数字电路分析与设计

数字电路基础

（1.1 ~ 1.4）

n 数字电路基础

ü 数字电路：用来加工和处理数字信号的电子电路。

ü 数字信号：幅度随时间断续变化。

（离散，非连续）

ü 例：雷达信号、电视中的同步信号、脉冲信号，计算机中的工作信号 ...

ü 显著特征：只有两种取值。

可以是一个脉冲的有无，电位的高低，开关的闭合与断开、事情的真和假等等描述。

ü 数字（逻辑/集成）电路：采用二进制的计数体制。

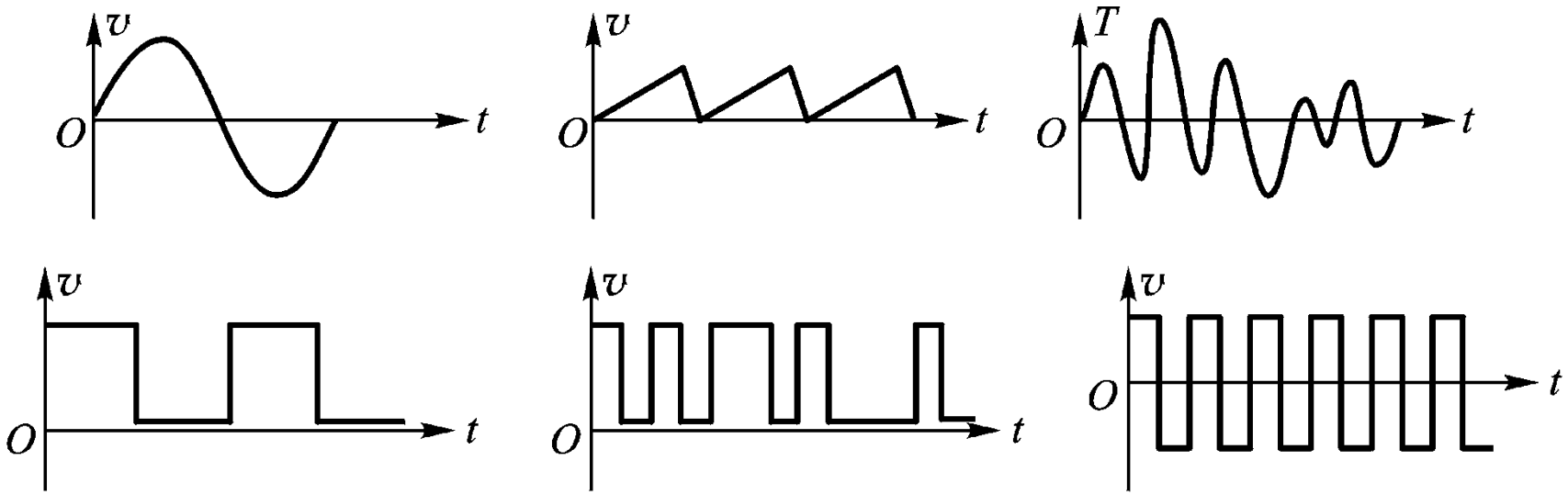
- 数字电路基础
  - 数字信号与数字电路（1.1）
  - 数字电路中的数制（1.2）
    （数制及其转换）
  - 数字电路中的代码（1.3）
    （码制、编码）
  - 数字电路中的基本逻辑函数（1.4）

v 数字信号与数字电路

- ü 模拟/数字信号
- ü 模拟/数字电路
- ü 数字信号处理
- ü 数字信号传输
- ü 数字逻辑器件（门）

Ø 模拟/数字信号

ü 模拟信号：信号幅值随时间连续变化；
数字信号：信号幅值在时间和数值上都是离散或断续的。



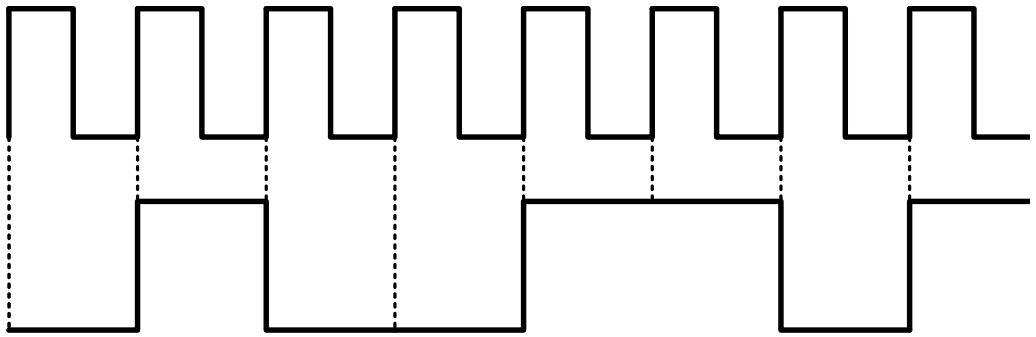
ü 模拟信号：细致，敏感（幅度）；
数字信号：抗干扰，时序性。
（逻辑信号、脉冲信号 ... 逻辑 ...）

Ø 数字信号（描述）

ü 两种描述方法：二值数字逻辑、数字信号波形。

ü 二值：数字（值）0、数字（值）1；
不代表具体的数值，只表示两种截然相反的状态；
如：高低电平、开关、通断、真假、是非等。
（也称逻辑 0、逻辑 1）

ü 波形：逻辑电平关于时间的图形表示。

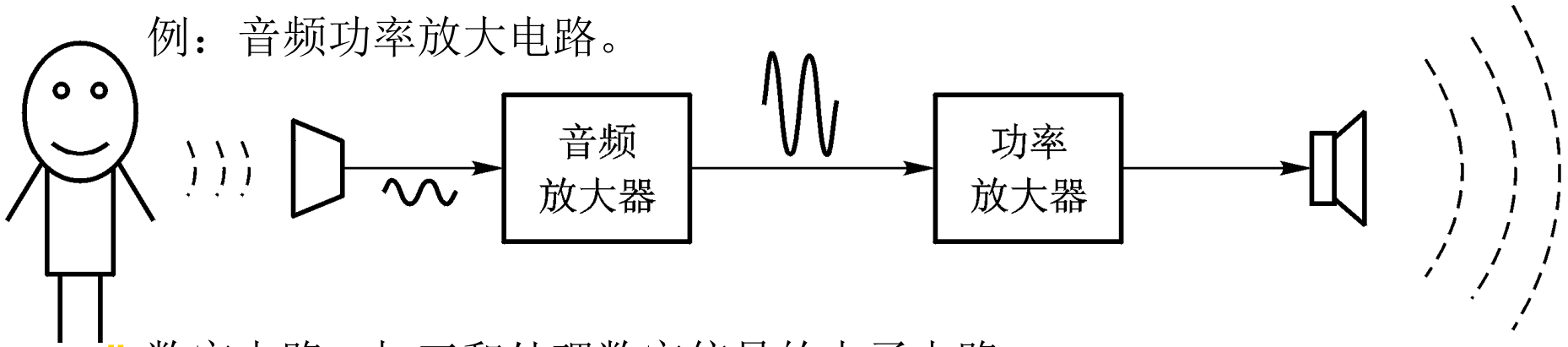


（在时钟脉冲控制下，包含信息 01001101）

Ø 模拟/数字电路

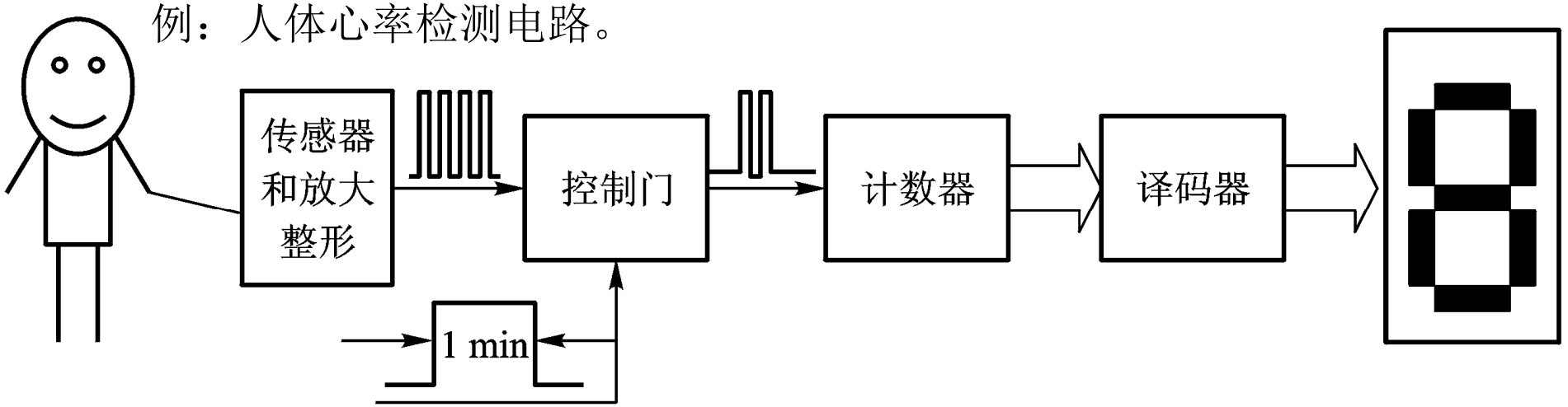
ü 模拟电路：加工和处理模拟信号的电子电路；

例：音频功率放大电路。



ü 数字电路：加工和处理数字信号的电子电路；

例：人体心率检测电路。



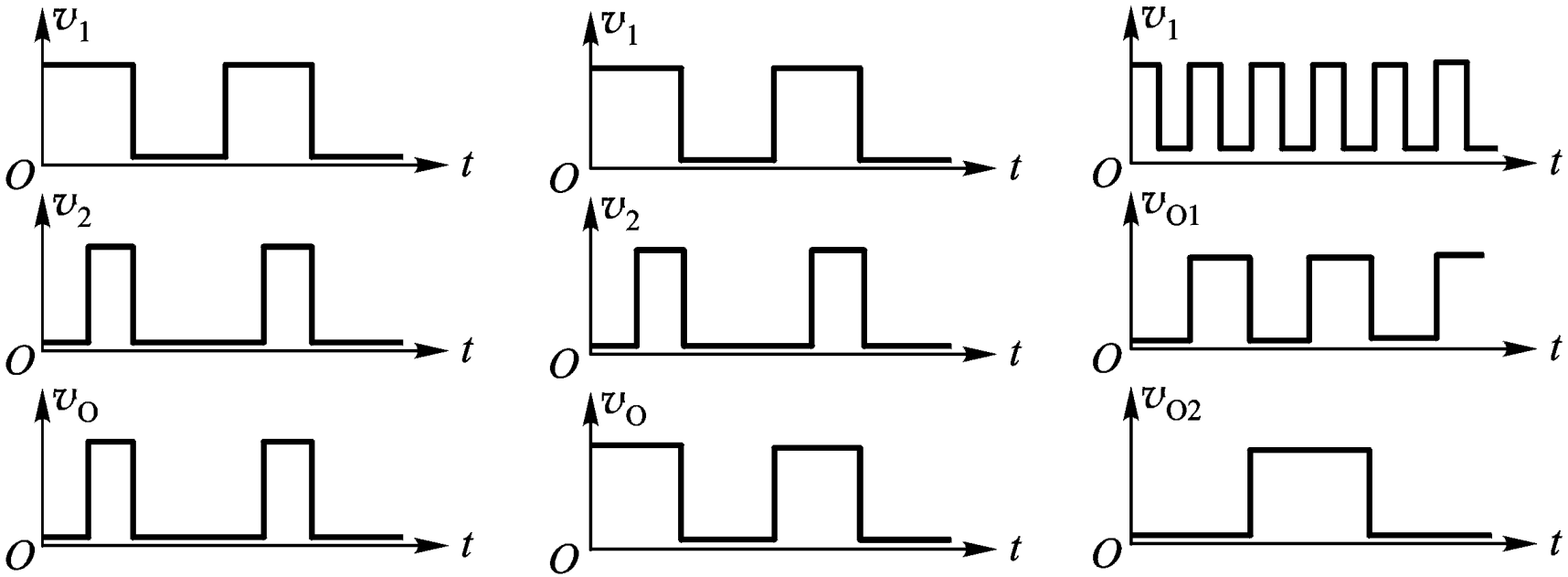
Ø 数字信号处理

ü 逻辑处理

例：“与”处理（所有输入信号同时为高时，输出才为高）

例：“或”处理（只要有一路输入信号为高，输出即为高）

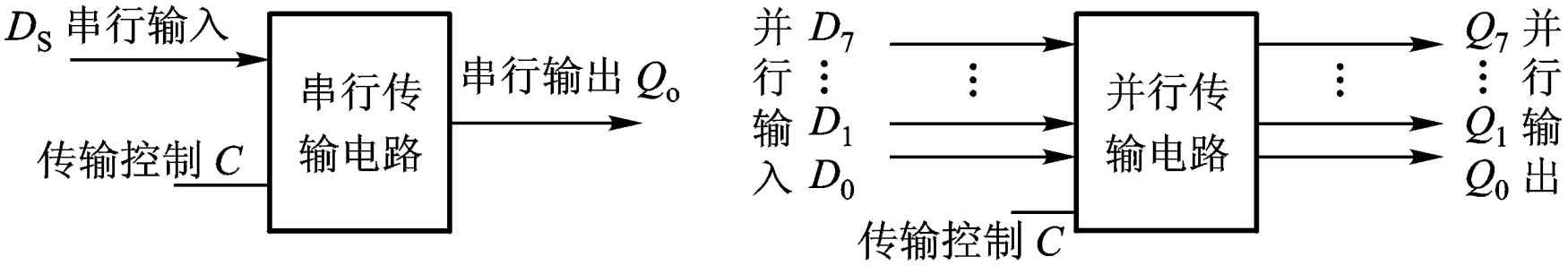
例：分频



例： ...

Ø 数字信号传输

ü 串行传输：数据信息串行排列成数据流，在一条信道上传输。
（简单，一条传输线，适合远距离传输，但速度慢）



ü 并行传输：数据信息分别在不同的并行信道上同时传输。
（速度快，但设备成本较高，且不宜远距离传输）

Ø 数字逻辑器件/门

ü 数字（集成）电路：

标准逻辑器件（Standard Logic Device）；

微控制器（Micro Control Unit）、微处理器（Micro Processor）；

专用集成电路 ASIC（Application Specific Integrated Circuit）；

可编程逻辑器件 PLD （Programmable Logic Device）等。

ü 标准逻辑器件：TTL74/54系列、CMOS4000/4500/74HC系列等。

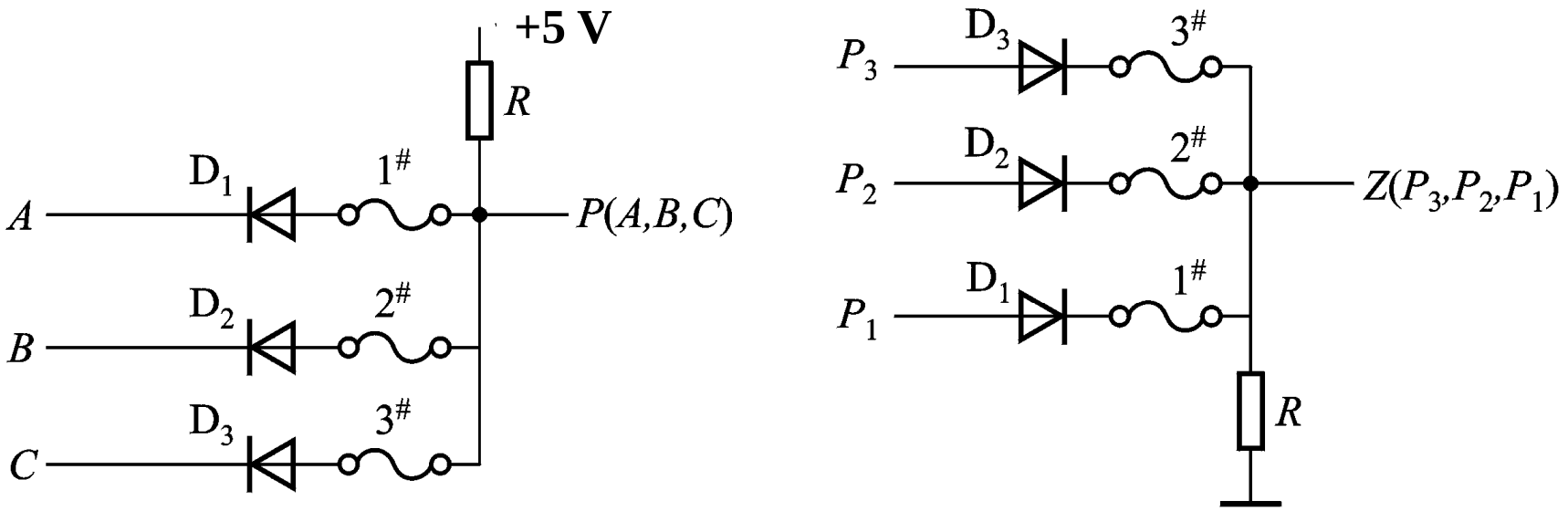
中小规模（集成数小于 1000）；型号齐全，接口规范，易于使用、匹配、互换，价廉；不适于作复杂数字系统（繁杂，可靠性低，成本高 ...）。

ü 可编程逻辑器件 PLD （Programmable Logic Device）：

逻辑关系不用固定的硬件（硬连接）来实现；利用某种电路，通过编程技术来实现各种逻辑关系，进而进行各种逻辑设计，从而达到硬件电路“软化”的目的。

Ø 数字逻辑器件/门（可编程逻辑器件）

ü 例：利用熔丝实现的二极管可编程与、或逻辑。

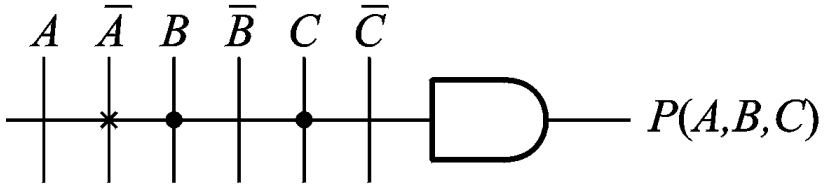


ü 熔丝的通断可以用一种外部可控（导通或截止）的器件代替，即可实现人为控制（软件编程）。

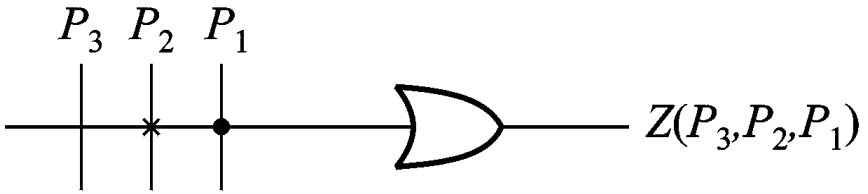
ü在 PLD 应用中，使用者在特定的开发平台上，对逻辑功能进行编程，然后编译，下载至具体芯片，则该芯片便具备了某种逻辑功能。

Ø 可编程逻辑器件（PLD 表示法）

ü PLD 与门



PLD 或门

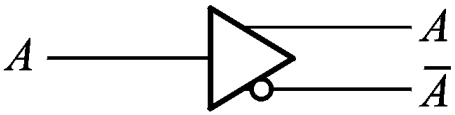


ü 硬（原始）连接：输入项、门输入线的交叉处用“•”表示；

（用户）编程连接：输入项、门输入线的交叉处用“×”表示；

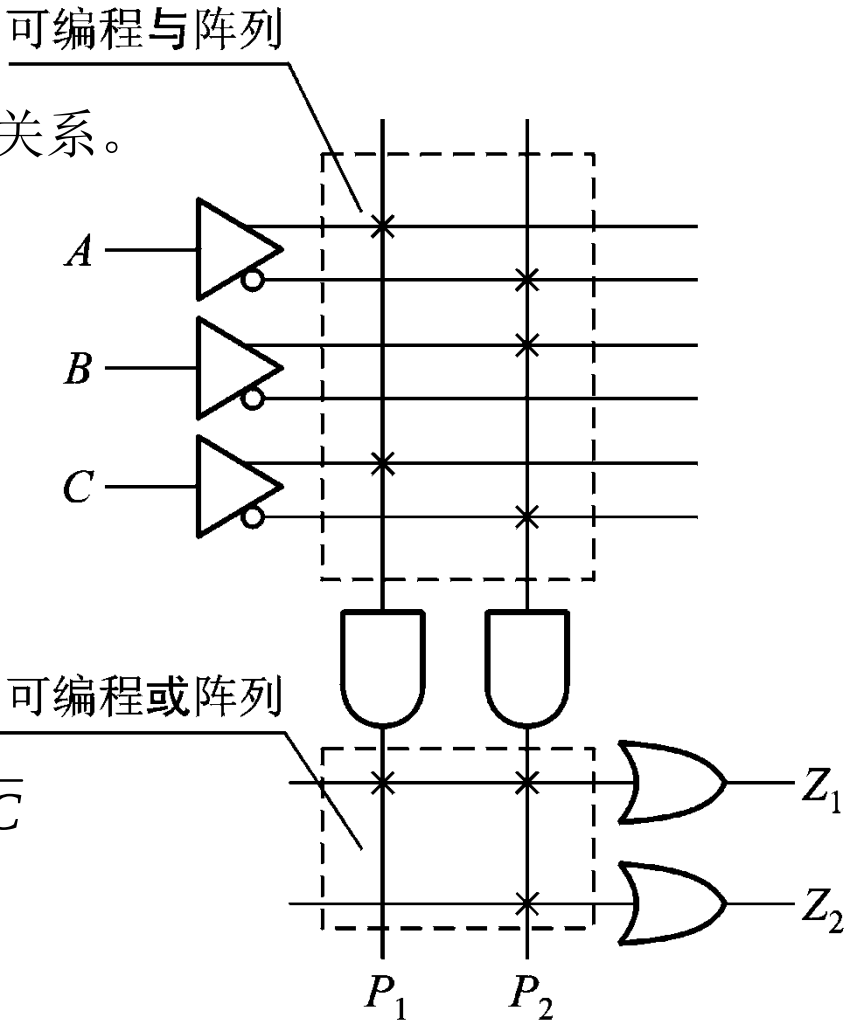
空（无）连接：输入项、门输入线之间断开。

ü PLD 缓冲器



【例1.1-1】

写出右图所示可编程门阵列电路的逻辑关系。



解：由图可得逻辑关系：

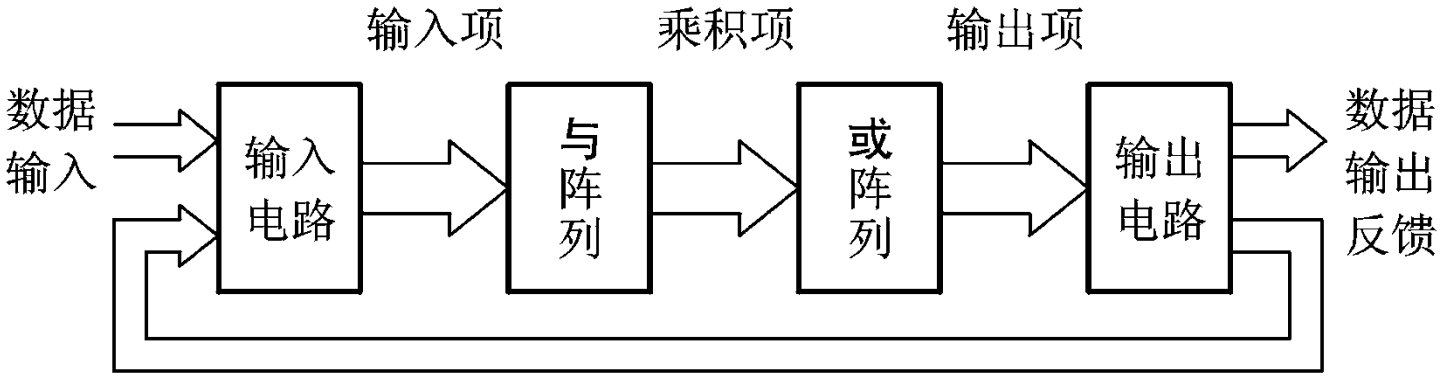
$$\begin{cases} P_1 = A\text{与}C \\ P_2 = (A\text{非})\text{与}(B)\text{与}(C\text{非}) \end{cases}$$

$$Z_1 = P_1\text{或}P_2 \quad , \quad Z_2 = P_2$$

逻辑函数式（以后会学到）：

$$Z_1 = A\cdot C + \overline{A}\cdot B\cdot\overline{C} \quad , \quad Z_2 = \overline{A}\cdot B\cdot\overline{C}$$

下图所示 PLD 的一般结构框图

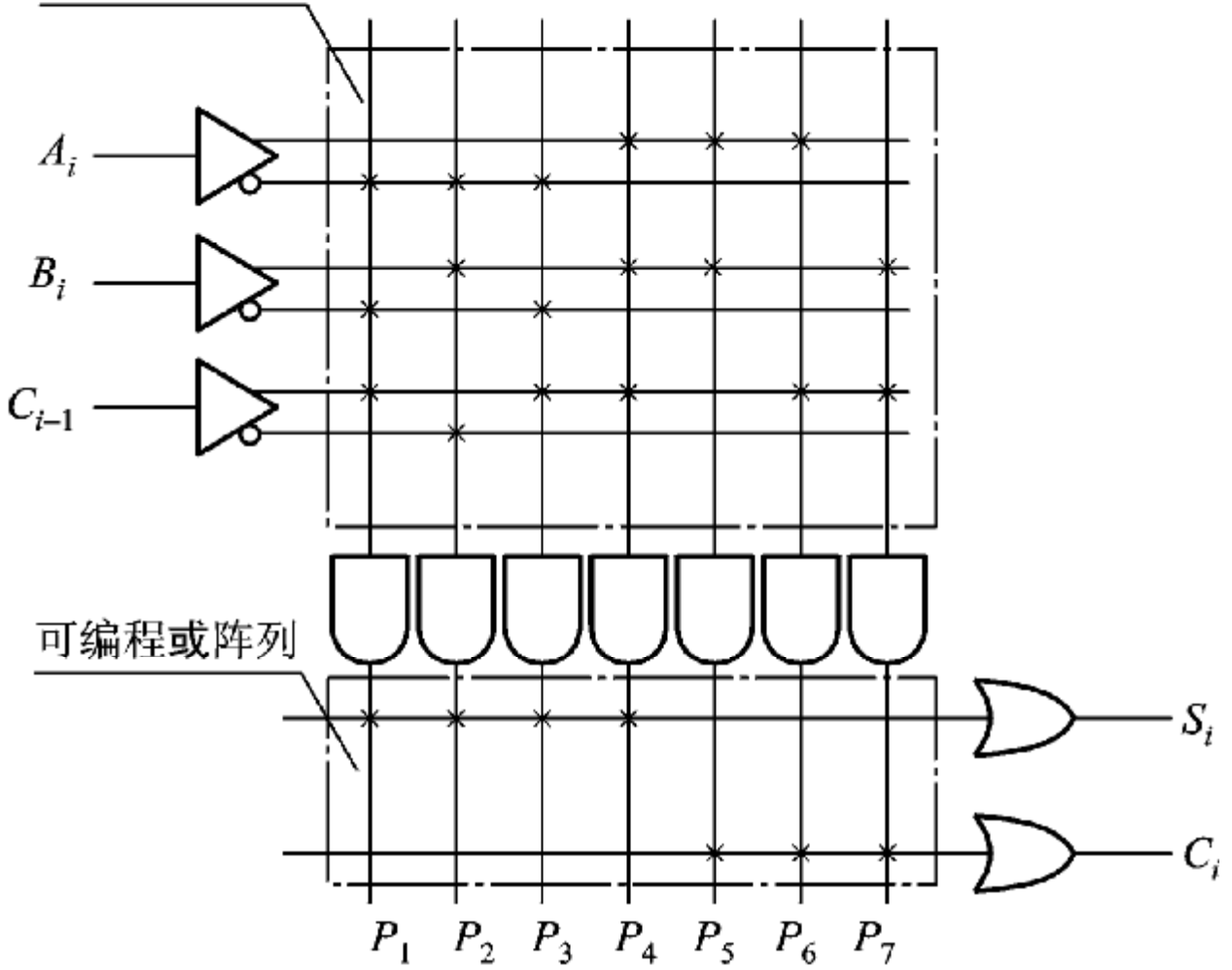


【例1.1-2】

用 PLD 实现的 1 位全加器分析。

$$S_i = \overline{A_i}\ \overline{B_i}\ C_{i-1} + \overline{A_i} B_i \overline{C_{i-1}} + A_i \overline{B_i}\ \overline{C_{i-1}} + A_i B_i C_{i-1}$$

$$C_i = A_i B_i + A_i C_{i-1} + B_i C_{i-1}$$



v 数字电路中的数制（数制及其转换）

- ü 数字信号只有两种电平或两种截然不同的状态。

- ü 日常生活中采用十进制计数体制，数字电路中采用二进制；
  数字电路中还有八进制和十六进制数。

- ü 十进制（Decimal）：（100）D、（100）$_{D}$、（100）$_{10}$
  二进制（Binary）：（01100100）B、（01100100）$_{B}$、（01100100）$_{2}$
  八进制（Octal）：（144）O、（144）$_{O}$、（144）$_{8}$
  十六进制（Hexadecimal）：（64）H、（64）$_{H}$、（64）$_{16}$

- ü 二进制是数字电路的需要，八、十六进制是为书写方便。

- ü 本节：计数体制及其相互转换。

Ø 十进制

ü 数码：0～9，共 10 个；

基数：10；

进位规则：逢 10 进 1；

权（表示某个数码的倍率）：10^i；

按权展开表达式（通式）：$(N)_{10}=\sum_{i=-m}^{n-1}K_i\times(r)^i$

（N：结果；r：基数；n：整数位数；m：小数位数；i：位）

ü 例：$(123.4)_{10}=(123.4)_{\mathrm{D}}=1\times10^2+2\times10^1+3\times10^0+4\times10^{-1}$

Ø 二进制

ü 数码：0～1，共 2 个；

基数：2；

进位规则：逢 2 进 1；

权（表示某个数码的倍率）：2^i；

按权展开表达式（通式）：$(N)_{10}=\sum_{i=-m}^{n-1}K_i\times(r)^i$

（N：结果；r：基数；n：整数位数；m：小数位数；i：位）

ü 例：$(101.1)_2=(101.1)_B=1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}$

Ø 八进制

ü 数码：0～7，共 8 个；

基数：8；

进位规则：逢 8 进 1；

权（表示某个数码的倍率）：8^i；

按权展开表达式（通式）：$(N)_{10}=\sum_{i=-m}^{n-1}K_i\times(r)^i$

（N：结果；r：基数；n：整数位数；m：小数位数；i：位）

ü 例：$(123.4)_8=(123.4)_O=1\times8^2+2\times8^1+3\times8^0+4\times8^{-1}$

Ø 十六进制

ü 数码：0～9、A、B、C、D、E、F，共 16 个；

基数：16；

进位规则：逢 16 进 1；

权（表示某个数码的倍率）：16^i；

按权展开表达式（通式）：$(N)_{10}=\sum_{i=-m}^{n-1}K_i\times(r)^i$

（N：结果；r：基数；n：整数位数；m：小数位数；i：位）

ü 例：$(1A3.F)_{16}=(1A.F)_H=1\times16^2+10\times16^1+3\times16^0+15\times16^{-1}$

要求熟记 0～15 各类进制的书写

Ø 数制转换（二、八、十六进制～十进制）

ü 转换依据：展开前后之和相等。

ü 二、八、十六进制数转换成十进制数：按权展开并求和。

$$(N)_{10}=\sum_{i=-m}^{n-1}K_i\times(r)^i$$

ü 例：$(101.1)_2=1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}=5.5=(5.5)_{10}$

$(123.4)_8=1\times8^2+2\times8^1+3\times8^0+4\times8^{-1}=83.5=(83.5)_{10}$

$(AC.5)_{16}=10\times16^1+12\times16^0+5\times16^{-1}=172.3125=(172.3125)_{10}$

LSB：Least Significant Bit

MSB：Most Significant Bit

Ø 数制转换（十进制 ~ 二进制）

ü 转换依据：展开前后之和相等。

ü 十进制整数部分转换成二进制数：除 2 取余。

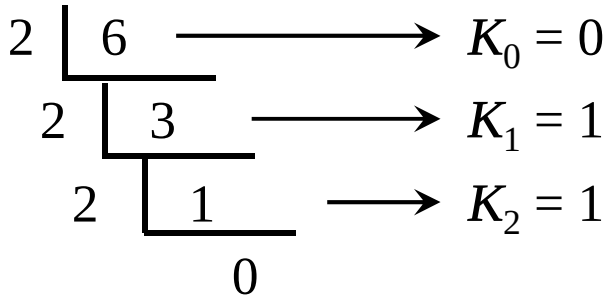
$$
\begin{aligned}
(N)_{10} &= K_{n-1}\times 2^{n-1} + K_{n-2}\times 2^{n-2} + \cdots + K_2\times 2^2 + K_1\times 2^1 + K_0\times 2^0 \\
&= 2\{K_{n-1}\times 2^{n-2} + K_{n-2}\times 2^{n-3} + \cdots + K_2\times 2^1 + K_1\times 2^0\} + K_0 \\
&= 2\{2[K_{n-1}\times 2^{n-3} + \cdots + K_2\times 2^0] + K_1\} + K_0 \\
&\cdots \\
&= 2\{2[\cdots 2(0) + K_{n-1} + \cdots + K_2] + K_1\} + K_0
\end{aligned}
$$

ü 最后结果：$(K_{n-1}K_{n-2}\cdots K_2K_1K_0)_2$

第一次除 2 后获得的余数为最低位（LSB）；

最后一次获得的余数为最高位（MSB）。

ü 例：



LSB：Least Significant Bit

MSB：Most Significant Bit

Ø 数制转换（十进制～二进制）

ü 转换依据：展开前后之和相等。

ü 十进制小数部分转换成二进制数：

乘 2 取整，直至积的小数部分等于零；

若积的小数达不到零时，根据转换的精度来取位数。

ü 最后结果：$(0.K_{-1}K_{-2}K_{-3}\cdots K_{-m})_2$

第一次乘 2 后获得的积为最高位 MSB ；

最后一次获得的积为最低位 LSB 。

ü 例：$(0.125)_{10} \times 2 \rightarrow \underset{0}{0.25} \times 2 \rightarrow \underset{0}{0.5} \times 2 \rightarrow \underset{1}{1.0} = (0.001)_2$

Ø 数制转换（十进制～八、十六进制）

ü 与十进制数转换成二进制数类似。

ü 整数部分除 8/16 取余（注意顺序）；
小数部分乘 8/16 取整（注意顺序）。

Ø 数制转换（二 ~ 八 ~ 十六进制）

- ü 八进制数的 8 个码元素和三位二进制数一一对应。
  （八进制和二进制之间转换时只需互相替换）

- ü 十六进制数的 16 个码元素和四位二进制数一一对应。
  （十六进制和二进制之间转换时只需互相替换）

- ü 八进制和十六进制之间转换可以二进制为桥梁。

- ü 例：$(2A.C)_{16} = (0010\ 1010.1100)_2$

$= (00\ 101\ 010.110\ 0)_2 = (52.6)_8$

Ø 正负数

ü 在数据最高位前设置一位符号位；
一般：符号位为 “0” 表示正数，符号位为 “1” 表示负数。

ü 例：–1.101⇒11.101

ü 符号位和数值位之间有三种编码形式：原码，反码和补码。

Ø 原码

ü 在数据最高位前设置一位符号位；
一般：符号位为“0”表示正数，符号位为“1”表示负数。

ü $[X]_{原}$ = 符号位 + 原数值
X1 = + 1001010，$[X1]_{原}$ = 01001010
X2 =－1001010，$[X2]_{原}$ = 11001010

ü 原码表示简单，直观，比较适合乘法和除法运算；
二进制乘法运算中用于决定积的符号也较容易；
减法运算时的符号位较难求出。

Ø 反码

ü 在数据最高位前设置一位符号位；
一般：符号位为 “0” 表示正数，符号位为 “1” 表示负数。

ü 若原数为正数：反码 = 符号位＋原数值
X1 = + 1001010，$[X1]_{反}$ = 01001010

ü 若原数为负数：反码 = 符号位 + 原数值的反码
（反码：原数值按位求反）
X2 =－1001010，$[X2]_{反}$ = 10110101

Ø 补码

ü 在数据最高位前设置一位符号位；
一般：符号位为 “0” 表示正数，符号位为 “1” 表示负数。

ü 若原数为正数：补码 = 符号位＋原数值
X1 = + 1001010，$[X1]_{补}$ = 01001010

ü 若原数为负数：补码 = 符号位 + 原数值的补码
（补码：原数值的反码 + 1）
X2 =－1001010，$[X2]_{补}$ = 10110110

ü 特性：[补码] + [补码] = [补码]，$[补码]_{补}$ = 原码

ü 例：1001 －1100
$[1001]_{补}$ + [－1100]$_{补}$ = 01001 + 10100 = $11101_{补}$ → $10011_{原}$ = -3

v 数字电路中的代码（码制、编码）

ü 生活中用一组十进制数来代表一个特定对象的情况很多：如电话号码、邮政编码等等。

ü 数字电路中，用一组二进制数来代替某一特定的对象，这组二进制数就是代表该对象的代码。

ü 代替的方法/种类很多。

Ø 二 — 十进制代码（BCD 码）

ü 0 ~ 9 十个数码，4 位二进制数：C_{16}^{10}

ü 常见二 — 十进制代码

十进制数	有 权 码					无 权 码	
	8421	5421	2421	2421＊	5211	余3码	余3循环码
0	0000	0000	0000	0000	0000	0011	0010
1	0001	0001	0001	0001	0001	0100	0110
2	0010	0010	0010	0010	0100	0101	0111
3	0011	0011	0011	0011	0101	0110	0101
4	0100	0100	0100	0100	0111	0111	0100
5	0101	1000	0101	1011	1000	1000	1100
6	0110	1001	0110	1100	1001	1001	1101
7	0111	1010	0111	1101	1100	1010	1111
8	1000	1011	1110	1110	1101	1011	1110
9	1001	1100	1111	1111	1111	1100	1010

Ø 二 — 十进制代码（BCD 码）

ü 有了二 — 十进制代码后，任何一个十进制数都可以用它们来代替。

ü 例：$(953)_{10} = (1001\ 0101\ 0011)_{8421码}$

$= (1100\ 1000\ 0011)_{5421码}$

$= (1111\ 0101\ 0011)_{2421码}$

$= (1100\ 1000\ 0110)_{余3码}$

$= (0111\ 0111\ 1001)_{2}$

Ø 循环码（格雷码）

ü 任意两组相邻代码之间只相差一位码不同（其余均相同）；具有反射性。

		000
		001
	00	011
0	01	010
1	11	110
	10	111
		101
		100

ü 用途：信息的交换和传递过程中可以减少差错。

Ø ASCII 码
（American Standard Code for Information Interchange）

ü 国际标准组织制定的代码（选用美国国家信息交换标准代码）。

ü 7 位二进制代码，包括十进制数的 10 个数码，52 个英文字母（大小写），以及+、-、×、÷、……等特定对象；
（有时增加一位作奇偶校验）

Ø 奇偶校验码

ü 校验数据传输正确性的最简单方法。

ü 发送端：传输数据的后面加一位奇偶校验位。

ü 奇校验：利用这一位将该组传输数据中 1 的个数补成奇数；
偶校验：利用这一位将该组传输数据中 1 的个数补成偶数。

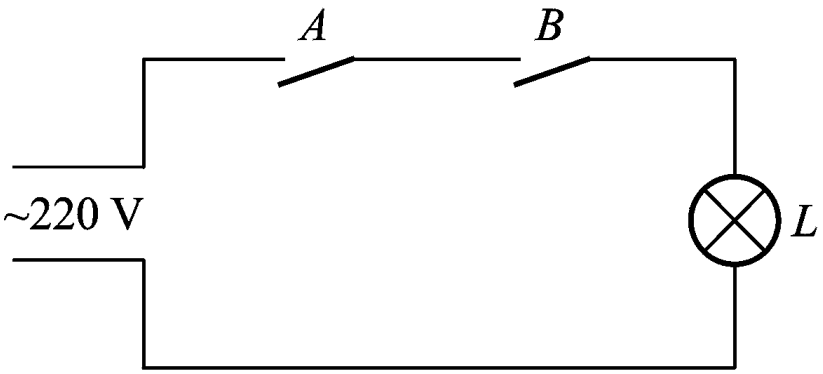
ü 例：10101011 — 101010110（使 1 的位数为奇数，奇校验）
10101011 — 101010111（使 1 的位数为偶数，偶校验）

ü 接收端：通过检查接收到 1 的个数的奇偶性，来判断数据传送过程中是否有错误。

v 数字电路中的基本逻辑函数

- ü 数字逻辑电路中，基本的逻辑关系有：与、或、非。
- ü 由此三种基本的逻辑关系可以派生出多种复杂的逻辑关系。

Ø 与 逻辑



ü 右图所示开关串联型控制电路。

ü 逻辑定义：

开关闭合为逻辑 1、开关断开为逻辑 0；

灯亮为逻辑 1、灯暗为逻辑 0 。

ü 真值表

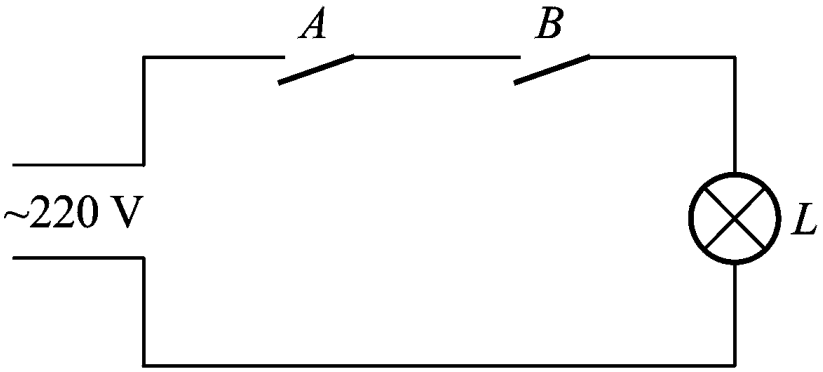
（反映输入输出逻辑关系的数学列表）

A	*B*	*L*
0	0	0
0	1	0
1	0	0
1	1	1

与逻辑：所有输入条件都具备时，输出结果才成立。

Ø 与 逻辑

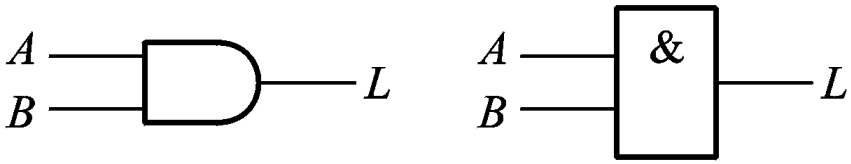
ü 与运算的逻辑函数式：$L = A \cdot B = AB$（逻辑乘）



ü 与运算规则：$0 \cdot 0 = 0$，$0 \cdot 1 = 0$，$1 \cdot 0 = 0$，$1 \cdot 1 = 1$

ü 与运算电路：与门；

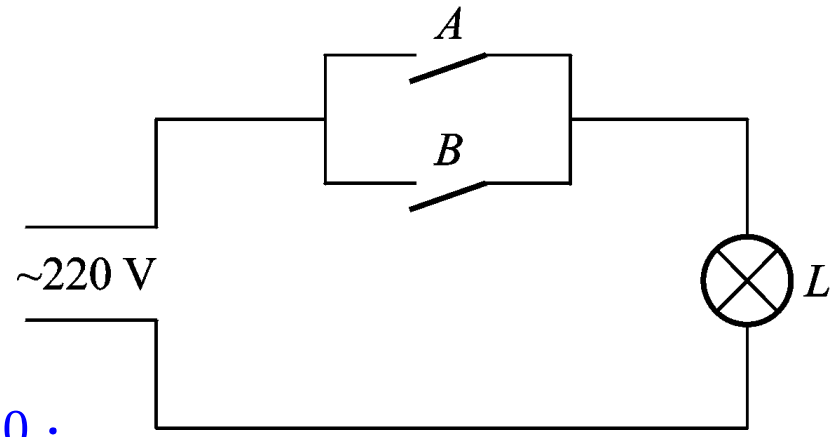
与门符号：



与逻辑：所有输入条件都具备时，输出结果才成立。

A	B	L
0	0	0
0	1	0
1	0	0
1	1	1

Ø 或 逻辑



ü 右图所示开关并联型控制电路。

ü 逻辑定义：

开关闭合为逻辑 1、开关断开为逻辑 0 ；

灯亮为逻辑 1、灯暗为逻辑 0 。

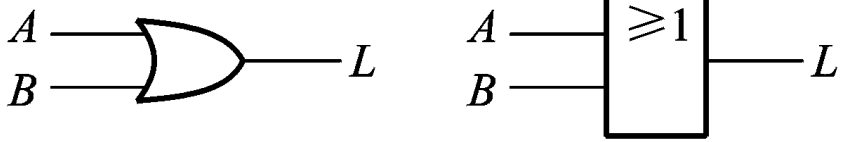
ü 真值表

ü 或运算的逻辑函数式：$L = A+B$（逻辑加）

或运算规则：0+0 = 0，0+1 = 1，1+0 = 1，1+1 = 1

或运算电路：或门；

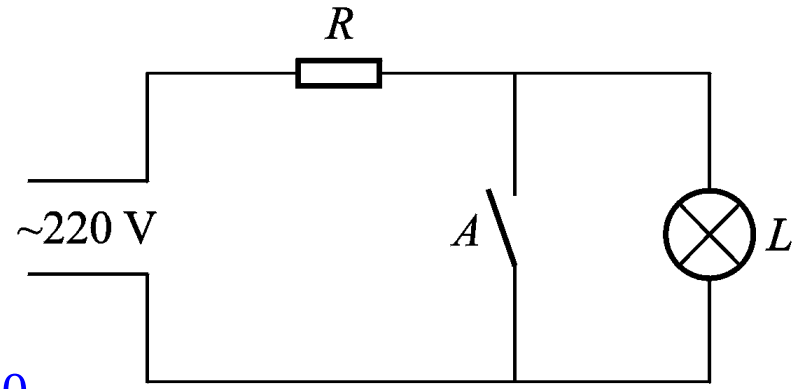
或门符号：



A	B	L
0	0	0
0	1	1
1	0	1
1	1	1

或逻辑：只要有一个（或以上）输入条件具备时，输出结果就能成立。

Ø 非 逻辑



ü 右图所示开关控制电路。

ü 逻辑定义：

开关闭合为逻辑 1、开关断开为逻辑 0；

灯亮为逻辑 1、灯暗为逻辑 0 。

A	L
0	1
1	0

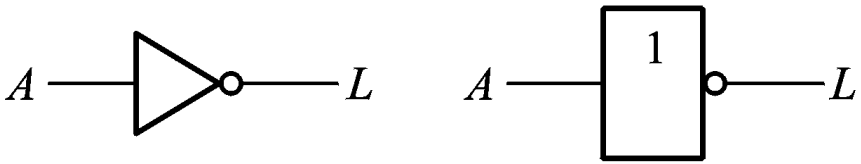
ü 真值表

ü 非运算的逻辑函数式：$L=\overline{A}$

非运算规则：$\overline{0}=1$ ， $\overline{1}=0$

非运算电路：非门；

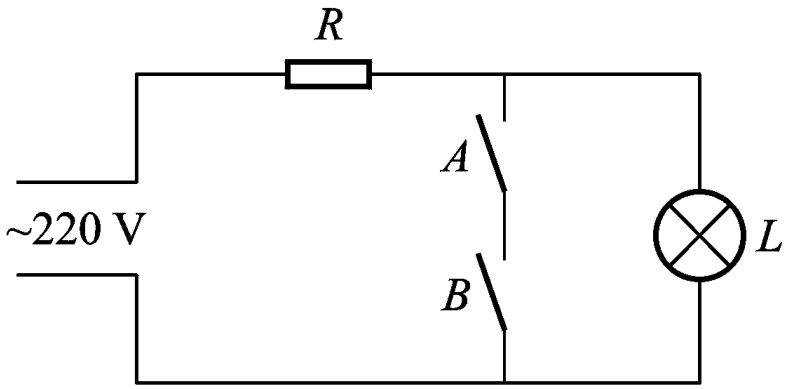
非门符号：



非逻辑：

输入条件具备时，输出结果不成立；输入条件不具备时，输出结果成立。

Ø 复杂逻辑关系（与非）

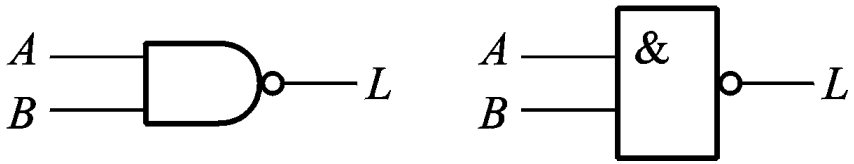


ü 右图所示开关控制电路。
任一开关打开时，灯就会亮。

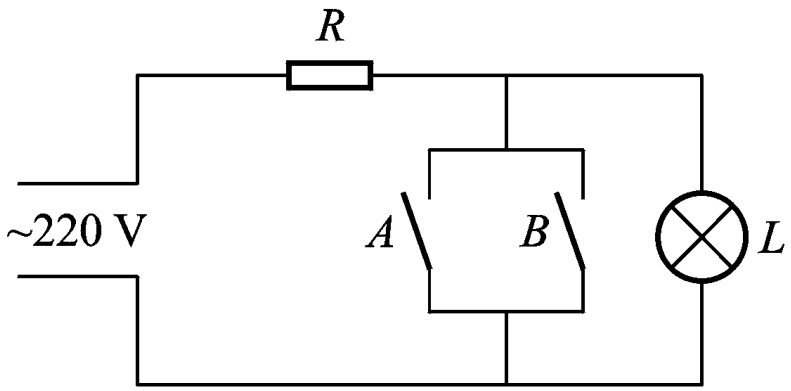
ü 真值表（按之前的逻辑定义）

A	B	L
0	0	1
0	1	1
1	0	1
1	1	0

ü 逻辑函数式：$L = \overline{AB}$
逻辑符号：



Ø 复杂逻辑关系（或非）

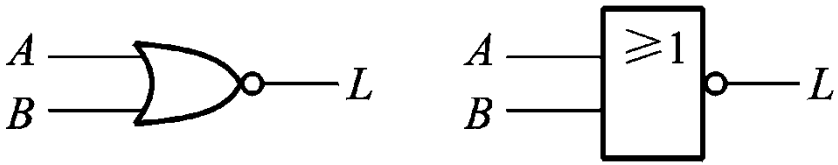


ü 右图所示开关控制电路。
仅当开关同时无效时，灯才会亮。

ü 真值表（按之前的逻辑定义）

A	B	L
0	0	1
0	1	0
1	0	0
1	1	0

ü 逻辑函数式:$L=\overline{A+B}$
逻辑符号:



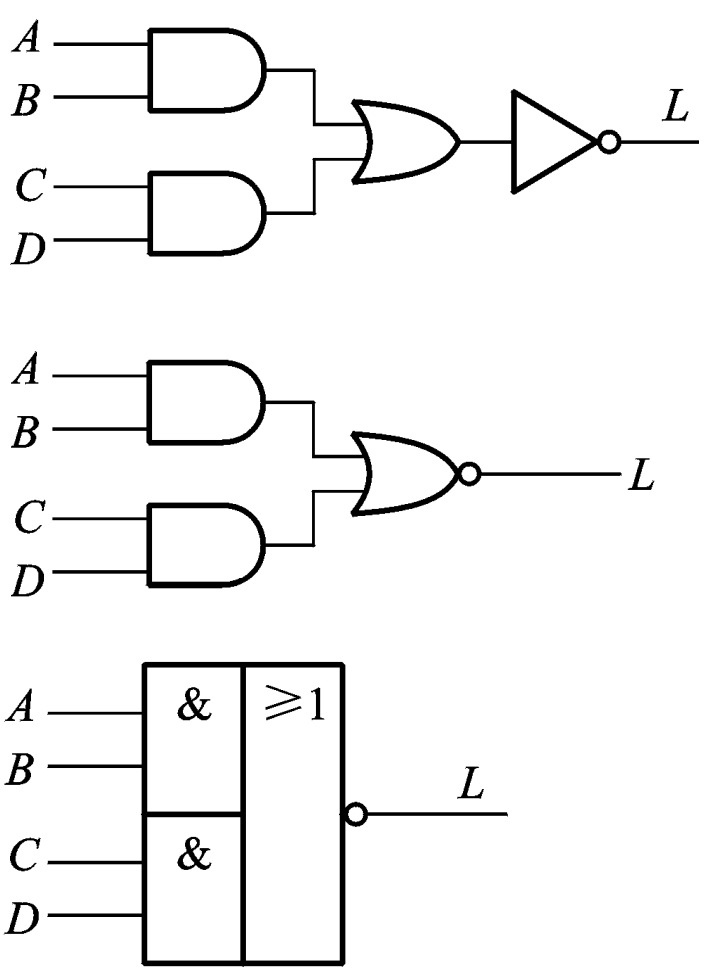
Ø 复杂逻辑关系（与或非）

ü 从名称上看，与、或、非的组合。

ü 逻辑函数式：$L = \overline{AB + CD}$

基本逻辑符号：

ü 真值表 ...



Ø 异或 逻辑

ü 异或：

决定结果的二个条件相异时，结果成立；

决定结果的二个条件相同时，结果不成立。

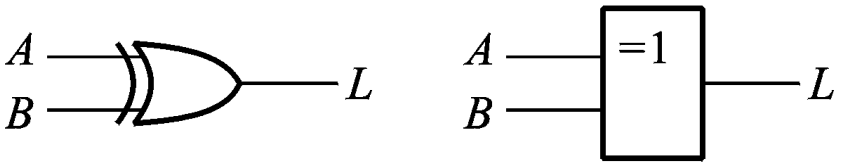
ü 异或运算的逻辑函数式：$L = \overline{A}B + A\overline{B} = A \oplus B$

ü 真值表

A	B	L
0	0	0
0	1	1
1	0	1
1	1	0

ü 异或运算电路：异或门；

异或门符号：



Ø 同或 逻辑

ü 同或：

决定结果的二个条件相异时，结果不成立；

决定结果的二个条件相同时，结果成立。

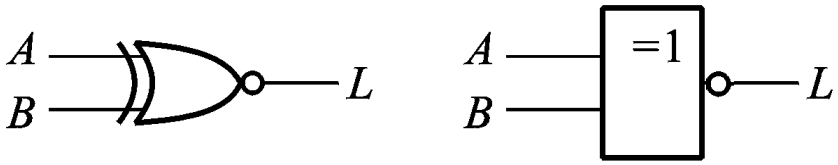
ü 同或运算的逻辑函数式：$L = \overline{A}\,\overline{B} + AB = A \odot B$

ü 真值表

A	B	L
0	0	1
0	1	0
1	0	0
1	1	1

ü 同或运算电路：同或门；

同或门符号：



v 本节作业

- ü 习题 1（P51）

  3.2/3、4.2/3、5.3、6.1、8、10、14