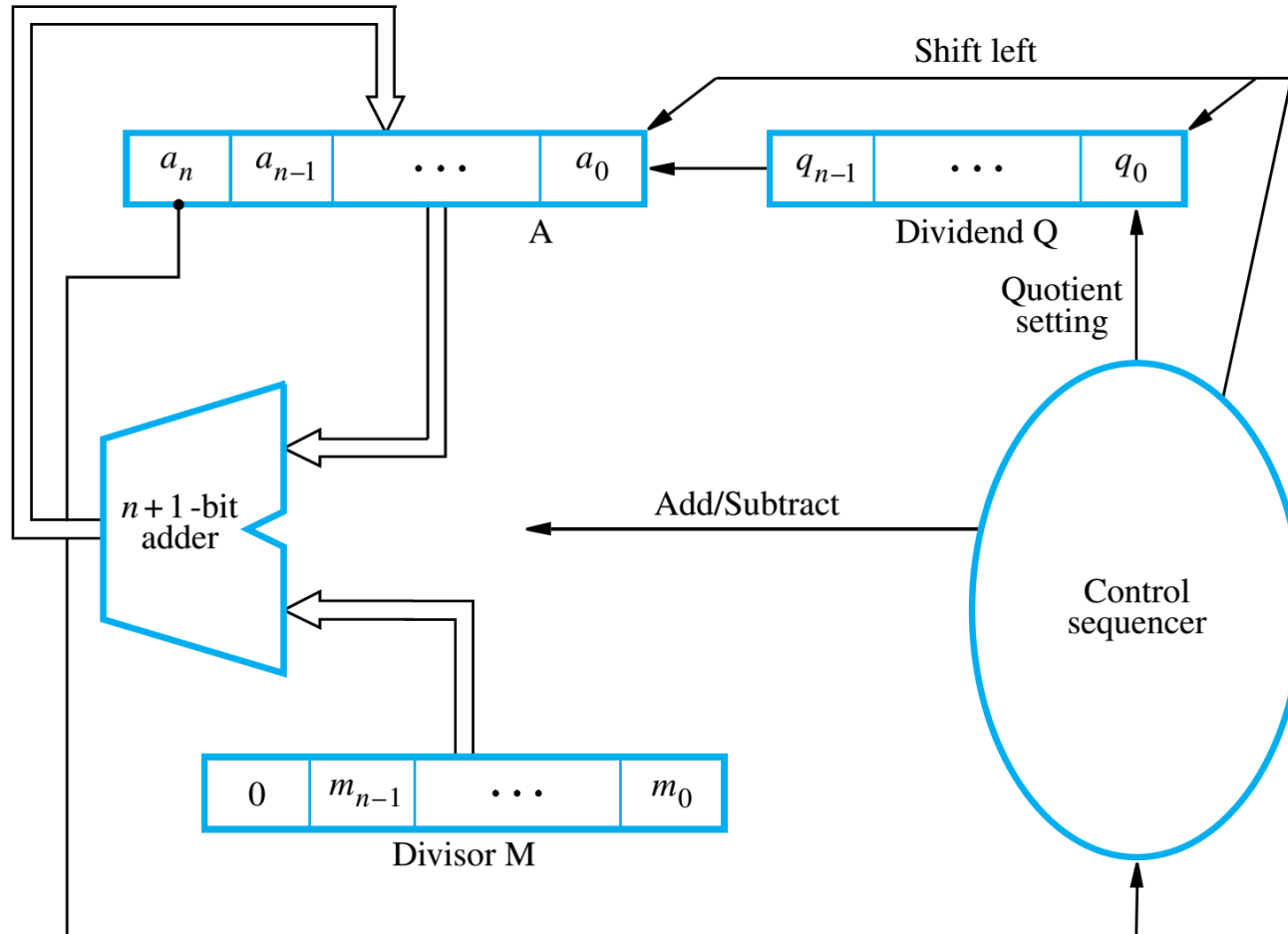
## ■ 9.6 Integer Division

- ☐ Manual Division
- ☐ Restoring Division
- ☐ Non-Restoring Division
- ☐ Summary

# Restoring Division (1)

## ■ Hardware

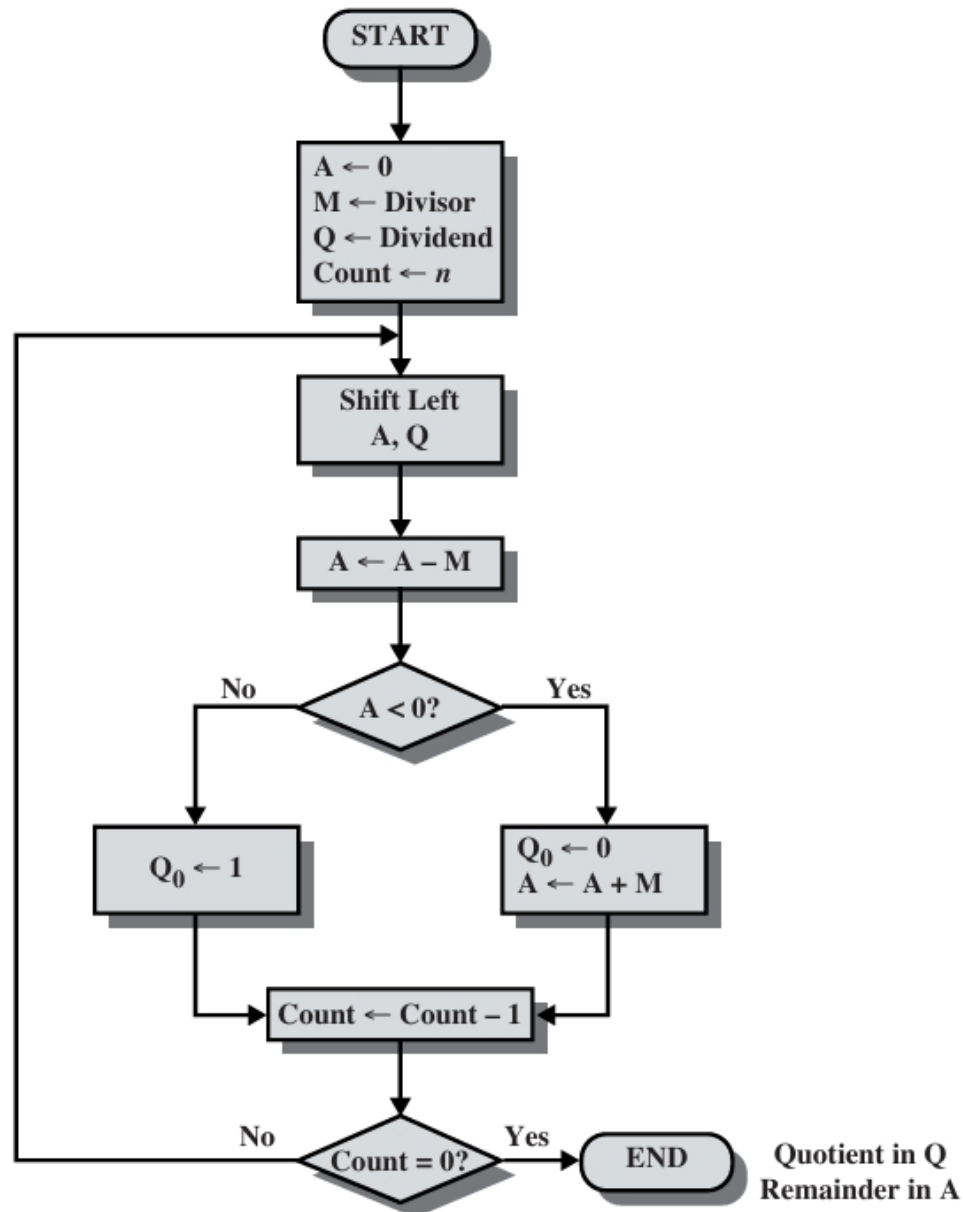
We assume that both the dividend and divisor are positive and hence the quotient and the remainder are positive or zero.



**Figure 9.23** Circuit arrangement for binary division.

# Restoring Division (2)

## ■ Flowchart



# Restoring Division (3)

## ■ Example

$  \begin{array}{r}  10 \\  11 \overline{) 1000} \\  \underline{11} \\  10  \end{array}  $	Initially	$  \begin{array}{r}  M \\  00011 \\  00000  \end{array}  $	1000
	Shift	$  \begin{array}{r}  A \\  00001  \end{array}  $	$  \begin{array}{r}  Q \\  000 \boxed{\phantom{0}}  \end{array}  $
	Subtract	$  \begin{array}{r}  11101 \\  \hline  \end{array}  $	
	Set $q_0$	$  \begin{array}{r}  \color{red}{1}1110 \\  \hline  \end{array}  $	
	Restore	$  \begin{array}{r}  00011 \\  \hline  00001  \end{array}  $	000 $\color{red}{0}$

First Cycle

# Restoring Division (4)

## ■ Example (ctd.)

Shift	0 0 0 1 0	0 0 <b>0</b> <span style="border: 1px solid black; display: inline-block; width: 20px; height: 20px; vertical-align: middle;"></span>	} Second Cycle
Subtract	<u>1 1 1 0 1</u>		
Set $q_0$	<b>1</b> 1 1 1 1		
Restore	<u>0 0 0 1 1</u>		
	0 0 0 1 0	0 0 <b>0 0</b>	
Shift	0 0 1 0 0	0 <b>0 0</b> <span style="border: 1px solid black; display: inline-block; width: 20px; height: 20px; vertical-align: middle;"></span>	} Third Cycle
Subtract	<u>1 1 1 0 1</u>		
Set $q_0$	<b>0</b> 0 0 0 1	0 <b>0 0 1</b>	

# Restoring Division (5)

## ■ Example (ctd.)

Shift	0 0 0 1 0	0 0 1	<div></div>	}	Fourth 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 1 0			
	<div></div>	<div></div>			
	Remainder	Quotient			



# 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.

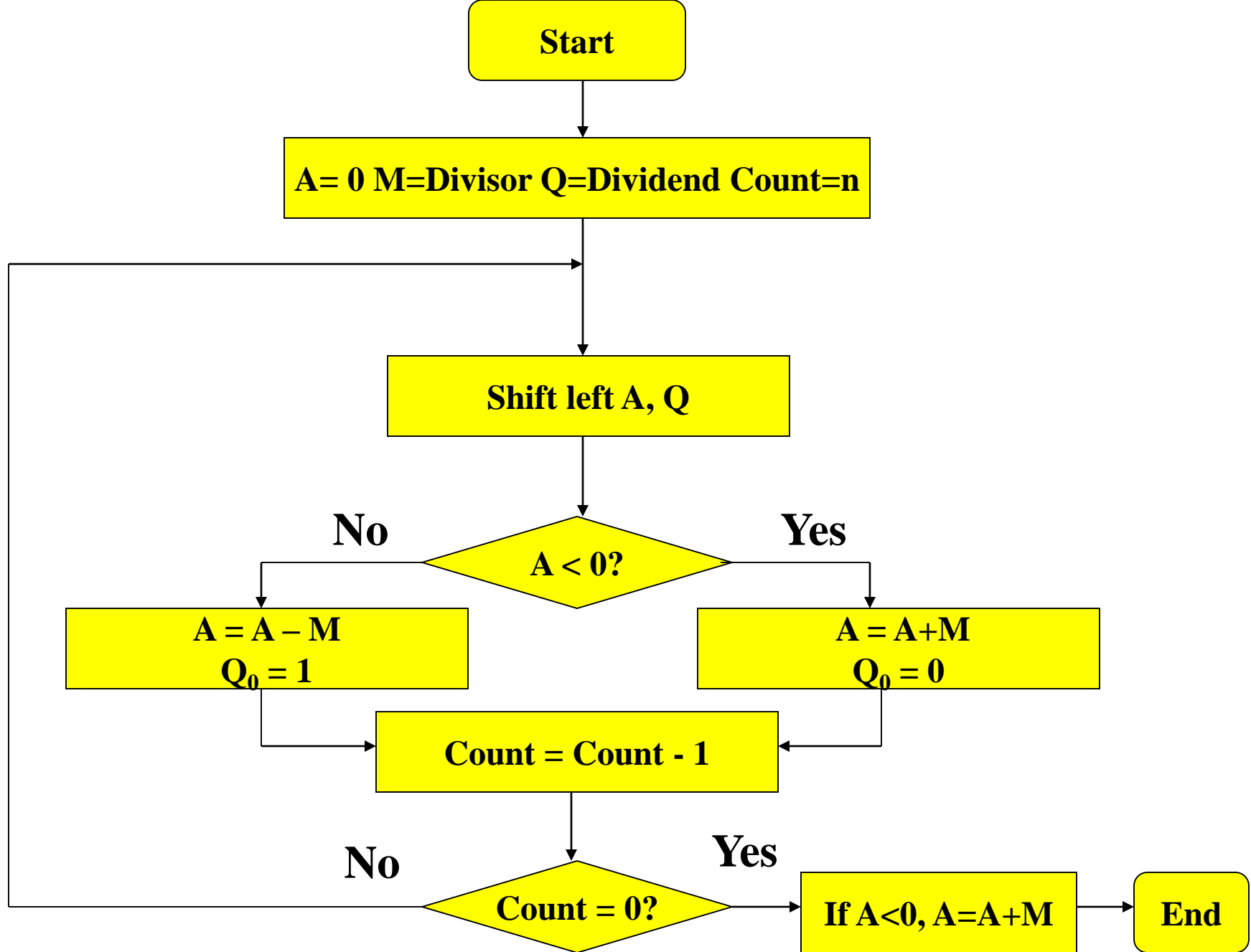
## ■ Non-Restoring Division

- Avoid the need for restoring the partial remainder after an unsuccessful subtraction.

# Non-Restoring Division (2)

- In restoring division

- If  $A$  is positive, we shift left and subtract  $M$ , that is, we perform  $2A - M$ .
- If  $A$  is negative, we restore it by performing  $A + M$ , and then we shift it left and subtract  $M$ . This is equivalent to performing  $2A + M$ .



# Non-Restoring Division (3)

## ■ Example

$$\begin{array}{r}
 10 \\
 11 \overline{) 1000} \\
 \underline{11} \\
 10
 \end{array}$$

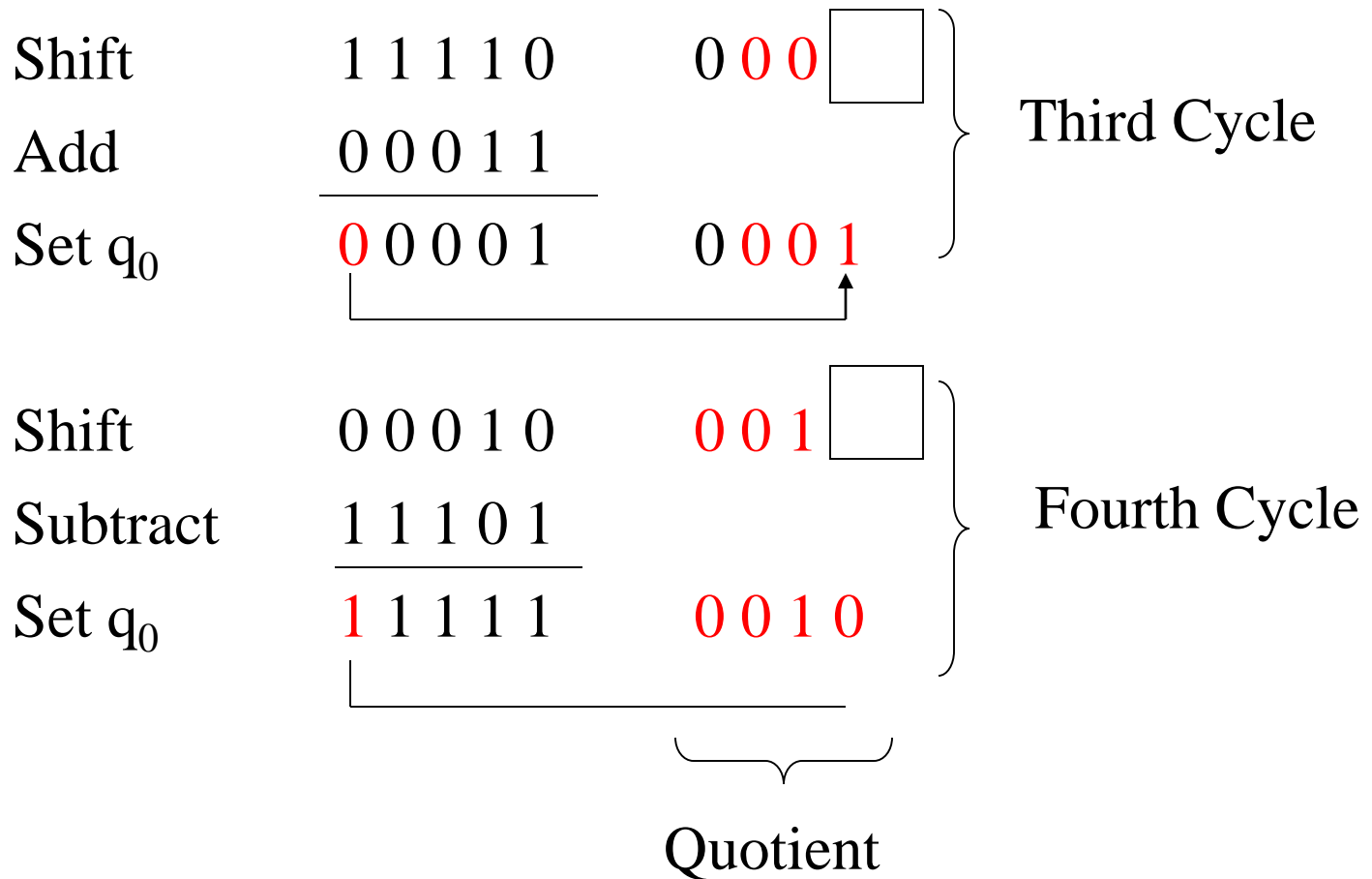
	M	
Initially	0 0 0 1 1	
	0 0 0 0 0	1 0 0 0
	A	Q
Shift	0 0 0 0 1	0 0 0 <span style="border: 1px solid black; display: inline-block; width: 20px; height: 20px; vertical-align: middle;"></span>
Subtract	1 1 1 0 1	
Set $q_0$	<span style="color: red;">1</span> 1 1 1 0	0 0 0 <span style="color: red;">0</span>
	<div style="display: flex; align-items: center; justify-content: center;"> <div style="border-top: 1px solid black; width: 100%;"></div> <div style="margin-left: 5px;">↑</div> </div>	
Shift	1 1 1 0 0	0 0 <span style="color: red;">0</span> <span style="border: 1px solid black; display: inline-block; width: 20px; height: 20px; vertical-align: middle;"></span>
Add	0 0 0 1 1	
Set $q_0$	<span style="color: red;">1</span> 1 1 1 1	0 0 <span style="color: red;">0</span> <span style="color: red;">0</span>
	<div style="display: flex; align-items: center; justify-content: center;"> <div style="border-top: 1px solid black; width: 100%;"></div> <div style="margin-left: 5px;">↑</div> </div>	

First Cycle

Second Cycle

# Non-Restoring Division (4)

## ■ Example (ctd.)



# Non-Restoring Division (5)

## ■ Example (ctd.)

Add	1 1 1 1 1	}	Restoring Remainder
	0 0 0 1 1		
	<hr/> 0 0 0 1 0		
	<u>          </u>		
	Remainder		

# 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

- P379 9.9 (a) (b)
- P380 9.20