

# 第 3 章 运算器与运算方法

本章学习导读：

- ( 1 ) 运算器由哪几部分组成？
- ( 2 ) 如何实现加、减、乘、除和移位等操作？
- ( 3 ) 为了提高运算器速度采取了哪些措施？

**运算器**：是计算机中的执行部件，

1. 对二进制数据进行各种**算术和逻辑运算**；
2. **CPU 内部数据信息的重要通路。**

## 3.1 运算器的基本组成

1. 算术逻辑单元 ALU
2. 通用寄存器组
3. 专用寄存器
4. 附加的控制线路

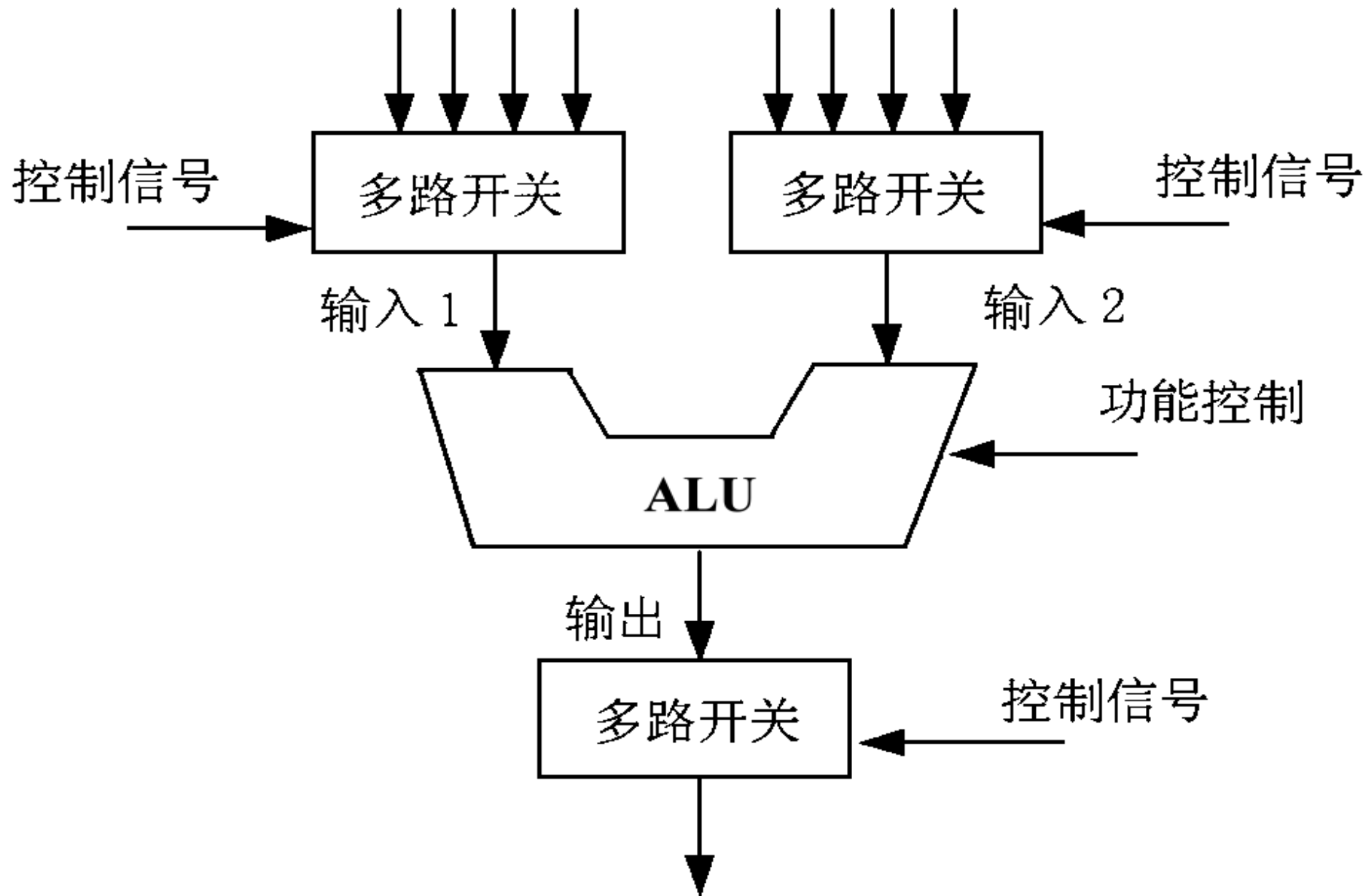
**重点：**算术逻辑单元的组成与工作原理；

# 1. 算术逻辑单元 ALU

算术逻辑单元 ALU(Arithmetic and Logic Unit) :

- 1) 运算器的核心；
- 2) 对数据进行加工处理：数值数据的算术运算和逻辑数据的逻辑操作。

ALU 的逻辑符号表示：通常表示为具有两个输入端口，一个输出端口和多个功能控制信号端的逻辑符号。



ALU 的逻辑符号表示与多路开关



## 2. 通用寄存器组

通用寄存器组：运算器内设置的若干通用寄存器；

- 1) 用于暂时存放参加运算的数据和某些中间结果。
- 2) 通用寄存器的数量越多，对提高运算器性能和程序执行速度越有利。
- 3) 通用寄存器组是对用户开放的，用户可以通过指令去使用这些寄存器。

累加器：在运算器中用来提供一个操作数并存放运算结果的通用寄存器。

如：ADD A, R<sub>i</sub>

### 3. 专用寄存

器

运算器中设置专用寄存器：

- 1) 记录运算器现行指令执行过程中的重要状态；
- 2) 提供运算前后数据的暂存缓冲等（透明）。
- 3) 堆栈指针 SP(Stack Pointer)：指示了堆栈的使用情况。

程序状态字 PSW(Program Status Word)：存放指令执行结果的某些状态，

- 1) 一条指令执行后，根据运算结果自动修改；
- 2) 如是否溢出 O、是否为零 Z、是否有进位 C、是否为负 N、奇偶位 P 等。
- 3) 对程序员是开放的。

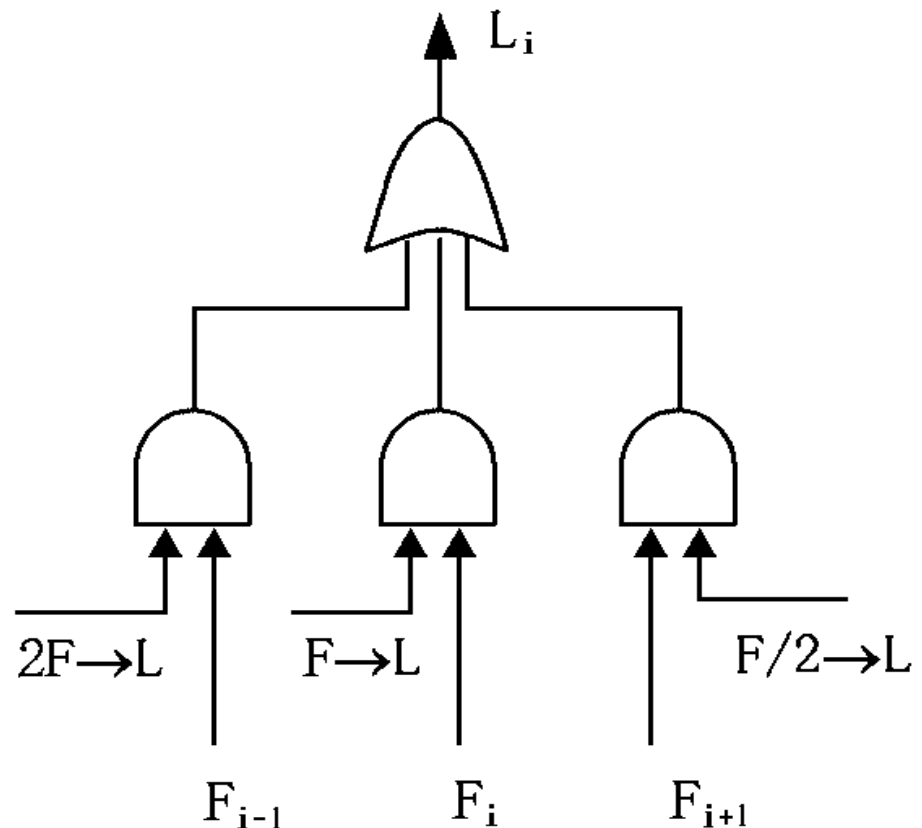
## 4. 附加的控制线路

在运算器中附加一些控制线路，以达到运算速度快、精度高、功能强的目的。

如：运算器中的乘除运算和某些逻辑运算是通过移位操作来实现的。在 ALU 的输出端设置移位线路来实现左移、右移和直送。

移位线路是一个多路选择器。

图 3.2 实现移位功能的多路选择器（1 位）





# 逻辑门电路的分类

## □ 组合逻辑电路

- 不具备记忆功能，任意时刻的输出信号仅取决于该时刻的输入信号，而与电路过去的电平状态无关。
- 建立在简单逻辑门基础上，可以直接用真值表和逻辑表达式表示。如移位器。

## □ 时序逻辑电路

- 具有记忆功能，电路的输出不仅取决于当时的输入状况，而且取决于电路的状态。
- 建立在触发器的基础上，如寄存器、计数器

# 常用的组合逻辑电路

## 加法器

**译码器**：把某组编码翻译为唯一的输出，实际应用中要用到的有**地址译码器**和**指令译码器**。

**多路选择器**：在**地址选择信号**的控制下，从多路数据中选择一种作为输出信号。

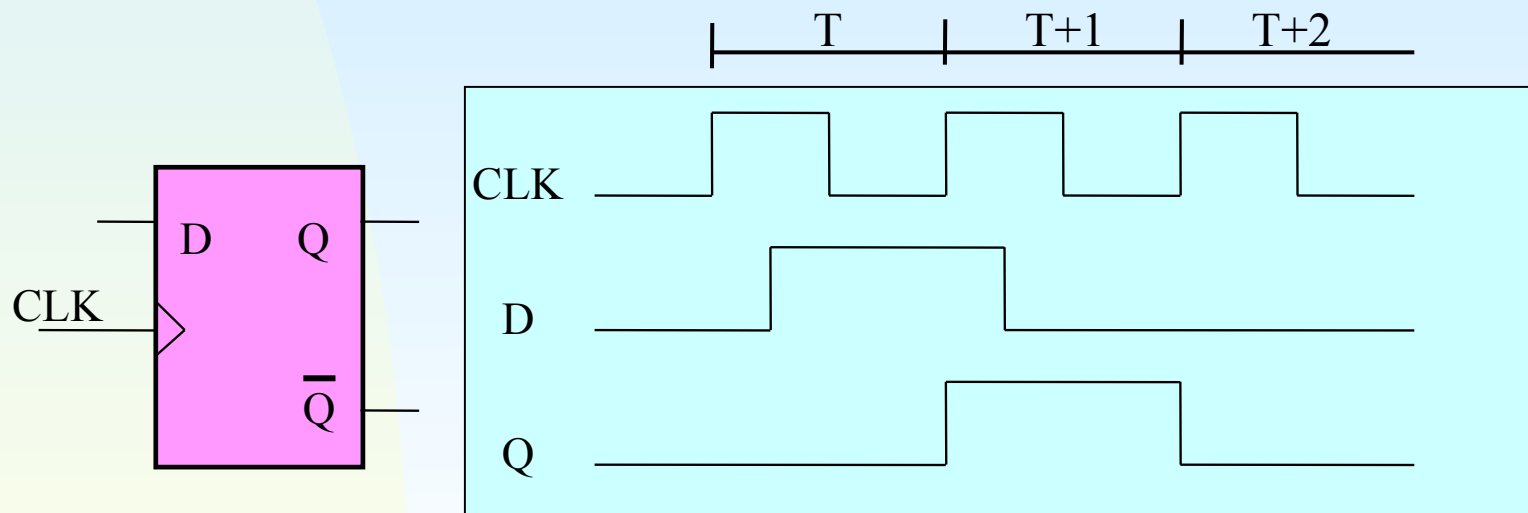
## 移位器

# 时序逻辑电路——D 触发器

一个时钟输入信号

上升沿，下降沿，周期，频率

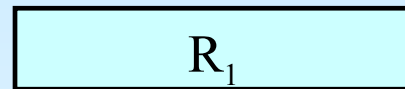
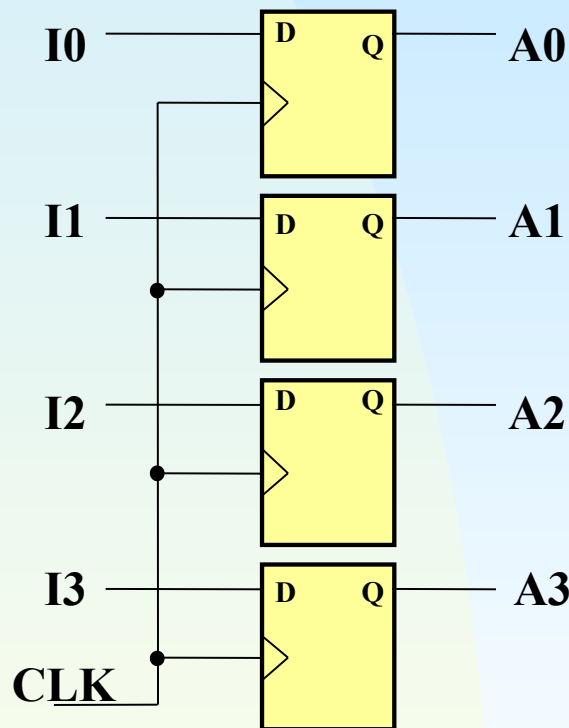
在时钟信号的上升沿，输入信号 **D** 送入内部，并改变输出 **Q**



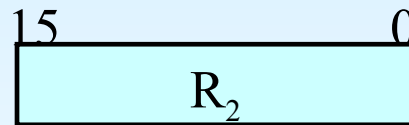
# 寄存器 ( Register )

由多个 **D 触发器** 构成，可以存放一个完整的二进制数据。

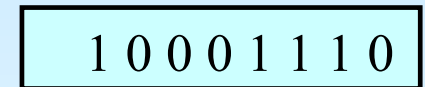
对寄存器内容的**操作**：移位、计数、清除、装入。



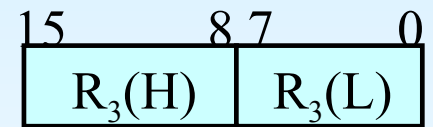
(a) 符号表示



(c) 表示位序



(b) 表示各个位



(d) 表示高低字段

或者：中、小规模集成 RAM。

# 顺序脉冲发生器

**顺序脉冲发生器**：产生一组在时间上有先后顺序的脉冲。

**用途**： 机器执行指令时，是将一条指令分成一些有先后顺序的基本动作；

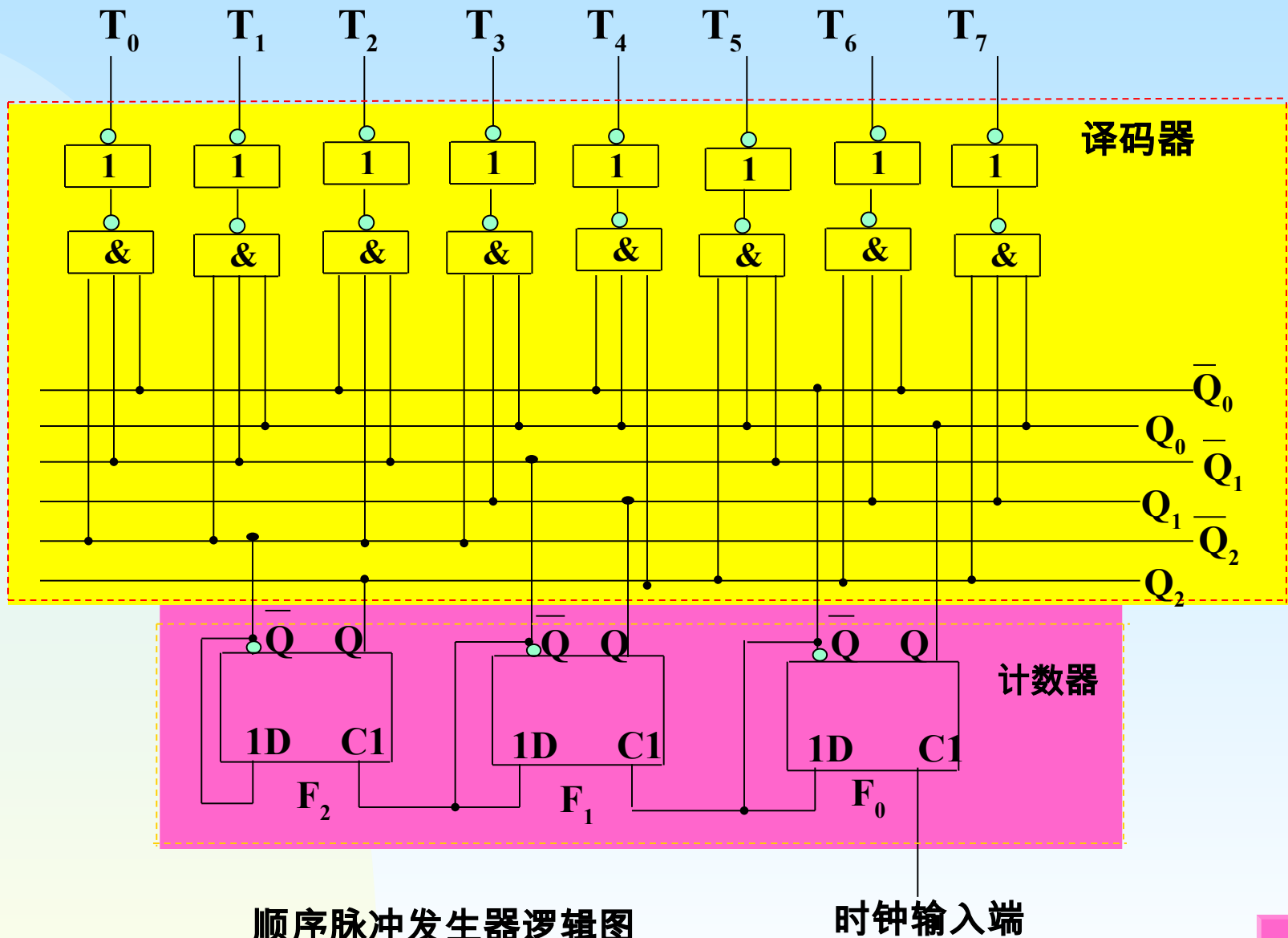
控制器发生一系列**节拍脉冲**，控制这些基本动作有顺序地的完成，实现一条指令的功能。

**电路组成**：

**计数器**：按设计要求计脉冲 CP 的个数。

**译码器**：将计数器状态翻译成对应输出端的高低电平，顺序输出脉冲信号。

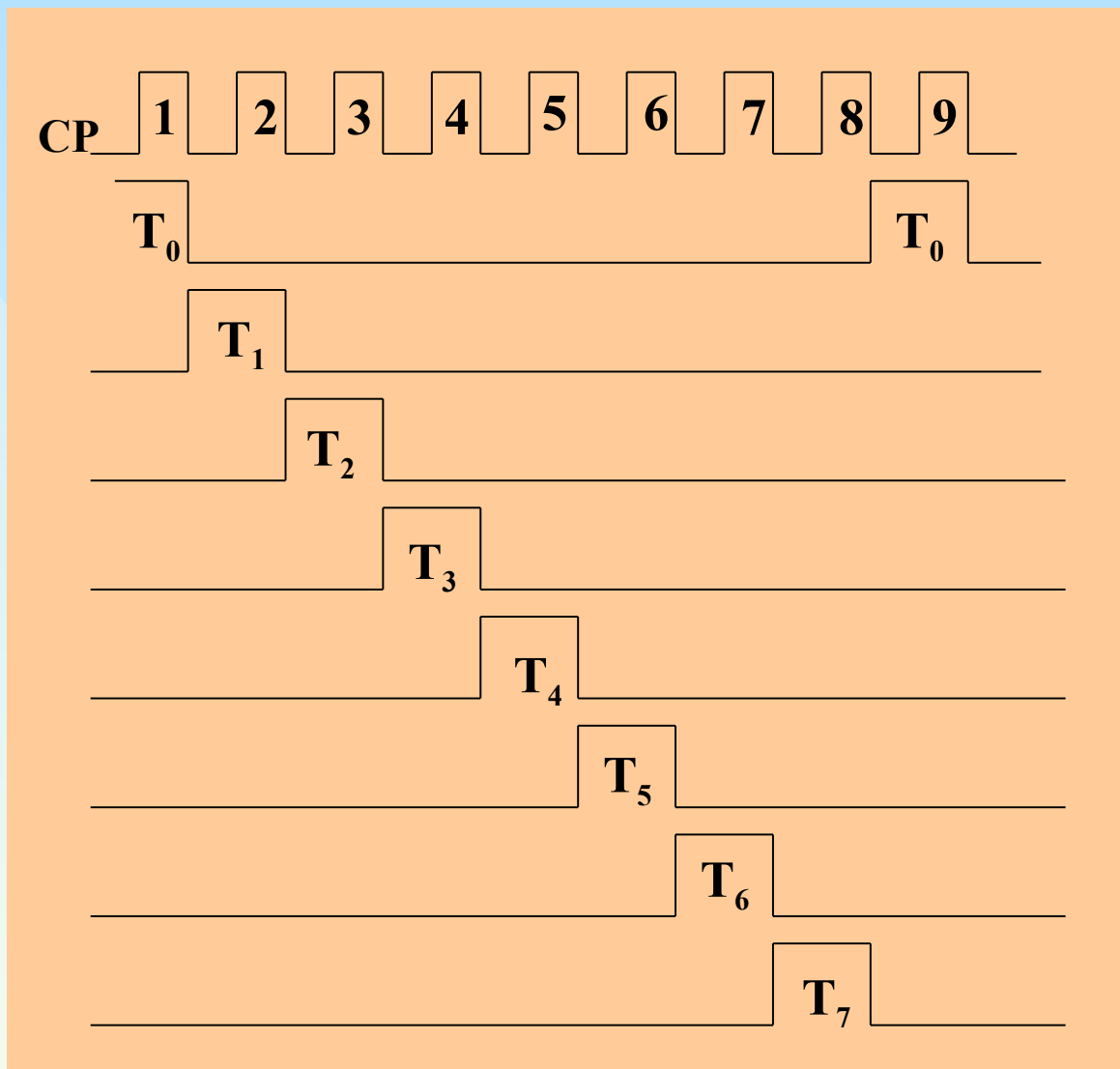




顺序脉冲发生器逻辑图

时钟输入端





顺序脉冲发生器的波形图

## 3.2 算术与逻辑单元

### 3.2.1 半加器与全加器

**加法器**：计算机中最基本的运算单元。

运算器中各种运算都是分解成加法运算进行的。

#### 1. 半加器

**半加**：两个一位二进制数相加。

**半加器**：实现半加操作的电路。

半加器不考虑低位的进位。



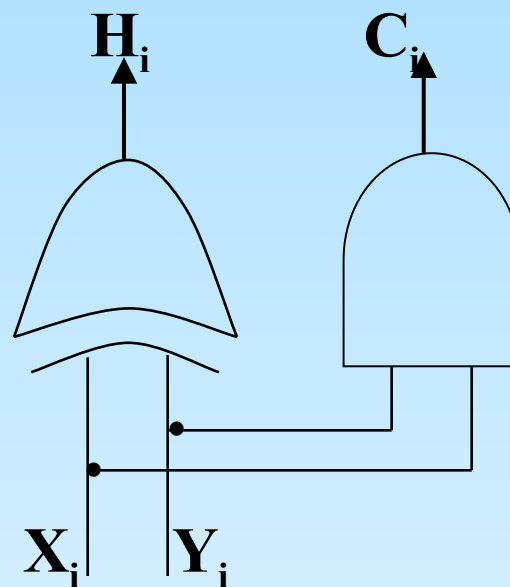
$X_i$	$Y_i$	$H_i$	$C_i$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

两个一位二进制数  $X_i$ 、 $Y_i$  相加的真值表

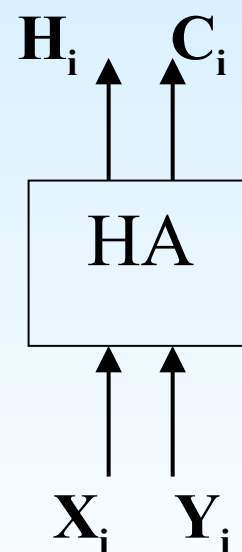
$H_i$  和  $C_i$  的逻辑表达式如下

$$H_i = \bar{X}_i Y_i + X_i \bar{Y}_i = X_i \oplus Y_i$$

$$C_i = X_i Y_i$$



(a) 逻辑图



(b) 符号表示

# 加法计算过程

计算  $0.111+0.011$

0	1	1	1	被加数
0	0	1	1	加数
<hr/>				
1	1	1	1	进位
<hr/>				
1	0	1	0	和

第 i 位

第 i 位加法过程：

一个全加器有 5 个端口

输入端：3

输出端：2



## 2. 全加器

**全加运算**：考虑低位进位的加法运算。

**全加器**：实现全加运算的电路。

全加运算的真值表：

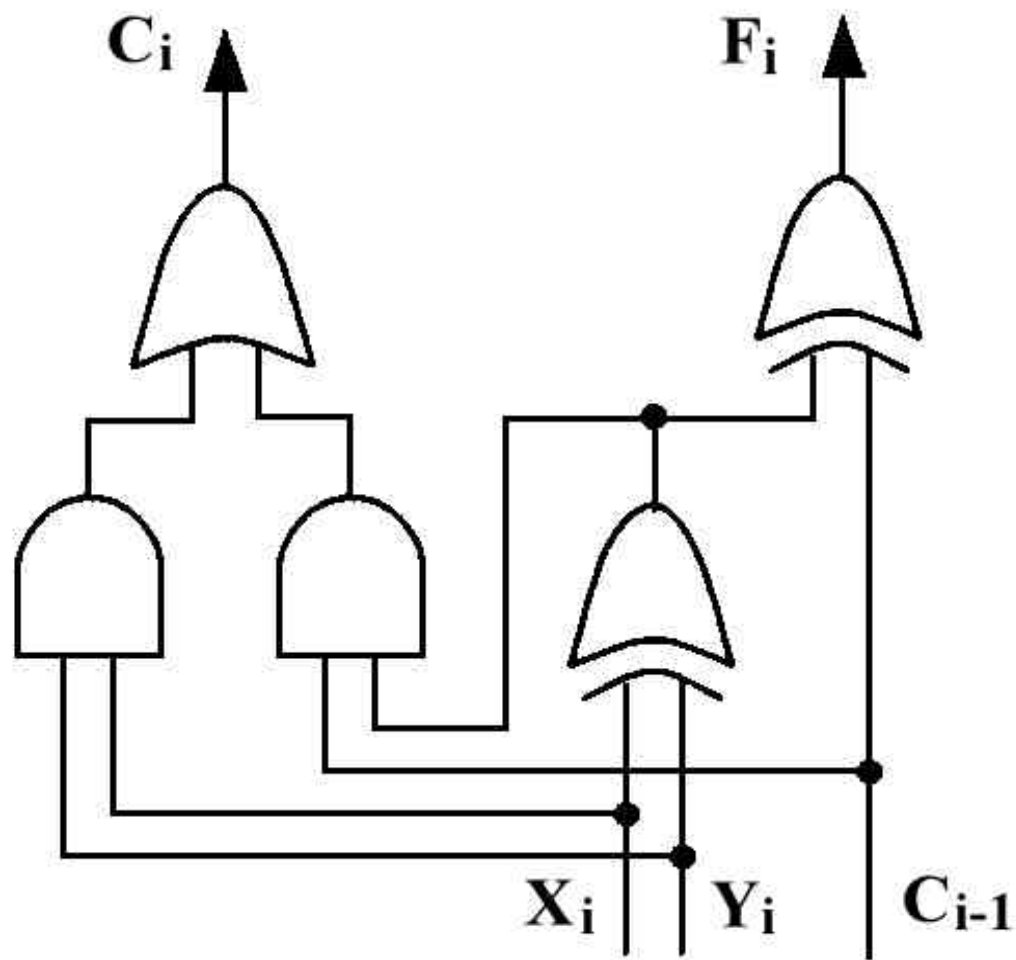
$X_i$	$Y_i$	$C_{i-1}$	$F_i$	$C_i$
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

根据真值表可写出  
 $F_i$  和  $C_i$  的逻辑  
表达式：

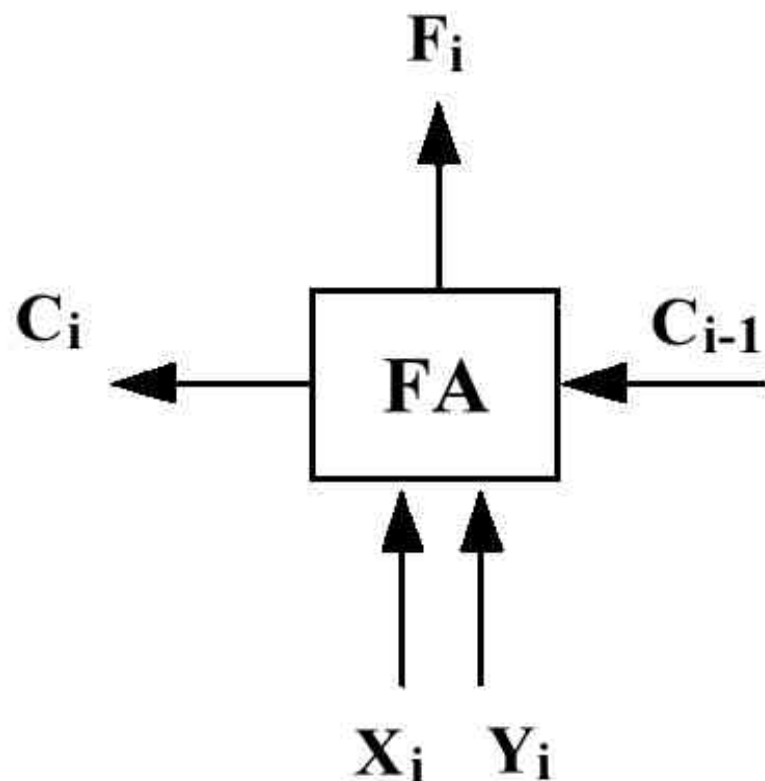
$$F_i = \bar{X}_i Y_i \bar{C}_{i-1} + X_i \bar{Y}_i \bar{C}_{i-1} + \bar{X}_i \bar{Y}_i C_{i-1} + X_i Y_i C_{i-1} = X_i \oplus Y_i \oplus C_{i-1}$$

$$C_i = X_i Y_i \bar{C}_{i-1} + \bar{X}_i Y_i C_{i-1} + X_i \bar{Y}_i C_{i-1} + X_i Y_i C_{i-1} = (X_i \oplus Y_i) C_{i-1} + X_i Y_i$$

## 实现逻辑表达式的全加器逻辑图和全加器的符号表示：



全加器的逻辑图



全加器的符号表示