数字电子技术基础

第一章　　数字逻辑基础

1.1脉冲信号及其参数

模拟信号和数字信号

- 模拟量——自然界存在的随时间连续变化的物理量，
- 1、模拟信号——与自然物理量成线性关系的电信号，幅度随时间连续变化。
- 例：
- 非周期性模拟信号（温度、压力等）
- ——主要参数：幅度的大小
- 周期性模拟信号（正弦信号、锯齿波信号）
- ——主要参数：幅度、频率和周期

2．数字信号------幅度大小在时间上离散变化

脉冲信号

——周期性的、具有高、低两种幅值的离散电信。

参数：

1、周期T——信号变化一个循环的时间。

频率f——（脉冲重复率PRR），每秒时间中的脉冲周期数。

2、脉冲幅度Vm——信号的最大变化值。

低电平V_L——信号的低幅值

高电平V_H——信号的高幅值

$Vm=V_H-V_L$

3、脉冲宽度Tw——信号从上升到50%Vm至下降到50%Vm所需的时间（或高电平时间）

4、上升时间tr、-----信号从10%Vm起上升到90%Vm所需的时间

5、下降时间tf-----信号从90%Vm起下降到10%Vm所需的时间

6、占空比q------脉宽与周期之百分比：$q = (Tw / T)\ \%$

1.2数字系统中数的表示方法

1.2.1 数制

一、进位计数制基本表示法

基本要素——基数和位权

1、位置记数法：

每个数码Ki所代表的数值与其所在位有关，括号外的下标表示其计数制（基数）值。

$$(N)_R=(K_{n-1}K_{n-2}\cdots\cdots K_1K_0 . K_{-1}K_{-2}\cdots\cdots)_R$$

整数部分　　小数部分

基数 = 数码的个数

位权= 数码所在位的数值大小，第i位的位权为基数的i 次幂。

整数部分为正幂、小数部分为负幂。

2、多项式展开表示法

各位数码乘以其所在位的位权相加后得其数值（用十进制表示）。

$$(N)_R = K_{n-1}R^{n-1} + K_{n-2}R^{n-2} + \cdots + K_1R^1 + K_0R^0 + K_{-1}R^{-1} + K_{-2}R^{-2} + \cdots$$

整数部分　　小数部分

二、常用计数体制

1、十进制（Decimal）

$$(N)_{10} = (D_{n-1}D_{n-2}\ldots D_0 \bullet D_{-1}D_{-2}\ldots)_{10}$$

$$(271.59)_{10} =$$

$$2\times10^2 + 7\times10^1 + 1\times10^0 + 5\times10^{-1} + 9\times10^{-2}$$

2、二进制（Binary）

基数： 2　　位权：2^i

数符B_i： 0、1 (可以用低、高电平表示)

位置表示法：$(N)_2=(B_{n-1}B_{n-2}...B_0 \bullet B_{-1}B_{-2}..)_2$

按权展开式：

$$(N)_2=B_{n-1}2^{n-1}+B_{n-2}2^{n-2}+...+B_02^0+B_{-1}2^{-1}+B_{-2}2^{-2}+...$$

例: $(1101.101)_2=$

$$1\times2^3+1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}$$

$$=8+4+0+1+0.5+0+0.125=(13.625)_{10}$$

二进制数各位的位权

i	2^i	i	2^i	i	2^i
-4	0.0625	4	16	12	4096
-3	0.125	5	32	13	8192
-2	0.25	6	64	14	16384
-1	0.5	7	128	15	32768
0	1	8	256	16	65536
1	2	9	512		
2	4	10	1024	题1.6	
3	8	11	2048		

二进制数的运算:

加法:逢二本位归零，高位加一。

$(10110)_2+(1101)_2=(100011)_2$

减法:不够减本位借二，高位退一。

$(10110)_2-(1101)_2=(1001)_2$

乘法:被乘数根据乘数各位为1的数码的位序i移位i次并相加。

积的位数等于被乘数位数及乘数位数之和。

$(10110)_2\times(1101)_2=(100011110)_2$

3、八进制（Octal）

基数：8　　位权：8^i

数码O_i：0、1、2、3、4、5、6、7

位置表示法：$(N)_8=(O_{n-1}O_{n-2}...O_0 \bullet O_{-1}O_{-2}..)_8$

按权展开式：

$$(N)_8=O_{n-1}8^{n-1}+O_{n-2}8^{n-2}+...+O_08^0+O_{-1}8^{-1}+O_{-2}8^{-2}+...$$

例：

$(172.54)_8=$

$1\times8^2+7\times8^1+2\times8^0+5\times8^{-1}+4\times8^{-2}$

$=64+56+2+0.625+0.0625=(122.6875)_{10}$

4、十六进制数（Hexadecimal）

基数：16　　位权：16^i

数码H_i：0、1、2、3、4、5、6、7、8、9、
A、B、C、D、E、F
（10、11、12、13、14、15）

位置表示法：$(N)_{16}=(H_{n-1}H_{n-2}...H_0 \bullet H_{-1}H_{-2}..)_{16}$

按权展开式：

$$(N)_2=H_{n-1}16^{n-1}+H_{n-2}16^{n-2}+...+H_0 16^0+H_{-1}16^{-1}+H_{-2}16^{-2}+...$$

$(C07.A4)_{16}$= $(C07.A4)_H$= C07.A4H=

$12\times16^2+0\times16^1+7\times16^0+10\times16^{-1}+4\times16^{-2}$

$=3072+0+7+0.625+0.015625=(3079.640625)_{10}$

1.2.1 不同数制之间的转换

二进制、八进制、十六进制和十进制的数值关系表

十进制	二进制	八进制	十六进制	十进制	二进制	八进制	十六进制
0	0	0	0	8	1000	10	8
1	1	1	1	9	1001	11	9
2	10	2	2	10	1010	12	A
3	11	3	3	11	1011	13	B
4	100	4	4	12	1100	14	C
5	101	5	5	13	1101	15	D
6	110	6	6	14	1110	16	E
7	111	7	7	15	1111	17	F

一、八进制与二进制之间的转换

1、八进制转换为二进制

根据数值关系表用三位二进制数码逐位替代各位八进制数码。

例：　$(52.4)_8=(101010.1)_2$

2、二进制转换为八进制

将二进制数从小数点起，分别按整数部分和小数部分以三位数符划组，最高位和最底位不足部分补0。然后每组用一个八进制数符替代。

例：　$(1111101.0100111)_2=$

$(001111101.010011100)_2=(175.234)_8$

二、十六进制与二进制转换

1、十六进制转换为二进制

根据数值关系表用四位二进制数码逐位替代各位十六进制数码。

$(52.4)_{16}=(01010010.0100)_2=(1010010.01)_2$

2、二进制转换为十六进制

将二进制数从小数点起，分别按整数部分和小数部分以四位数符划组，最高位和最底位不足部分补0。然后每组用一个十六进制数码替代。

例：$(1111101.0100111)_2=$

$(01111101.01001110)_2=(7D.4E)_8$

三、 十进制数与非十进制数转换

转换条件:数值相等

1、非十进制数转换为十进制数

按权展开，多项式求和

2、十进制数转换为非十进制数

整数部分：

除基数取余数、从低位到高位求各位数码直到商为0。

小数部分：

乘基数取整数、从高位到低位求各位数码直到小数部分为0或满足精度要求。

转换原理：X进制数的多项式展开

$(N)_X=$

$$k_{n-1}x^{n-1}+k_{n-2}x^{n-2}+\ldots+k_0x^0+k_{-1}x^{-1}+k_{-2}x^{-2}+\ldots+k_{-m}x^{-m}$$

整数部分　※　小数部分

整数部分除以X: $(k_{n-1}x^{n-1}+k_{n-2}x^{n-2}+\ldots+k_1x^1+k_0x^0)/x$

$=(k_{n-1}x^{n-2}+k_{n-2}x^{n-3}+\ldots+k_1x^0)\ldots\ldots k_0$

第一次商　余数

第一次商/ x $=(k_{n-1}x^{n-3}+k_{n-2}x^{n-4}+\ldots+k_2x^0)\ldots\ldots k_1$

第二次商　余数

小数部分乘以X:

$(k_{-1}x^{-1}+k_{-2}x^{-2}+\ldots+k_{-m}x^{-m})\,x$

$= k_{-1}+(k_{-2}x^{-1}+\ldots+k_{-m}x^{-m+1})$

整数　第一次小数

第一次小数×X $= k_{-2}+(k_{-3}x^{-1}+\ldots+k_{-m}x^{-m+2})$

整数　小数

1、十进制数转换成二进制数

$(N)_2=$

$$B_{n-1}2^{n-1}+B_{n-2}2^{n-2}+\ldots+B_0 2^0+B_{-1}2^{-1}+B_{-2}2^{-2}+\ldots+B_{-m}2^{-m}$$

整数部分　※　小数部分

整数部分除以2: $(B_{n-1}2^{n-1}+B_{n-2}2^{n-2}+\ldots+B_1 2^1+B_0 2^0)/2$

$=(B_{n-1}2^{n-2}+B_{n-2}2^{n-3}+\ldots+B_1 2^0)$ ..... B_0

第一次商　余数

第一次商/ 2 $=(B_{n-1}2^{n-3}+B_{n-2}2^{n-4}+\ldots+B_2 2^0)$ ..... B_1

第二次商　余数

小数部分乘以2:

$(B_{-1}2^{-1}+B_{-2}2^{-2}+\ldots+B_{-m}2^{-m})\,2$

$= B_{-1}+(B_{-2}2^{-1}+\ldots+B_{-m}2^{-m+1})$

整数　第一次小数

第一次小数×2 $= B_{-2}+(B_{-3}2^{-1}+\ldots+B_{-m}2^{-m+2})$

整数　第二次小数

例1：$(11.625)_{10}=(B_{n-1}B_{n-2}\ldots B_1B_0 \bullet B_{-1}B_{-2}\ldots\ldots)_2$

整数部分：

整数部分除基数取余数、从低位到高位求各位数码直到商为0

商	余数	各位数码
11/2=5	1	B_0
5/2 =2	1	B_1
2/2 =1	0	B_2
1/2 =0	1	B_3

$(11)_{10}=(1011)_2$

小数部分：

小数部分乘基数取整数、从高位到低位求各位数码
直到小数部分为0或满足精度要求。

	取整数	各位数码
$0.625\times2=1.25$	1	B_{-1}
$0.25\times2=0.5$	0	B_{-2}
$0.5\times2=1$	1	B_{-3}

$(0.625)_{10}=(0.101)_2$

所以:$(11.625)_{10}=(1011.101)_2$

例1-3：$(0.562)_{10}=(B_{n-1}B_{n-2}\ldots B_1B_0 \bullet B_{-1}B_{-2}\ldots\ldots)_2$

误差不大于2^{-6}．即需要转换$B_{-1}B_{-2}B_{-3}B_{-4}B_{-5}$，$B_{-6}$以后的数码位权小于或等于$2^{-6}$，舍去。

	取整数	各位数码	位权
$0.562\times2=1.124$	1	B_{-1}	2^{-1}
$0.124\times2=0.248$	0	B_{-2}	2^{-2}
$0.248\times2=0.496$	0	B_{-3}	2^{-3}
$0.496\times2=0.992$	0	B_{-4}	2^{-4}
$0.992\times2=1.984$	1	B_{-5}	2^{-5}
$0.984\times2=1.968$	1	B_{-6}	2^{-6}
$0.968\times2=1.936$	1	B_{-7}	2^{-7}

$(0.562)_{10}=(0.100011)_2$　　误差$=\sum_{i=-7\sim-\infty} B_{-i}2^{-i} < 2^{-6}$

1.2.3码制

用0和1组合表示信息的编码形式

编码位数n和信息量N的关系：

$$N \leq 2^n$$

一、无符号数的自然二进制代码

n位码表示的数值范围：$0 \sim 2^n-1$

编码形式与二进制数完全相同，每位数码有位权的数值意义（有权码），但每组代码的位数确定 。

例：8位自然二进制码（表示的数值范围为0~255）

码：00000000, 00000101, 01111111, 10000000, 11111111,

数值：0 , 5 , 127 , 128 , 255

二、带符号二进制代码

n位二进制数值码（真值）加一位符号位构成机器数。

常用的带符号二进制代码：

原码（True Form）$[X]_{原}$

反码（One's Complement）$[X]_{反}$

补码（Two's Complement）$[X]_{补}$

最高位为符号位："0"表示正数，"1"表示负数。

正数的三种代码相同，都是数值码最高位加符号位"0"。

即$X \geq 0$时,真值与码值相等,且:$X=[X]_{原}=[X]_{反}=[X]_{补}$例：4位二进制数X=1101和Y=0.1101

$[X]_{原}=[X]_{反}=[X]_{补}=01101$，$[Y]_{原}=[Y]_{反}=[Y]_{补}=0.1101$

1、负数的二进制原码$[X]_{原}$。

原码表示方式：

n位数值码加最高位符号位“1”。

负整数的n+1位二进制原码值与真值X的关系：

$$[X]_{原} = 2^n - X = 2^n + |X|, \quad -2^n < X \leqslant 0$$

例： 4位二进制整数 $X = -1101$, $[X]_{原} = 11101$

负小数的原码值与真值X的关系：

$$[X]_{原} = 1 - X = 1 + |X|, \quad -1 < X \leqslant 0$$

$$[+0]_{原} = 0.000\ldots\ldots0, \quad [-0]_{原} = 1.000\ldots\ldots0$$

例： 4位二进制小数 $Y = -0.1101$, $[X]_{原} = 1.1101$

原码表示法的特点:

1、代码直观,求取方便,符号位加绝对值的二进制码。

2、 0有两组代码。

3、异号加运算步骤复杂，要判断符号和两数的绝对值大小。将绝对值大的数减去绝对值小的数，运算结果的符号位与绝对值大的数相同。

例： A=1101， B=-1001，C=0111， 求D=A+B，E=C+B

①$[A]_{原}$=01101 $[B]_{原}$=11001，因｜A｜＞｜B｜,D ＞ 0。

｜D｜=｜A｜-｜B｜= 1101-1001=0100，$[D]_{原}$= 00100.

② $[C]_{原}$=00111 ，因｜B｜＞｜C｜，所以E＜ 0

｜ E｜ = ｜B｜-｜C｜ = 1001-0111=0010，

$[E]_{原}$= 10010.

2、负数的二进制反码$[X]_{反}$

负整数反码表示方式：

n位真值码各位取反再加最高位符号位“1”。n+1位二进制反码值与真值X的关系：

$$[X]_{反} = 2^{n+1} - 1 + X, \quad -2^n < X \leqslant 0$$

例： 4位二进制整数

$X = -1101$, $[X]_{反} = 10010$

负小数反码表示方式：

n位数值码各位取反，整数位为符号位“1”，反码值与真值X的关系。

$$[X]_{反} = 2 - 2^{-n} + X, \quad -1 < X \leqslant 0$$

$[+0]_{反} = 0.000\ldots\ldots0$, $[-0]_{反} = 1.111\ldots\ldots1$

例： 4位二进制小数

$Y = -0.1101$, $[X]_{原} = 1.0010$

运用反码进行二进制整数减法运算：

$A-B=A+(-B)= A + X = D$，　$A、B > 0$，$X < 0$

运算结果： $[A]_{反}+[X]_{反}= A+ 2^{n+1}-1+X =D + 2^{n+1}-1$

运算后和的反码 $[D]_{反}= [A]_{反}+[X]_{反}$，与D的符号有关。

若 $A > B$ ，$D > 0$，　$[D]_{反}=D$

而加反码的运算和：$D + 2^{n+1}-1 \geq 2^{n+1}$ ，第n+2位=1，

运算和略去进位2^{n+1}再加1才等于D的反码。

若$A<B$，$D<0$，　$[D]_{反} = D+2^{n+1}-1$

而加反码的运算和：$D + 2^{n+1}-1 \leq 2^{n+1}$ ，第n+2位=0

运算和等于D的反码，不需处理（或加0）。

所以，电路进行整数反码加运算后必须对和再处理后才能得到真正的运算结果，方法：

将n+1位二进制加运算的进位信号再加到和的最低位。

反码表示法的特点:

1、负数的反码表示不太直观。

2、 0的反码有两组。

3、异号加运算步骤较简单，将最高位进位加到运算结果的最低位。

例1: A=1101, B=-1001, C=0111, 求D=A+B, E=C+B

① $[A]_{反}$=01101 $[B]_{反}$=10110,

因 |A| > |B|,D > 0

$[A]_{反}$+ $[B]_{反}$= 01101+10110=“1”00011,

$[D]_{反}$=00011+1= 00100 , D=0100 。

②$[C]_{反}$=00111 , 因 |C| < |B|,E < 0

$[C]_{反}$+ $[B]_{反}$= 00111+10110=“0”11101, $[E]_{反}$= 11101,

E = $[E]_{反}$-2^{n+1}+1= 11101-100000 +1= -0010 。

3、负数的二进制补码$[X]_{补}$

负整数补码表示方式：（反码加1）

n位数值码各位取反加1再加最高位符号位“1”。

n+1位二进制补码值与真值X的关系：

$$[X]_{补} = 2^{n+1} + X, \quad -2^n \leqslant X < 0$$

例：4位二进制整数

$X = -1101$，$[X]_{反} = 10010$，$[X]_{补} = 10011$

负小数补码表示方式：（反码加2^{-n}）

n位数值码各位取反加2^{-n}，整数位为符号位“1”。

补码值与真值X的关系：

$$[X]_{补} = 2 + X, \quad -1 \leqslant X < 0$$

$[0]_{补} = 0.000\ldots\ldots0$

例：4位二进制小数

$Y = -0.1101$，$[X]_{补} = 1.0011$

运用补码进行二进制整数减法运算：

$A-B=A+(-B)= A + X = D$，　$A、B > 0$，$X < 0$

运算结果： $[A]_{补}+[X]_{补}= A+ 2^{n+1}+X =D + 2^{n+1}$

运算后和的补码 $[D]_{补}= [A]_{补}+[X]_{补}$，与D的符号有关。

若 $A > B$ ，$D > 0$，　$[D]_{补}=D$

而加补码的运算和：$D + 2^{n+1} \geq 2^{n+1}$ ，第n+2位=1，

运算和略去进位2^{n+1}等于D的补码。

若$A<B$，$D<0$，　$[D]_{补} = D+2^{n+1}$

而加补码的运算和：$D + 2^{n+1} < 2^{n+1}$ ，第n+2位=0，

运算和等于D的补码。

所以，电路进行整数补码加运算后略去进位信号就是和的补码。

补码表示法的特点：

1、负数的补码表示不直观，求取不方便。

2、 0的补码只有一组。

3、异号加运算步骤最简单，舍去最高位进位即可。

例1： A=1101， B=-1001，C=0111， 求D=A+B，E=C+B

① $[A]_{补}$=01101 $[B]_{补}$=10111，

因 |A| > |B|，所以 D > 0 ， $[D]_{补}$= D

$[A]_{补}$+ $[B]_{补}$=01101+10111=“1” 00100，

$[D]_{补}$= 00100，D=0100

② $[C]_{补}$=00111 ，因 |C| < |B|，所以E < 0，

$[E]_{补}= 2^{n+1}+E$

$[C]_{补}$+ $[B]_{补}$= 00111+10111=“0” 11110

$[E]_{补}$=11110， E = $[E]_{补}- 2^{n+1}$ =11110-100000 = -0010

4位二进制典型数的真值、原码、反码和补码

X 真值	[X]原	[X]反	[X]补	X 真值	[X]原	[X]反	[X]补
+1001	01001	01001	01001	-0.0000	1.0000	1.1111	0.0000
+0001	00001	00001	00001	-0.0010	1.0010	1.1101	1.1110
+0.1101	0.1101	0.1101	0.1101	-0011	10011	11100	11101
+0.0000	0.0000	0.0000	0.0000	-1010	11010	10101	10110

负数的原、反、补码关系对照表

	原码	反码	补码
整数编码形式	符号 1、数值码	符号 1、数值码各位取反	反码加 1
表示范围	$-2^n<X\leqslant 0$	$-2^n<X\leqslant 0$	$-2^n\leqslant X<0$
码值与真值的关系	$[X]_{原}=2^n-X$	$[X]_{反}=2^{n+1}-1+X$	$[X]_{补}=2^{n+1}+X$
小数编码形式	符号 1、数值码	符号 1、数值码各位取反	反码加 2^{-n}
表示范围	$-1<X\leqslant 0$	$-1<X\leqslant 0$	$-1\leqslant X<0$
码值与真值的关系	$[X]_{原}=1-X$	$[X]_{反}=2-2^{-n}+X$	$[X]_{补}=2+X$
码值相加结果 处理为相同编码	复杂	将进位信号加至最低位 保留n+1位	去掉进位 保留n+1位

4位二进制负整数的原、反、补码对照表

十进制数	0	-0	-1	-2	-3	-4	-5	-6	-7
真值	0000	-0000	-0001	-0010	-0011	-0100	-0101	-0110	-0111
原码	00000	10000	10001	10010	10011	10100	10101	10110	10111
反码	00000	11111	11110	11101	11100	11011	11010	11001	11000
补码	00000	-----	11111	11110	11101	11100	11011	11010	11001
十进制数	-8	-9	-10	-11	-12	-13	-14	-15	-16
真值	-1000	-1001	-1010	-1011	-1100	-1101	-1110	-1111	-10000
原码	11000	11001	11010	11011	11100	11101	11110	11111	------
反码	10111	10110	10101	10100	10011	10010	10001	10000	------
补码	11000	10111	10110	10101	10100	10011	10010	10001	10000

三、二——十进制编码（Binary Code Decimal码）

用4位二进制码表示十进制数符

从0000~1111十六组码中取十组代表0~9十个数符

主要有：

1、8421码：四位码都有位权，各为8，4，2，1。

2、5421码：四位码都有位权，各为5，4，2，1。

3、2421码：四位码都有位权，各为2，4，2，1。

4、余3码：各位码没有位权值，但各组二进制码值比其表示的十进制数符值多3。

前三种为有权码，后一种为无权码。

十进制	8421_{BCD}	2421_{BCD}	5421_{BCD}	余3码
0	0000	0000	0000	0011
1	0001	0001	0001	0100
2	0010	0010	0010	0101
3	0011	0011	0011	0110
4	0100	0100	0100	0111
5	0101	0101	1000	1000
6	0110	0110	1001	1001
7	0111	0111	1010	1010
8	1000	1110	1011	1011
9	1001	1111	1100	1100
六组伪码	1010~1111	1000~1101	0101~0111 1101~1111	0000~0010 1101~1111

计数体制为十进制方式，第i组码的位权为10^i。

例：$(271.59)_{10}$

$=(001001110001.01011001)_{8421BCD}$

$=(001001110001.\ 01011111)_{2421BCD}$

$=(001010100001.10001100)_{5421BCD}$

$=(010110100100.10001100)_{余3BCD}$

特

1、

2、

3、

4、

十进制数	二进制码	典型格雷码	格雷码 1（·BCD）	格雷码 2（BCD）	步进码
0	0000	0000	0000	0000	00000
1	0001	0001	0001	0001	00001
2	0010	0011	0011	0011	00011
3	0011	0010	0010	0010	00111
4	0100	0110	0110	0110	01111
5	0101	0111	1110	0111	11111
6	0110	0101	1010	0101	11110
7	0111	0100	1011	0100	11100
8	1000	1100	1001	1100	11000
9	1001	1101	1000	1000	10000
10	1010	1111			
11	1011	1110			
12	1100	1010			
13	1101	1011			
14	1110	1001			
15	1111	1000			

5、步进格雷码的特点：

是表示十进制数符的十组BCD 码。每一组码的最高位取反后移到最低位，即构成其相邻高一数序符的步进码组。

6、n位二进制码与n位典型格雷码的关系：

①最高位相同，即$G_{n-1}=B_{n-1}$；

②其余各位格雷码由与其位序相同的二进制码及高一位的二进制码决定：

当两位二进制码相同时，该位格雷码为0，不同时为1，即$G_i=B_{i+1} \oplus B_i$，$i \neq n-1$。

例：$(0110)_2=(0101)_G$，$(1110)_2=(1001)_G$

十进制数	二进制码	典型格雷码	格雷码 1（·BCD ）	格雷码 2（ BCD ）	步进码
0	0000	0000	0000	0000	00000
1	0001	0001	0001	0001	00001
2	0010	0011	0011	0011	00011
3	0011	0010	0010	0010	00111
4	0100	0110	0110	0110	01111
5	0101	0111	1110	0111	11111
6	0110	0101	1010	0101	11110
7	0111	0100	1011	0100	11100
8	1000	1100	1001	1100	11000
9	1001	1101	1000	1000	10000
10	1010	1111			
11	1011	1110			
12	1100	1010			
13	1101	1011			
14	1110	1001			
15	1111	1000			

五、文

ASCI 位二进制

1、十 为011，A

2、大 码为41

3、小 码为61

4、0

5、其

ASCII 编码文字符的含义

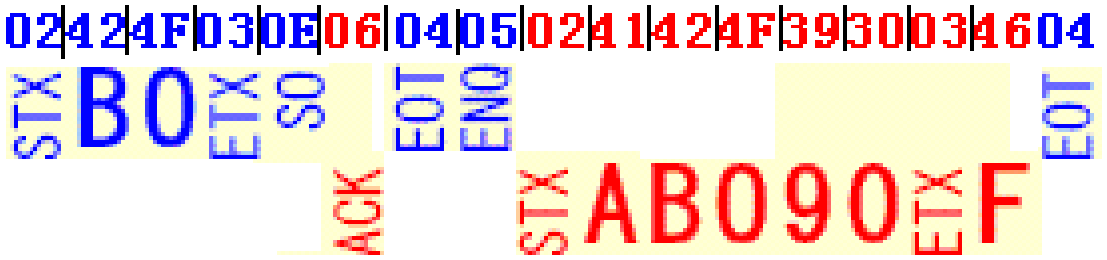
字 符	含 义	字 符	含 义
NUL	空，无效	DLE	数据键换码
SOM	标题开始	DC1	设备控制 1
STX	正文开始	DC2	设备控制 2
ETX	文本开始	DC3	设备控制 3
EOT	传输结束	DC4	设备控制 4
ENQ	询　　问	NAK	否　　定
ACK	承　　认	SYN	空转同步
BEL	报警符（可听见的信号）	ETB	信息组传输结束
BS	退一格	CAN	作　　废
HT	横向列表（穿孔卡片指令）	EM	纸　　尽
LF	换　　行	SUB	减
VF	垂直制表	ESC	换　　码
FF	走纸控制	FS	文字分隔符
CR	回　　车	GS	组分隔符
SO	移位输出	RS	记录分隔符
SI	移位输入	US	单元分隔符
SP	空间（空格）	DEL	作　　废

$b_{3\sim0}$ \ $b_{6\sim4}$	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	\	p
0001	SOM	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	,	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VF	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	!
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	↑	n	~
1111	SI	US	/	?	O	↓	o	DEL

ASCII码

CPU发送命令

读、制券机应答



1.3逻辑函数

一、逻辑函数的基本概念

逻辑变量：

描述具有两个对立状态的某一对象，分析时用“0”和“1”表示这两个状态，在电路中用高电平V_H和低电平V_L表示。

对于具体问题必须先规定对象的变量取值。

数字电路：

处理数字信号的电路，所有元件的输入、输出都只有高电平V_H和低电平V_L两种状态.

逻辑函数： $F=f\ (A_0、A_1、\ldots、A_{n-1})$

描述影响某一逻辑事件的诸条件间的关系

函数值F和变量值A_i都只有两种取值“0”、“1”

二、逻辑函数的表示方法

1、真值表

以表格形式列出所有变量取值所对应的函数值。n个变量有2^n种取值组合，以自然二进制码递增的方式排列。

ABC	F
000	0
001	1
010	1
011	0
100	1
101	0
110	0
111	1

2、卡诺图（真值表的方格图形式）

变量分为行、列两组，以格雷码形式排列在图旁，函数值填在格内。

F: A \ BC	00	01	11	10
0	0	1	0	1
1	1	0	1	0

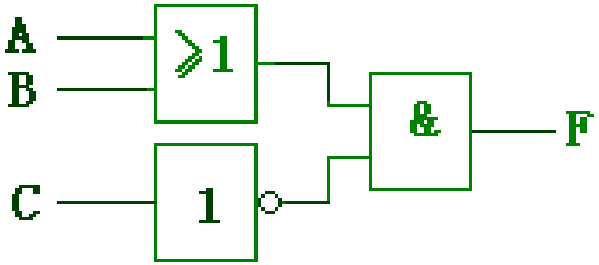
3、逻辑代数表达式

用三种基本布尔运算符“·”、“+”“-”和两种关系判断符“⊙”、“⊕”符描述的表达式。

例：$F=A\,B+C\,D$

4、逻辑图

用逻辑符号表示的信号传输关系



5、硬件描述语言（Hard Description Language）

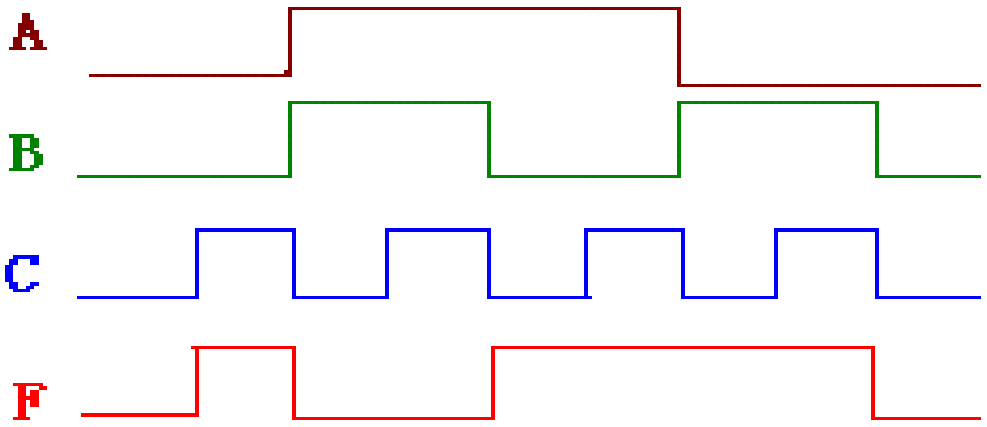
类似计算机软件编程语言的形式

常用的有ABLE-HDL、VHDL和Verilog HDL

波形图:

输出信号（函数值）与输入信号（函数变量）的时序对应关系图。

根据电路的逻辑关系，对照输入信号（如A、B、C）波形画出输出（如F）波形。



1.3.1基本逻辑运算关系

与-逻辑乘：$F=A_0 \cdot A_1 \cdot A_2 \cdot \ldots \cdot A_{n-1}$

或-逻辑加：$F=A_0+A_1+A_2+\ \ldots+A_{n-1}$

非-逻辑反：$F=\ \overline{A}$

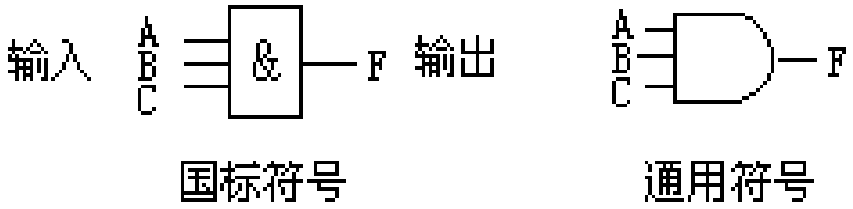
与运算：　所有变量都为“1”时函数值为“1”

三输入与门：F = A B C　　逻辑乘

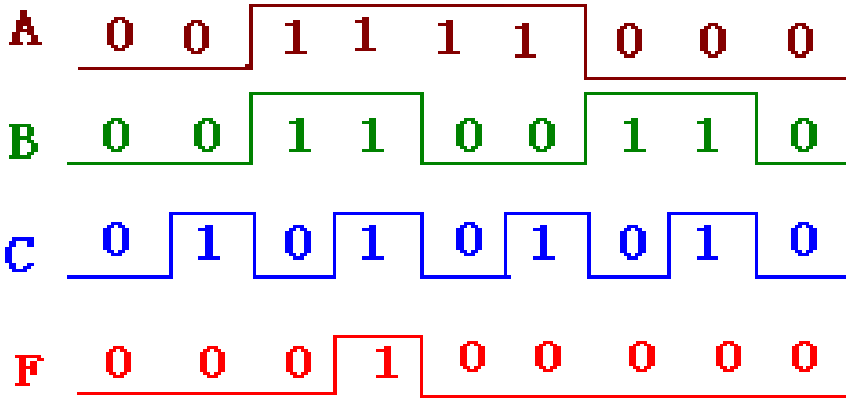
真值表

A B C	F
0 0 0	0
0 0 1	0
0 1 0	0
0 1 1	0
1 0 0	0
1 0 1	0
1 1 0	0
1 1 1	1

逻辑符号　　全“1”出“1”，有“0”出“0”



波形图



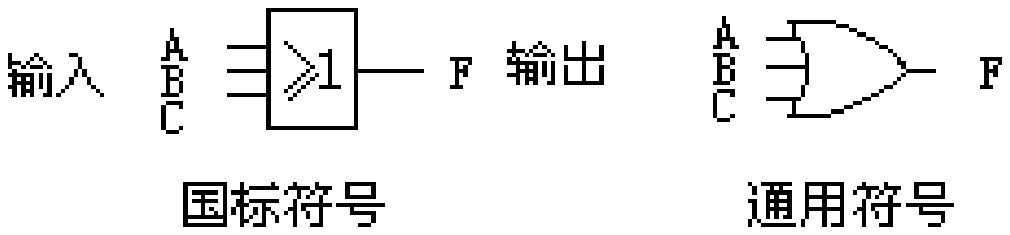
或运算：　只要有一个变量为“1”，函数值为“1”

三输入或门：F = A +B +C　　逻辑加

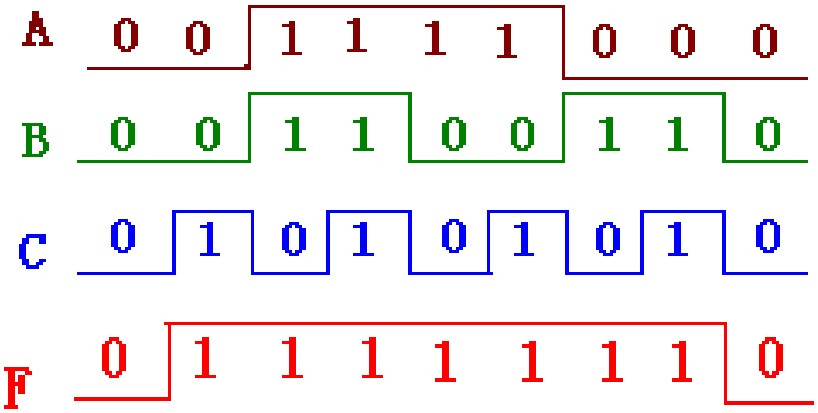
真值表

A B C	F
0 0 0	0
0 0 1	1
0 1 0	1
0 1 1	1
1 0 0	1
1 0 1	1
1 1 0	1
1 1 1	1

逻辑符号　有“1”出“1”，全“0”出“0”



波形图



非运算:

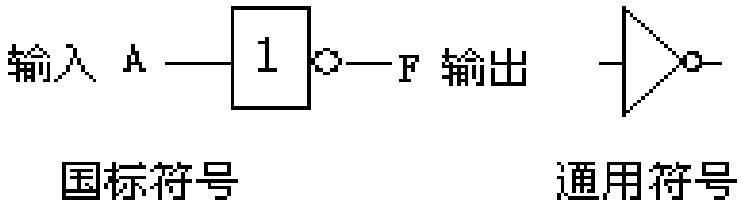
函数值与变量值相反.输出与输入电平相反

$F=\overline{A}$　　$A=0,\ \overline{A}=1;\ A=1,\ \overline{A}=0$

原变量A, 反变量$\overline{A}$

当A=1, 原变量为1, 当A=0, 反变量为1.

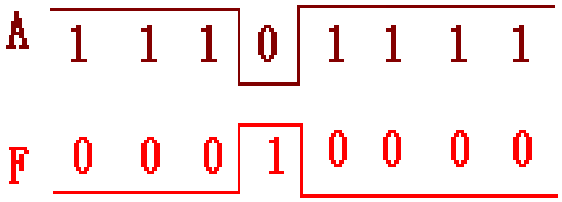
逻辑符号



国标符号　　通用符号

逻辑图中用圈表示反相运算

波形图

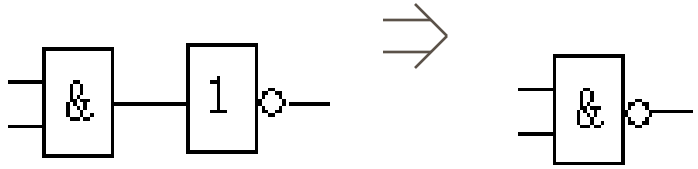


复合运算

与非运算：先“与”后“非”

非门

$F=A_0 A_1 A_2 \cdot \ldots \cdot A_{n-1}$

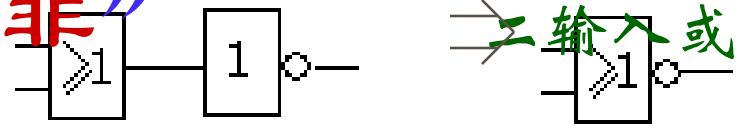


二输入与非门

或非运算：先“或”后“非”

非门

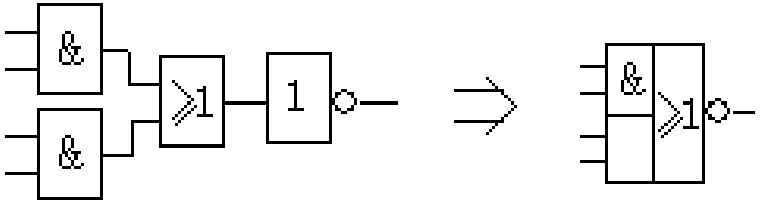
$F=A_0+A_1+A_2+ \ldots +A_{n-1}$



二输入或非门

与或非运算：先“与”后“或非”门

$F=A_0A_1+B_0B_1 \ldots$



异或运算：

两个输入相异时输出为“1”，相同时输出为“0”。

$A \oplus 0 = A$　$A \oplus \overline{1} = A$

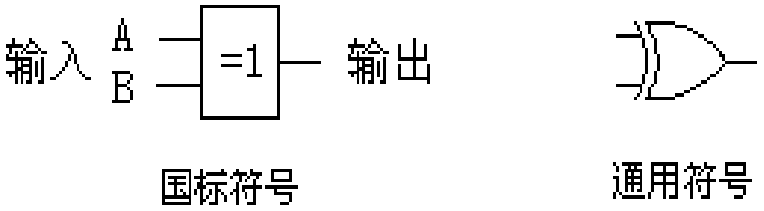
$$F = A \oplus B = \overline{A}B + A\overline{B}$$

$A \oplus \overline{A} = 1$　$A \oplus A = 0$

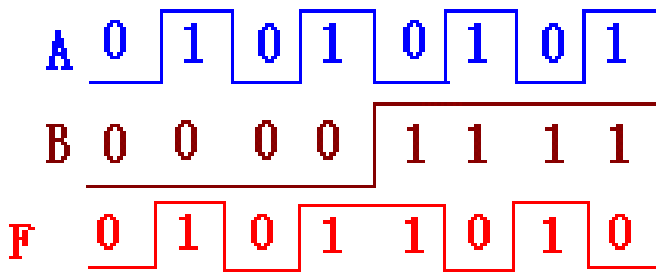
真值表

A	B	F
0	0	0
0	1	1
1	0	1
1	0	1
1	1	0

逻辑符号



波形图



同或运算：

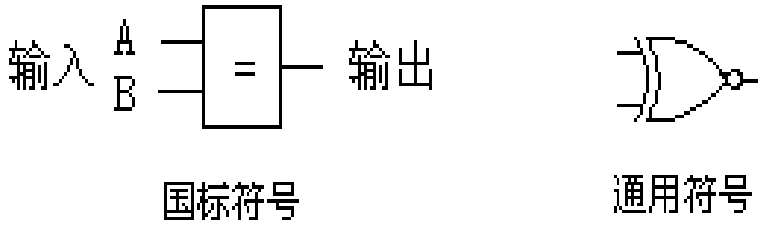
两个输入相异时输出为“0”，相同时输出为“1”。

$$F = A \odot B = \bar{A}\bar{B} + AB = \overline{A \oplus B}$$

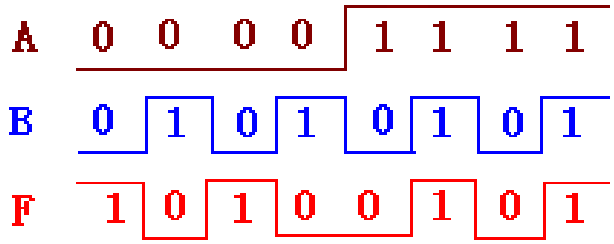
真值表

A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

逻辑符号



波形图



1.3.2逻辑代数的基本运算和基本定律

两个基本规则

1、代入规则：

当逻辑等式中两边的某变量用相同的逻辑函数式代替时，等式仍成立。

2、对偶规则：

当两个逻辑式相等时，它们个自的对偶式也相等。

逻辑恒等式的对偶式：

将原式中的逻辑常量“0”–“1”对换；“与”–“或”运算

关系对换，保持运算顺序不变，即得原式的对偶式。

一、基本定律

0-1律	$A+0=A$	$A\cdot 1=A$
	$A+1=1$	$A\cdot 0=0$
重叠律	$A+A=A$	$A\cdot A=A$
互补律	$A+\overline{A}=1$	$A\cdot \overline{A}=0$
结合律	$(A+B)+C=A+(B+C)$	$(AB)C=A(BC)$
交换率	$A+B=B+A$	$AB=BA$
分配律	$A(B+C)=AB+AC$	$A+BC=(A+B)(A+C)$
摩根定律	$\overline{A+B+C}=\overline{A}\,\overline{B}\,\overline{C}$	$\overline{ABC}=\overline{A}+\overline{B}+\overline{C}$
反演律	函数的变量取反、常量（0、1）取反、与或关系对换，得其反函数。	
否定律	$\overline{\overline{A}}=A$	

二、基本定理

定理1 $A+AB=A$ $A(A+B)=A$

定理2 $A+\bar{A}B=A+B$ $A(\bar{A}+B)=AB$

定理3 $AB+\bar{A}C+BC=AB+\bar{A}C$ 冗余项可消去

$(A+B)(\bar{A}+C)(B+C)=(A+B)(\bar{A}+C)$

三、有关异或运算的公式

交换率 $A \oplus B = B \oplus A$

结合律 $(A \oplus B) \oplus C = A \oplus (B \oplus C)$

奇偶律： $A_0 \oplus A_1 \oplus A_2 \oplus A_3 \oplus \cdots\cdots \oplus A_{n-1} = ?$

若n个输入变量中有奇数个“1”异或结果为“1”;

若有偶数个“1”结果为“0”。

1.3.3逻辑函数表示方法的相互转换

1、由真值表写函数的逻辑表达式

ⅰ将每组使函数值为“1”的变量取值组合写成一个与项(最小项)，其中变量取值“1”的写原变量，取值“0”的写反变量；

ⅱ将所有的与项相加得原函数的标准与或表达式(最小项表达式)。

例：写出右表两个函数的逻辑表达式。

解：

$$F1=\bar{A}\bar{B}C+\bar{A}B\bar{C}+\bar{A}BC+ABC$$

$$F2=\bar{A}\bar{B}C+\bar{A}B\bar{C}+A\bar{B}\bar{C}+ABC$$

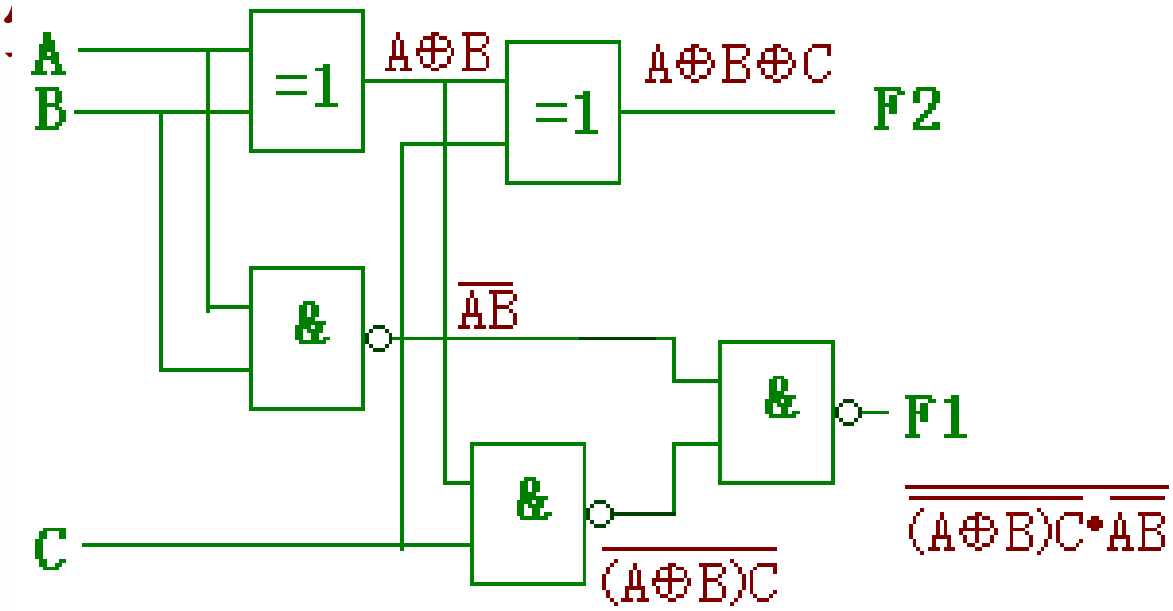
真值表

ABC	F1	F2
000	0	0
001	1	1
010	1	1
011	1	0
100	0	1
101	0	0
110	0	0
111	1	1

2、由逻辑图写函数逻辑表达式

按信号的传输路径从输入到输出逐级写每个逻辑图形符号对应的运算关系，得函数的逻辑表达式.

例：



3、由逻辑表达式列真值表

将输入变量的所有取值组合代入逻辑表达式，求函数值。

例：列出下列函数的真值表

$F1=AB+BC+AC$

$F2=A \oplus B \oplus C$

解：F1、F2都是3变量函数，列出真值表的8个变量取值组合，代入函数式，将求得的函数值填入表内。

真值表

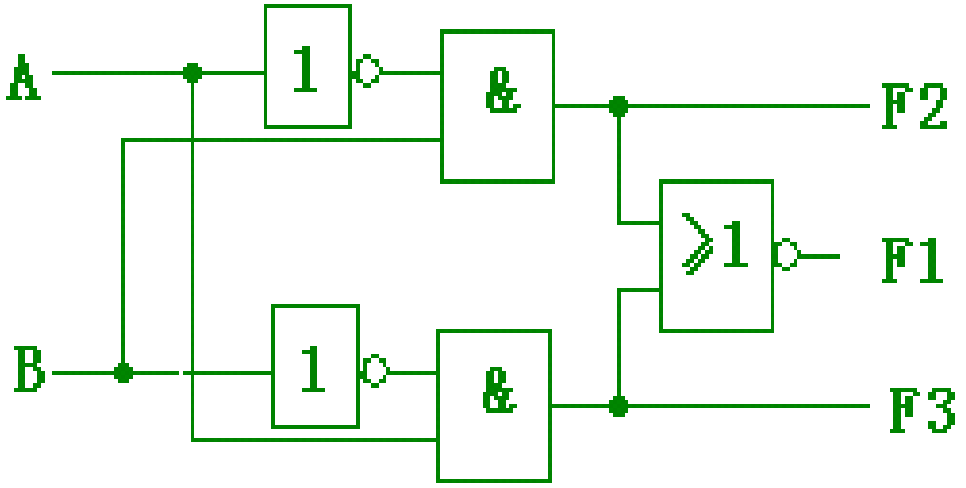
ABC	F1	F2
0 0 0	0	0
0 0 1	0	1
0 1 0	0	1
0 1 1	1	0
1 0 0	0	1
1 0 1	1	0
1 1 0	1	0
1 1 1	1	1

4、由逻辑表达式画逻辑图

用逻辑符号替代表达式中的逻辑运算关系符

例：画出下列三个函数表达式的逻辑图。

$$F1=\overline{\bar{A}B+A\bar{B}}，F2=\bar{A}B，F3=A\bar{B}$$



1.3.4逻辑函数的化简

化简要求

要求1、逻辑表达式最简（器件最少，速度最快）

要求2、逻辑运算关系统一（器件型号统一）

化简目标：最简与或表达式——

乘积项最少且乘积项中变量因子最少。

逻辑表达式的类型：

$$Z=\overline{A\bar{C}+\bar{A}\bar{B}}=\overline{\overline{A}+C}+\overline{A+B}=(\bar{A}+C)(A+B)$$

与或非　　或非-或非　　或与

$$=A\bar{A}+AC+\bar{A}B+BC=AC+\bar{A}B+BC=AC+\bar{A}B=\overline{\overline{AC}\cdot\overline{\bar{A}B}}$$

冗余　　与或　　与非-与非

例　以下4个“与或”表达式是相等的，即它们表示同一个函数：

$Z = A\bar{C} + B\bar{C} + \bar{A}B + \bar{A}C$　　(1)　$B\bar{C}+\bar{A}C+\bar{A}B=B\bar{C}+\bar{A}C$

$= A\bar{C} + \bar{A}B\bar{C} + \bar{A}C$　　(2)　$A\bar{C}+B\bar{C}=(A+B)\bar{C}=(A+\bar{A}B)\bar{C}=A\bar{C}+\bar{A}B\bar{C}$

$= A\bar{C} + B\bar{C} + \bar{A}C$　　(3)

$= A\bar{C} + \bar{A}B + \bar{A}C$　　(4)　$\bar{A}B\bar{C}+\bar{A}C=\bar{A}(B\bar{C}+C)=\bar{A}B+\bar{A}C$

试判断哪一个是最简“与或”表达式。

解：①对比可知式1含4个与项，其他3式都只含3个与项，所以式1肯定不是最简；②式3、4中各与项都含2个变量，而式2中有一个与项含3个变量。结论：式3、4同为该函数的最简与或表达式。

一、逻辑函数的公式法化简：

并项法： 利用$A+\overline{A}=1$并项，消变量。

例： $F=AB\overline{C}+ABC=AB(C+C)=AB$

吸收法：利用$A+AB=A$并项，消变量。

例： $F=AB+ABCD(E+F)=AB(1+CDE+CDF)=AB$

消去法：利用$A+\overline{A}B=A+B$，消变量。

例： $F=AB+\overline{A}C+\overline{B}C=AB+(\overline{A}+\overline{B})C=AB+\overline{AB}C=AB+C$

配项法：利用$A=A(B+\overline{B})$配项，消去其他项的变量。

例： $F=AB+\overline{A}C+BC=AB+\overline{A}C+(A+\overline{A})BC$

$=AB+ABC+\overline{A}C+\overline{A}BC=AB(1+C)+\overline{A}C(1+B)$

$=AB+\overline{A}C$

二、逻辑函数的最小项和标准与或表达式

表1-3-6　　三变量A、B、C的8个最小项及各自的真值表

序号	最小项编号 / 变量取值			m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7
	A	B	C	$\bar{A}\bar{B}\bar{C}$	$\bar{A}\bar{B}C$	$\bar{A}B\bar{C}$	$\bar{A}BC$	$A\bar{B}\bar{C}$	$A\bar{B}C$	$AB\bar{C}$	ABC
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0
5	1	0	1	0	0	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	0	1

最小项为“1”，两者有一一对应的关系；

任两个最小项的乘积为“0”；

所有最小项之和为1。

3、最小项符号 m_i

序号i=0~2^n-1 ,是最小项对应的变量取值组合（原变量取1，反变量取0，按变量排序组成的二进制数对应）的十进制值。所以写最小项符时必须规定变量的排列位序。

4、标准与或表达式（最小项表达式）

由函数值为1的变量取值组合对应的最小项相加构成的与或表达式。

5、最小项和式$\sum m_i$

用最小项符m_i构成的逻辑表达式，函数值用括号说明变量位序排列。

$F(A, B, C, \cdots) = \sum m_i$

6、标准与或表达式的求取

①由真值表求标准与或表达式

将使函数值为1的所有变量取值对应的最小项的相或得标准与或表达式。

写最小项时，变量取值“0”的写反变量，取值“1”的写原变量。

②由一般与或表达式写标准与或表达式

利用$A=A(B+\overline{B})=AB+A\overline{B}$对非最小项的与项配缺失变量因子，构成全最小项的表达式。

- 例：$AB+BC=AB(C+\overline{C})+BC(A+\overline{A})$
- $=ABC+AB\overline{C}+ABC+\overline{A}BC=ABC+AB\overline{C}+\overline{A}BC$

例：根据函数真值表写标准与或表达式

解:将真值表中函数值为1的变量取值组合对应的最小项相或，得标准与或表达式。写最小项时，变量取值0的写反变量，取值1的写原变量。

$$F=\bar{A}\bar{B}\bar{C}+\bar{A}\bar{B}C+\bar{A}B\bar{C}+A\bar{B}\bar{C}$$

$$F（A、B、C）$$

$$= m_0 + m_1 + m_2 + m_4$$

$$=\sum m（0,1,2,4）$$

最小项数与真值表中函数为“1”的项数相同。

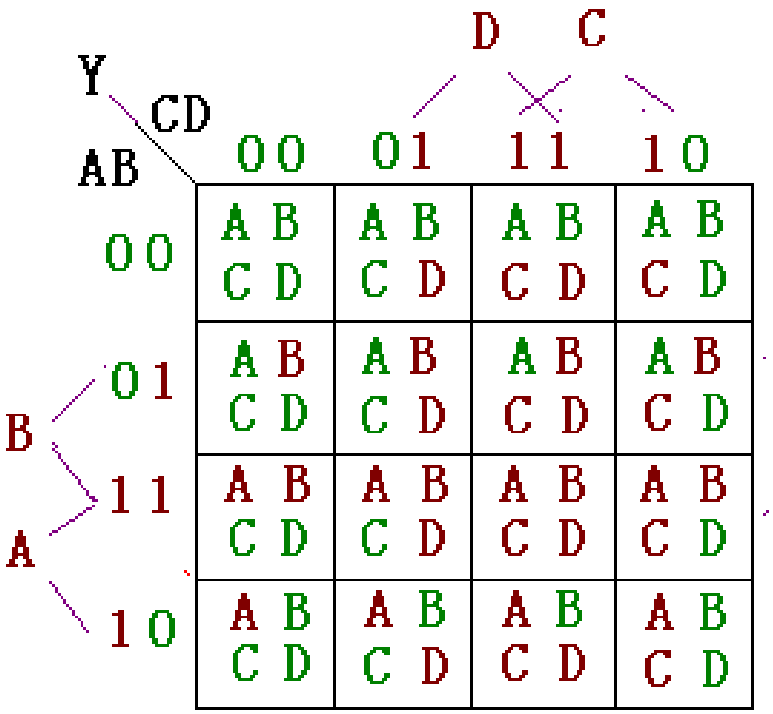
变量取值	最小项	最小项符	函数值
A B C		m_i	F
0 0 0	$\bar{A}\bar{B}\bar{C}$	m_0	1
0 0 1	$\bar{A}\bar{B}C$	m_1	1
0 1 0	$\bar{A}B\bar{C}$	m_2	1
0 1 1	$\bar{A}BC$	m_3	0
1 0 0	$A\bar{B}\bar{C}$	m_4	1
1 0 1	$A\bar{B}C$	m_5	0
1 1 0	$AB\bar{C}$	m_6	0
1 1 1	ABC	m_7	0

三、卡诺图----结构和特点

①将变量分为行、列两组，变量取值按典型格雷码排列，相邻列（行）之间只有一个变量取值不同。

②具有循环邻接性。

③卡诺图的每个格代表了函数的一个最小项。



A、B、C、D 取值 1

A、B、C、D 取值 0

四、用卡诺图表示逻辑函数

1、由真值表写其卡诺图

将各变量组合对应的函数值填入相应的卡诺图格中。

2、由函数表达式写其卡诺图

①由标准与或表达式写卡诺图

将表达式中出现的最小项所对应的卡诺图格中填入**1**，其余格填**0**。

②由非标准与或表达式写卡诺图

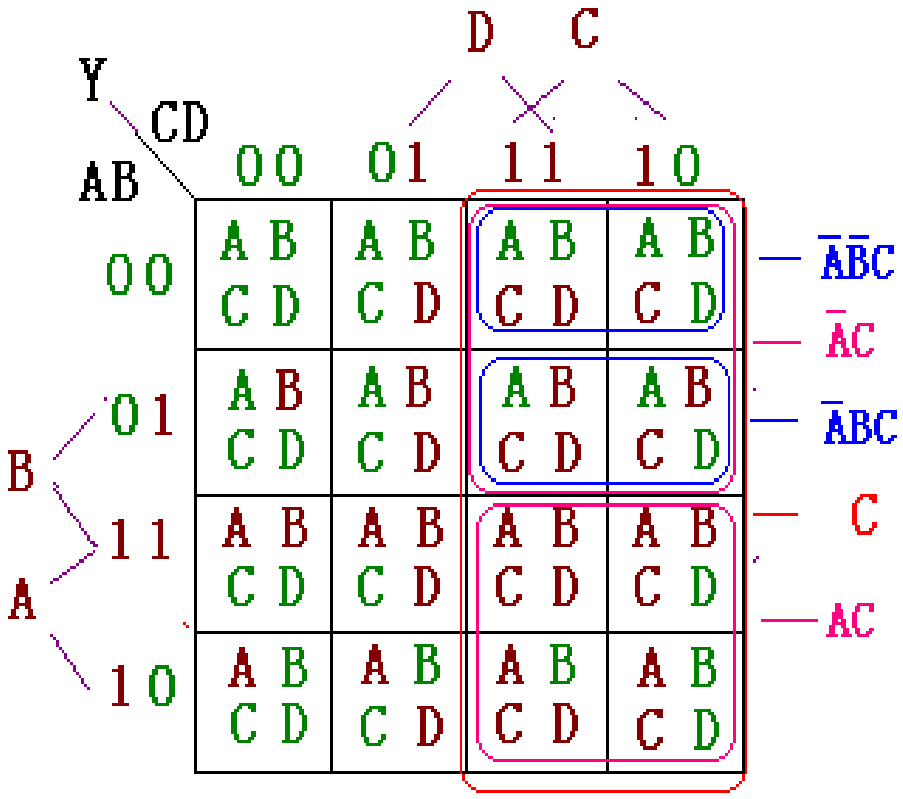
将函数转换成与或表达式，在每个乘积项的变量取值范围内填入**1**，其余格填**0**。

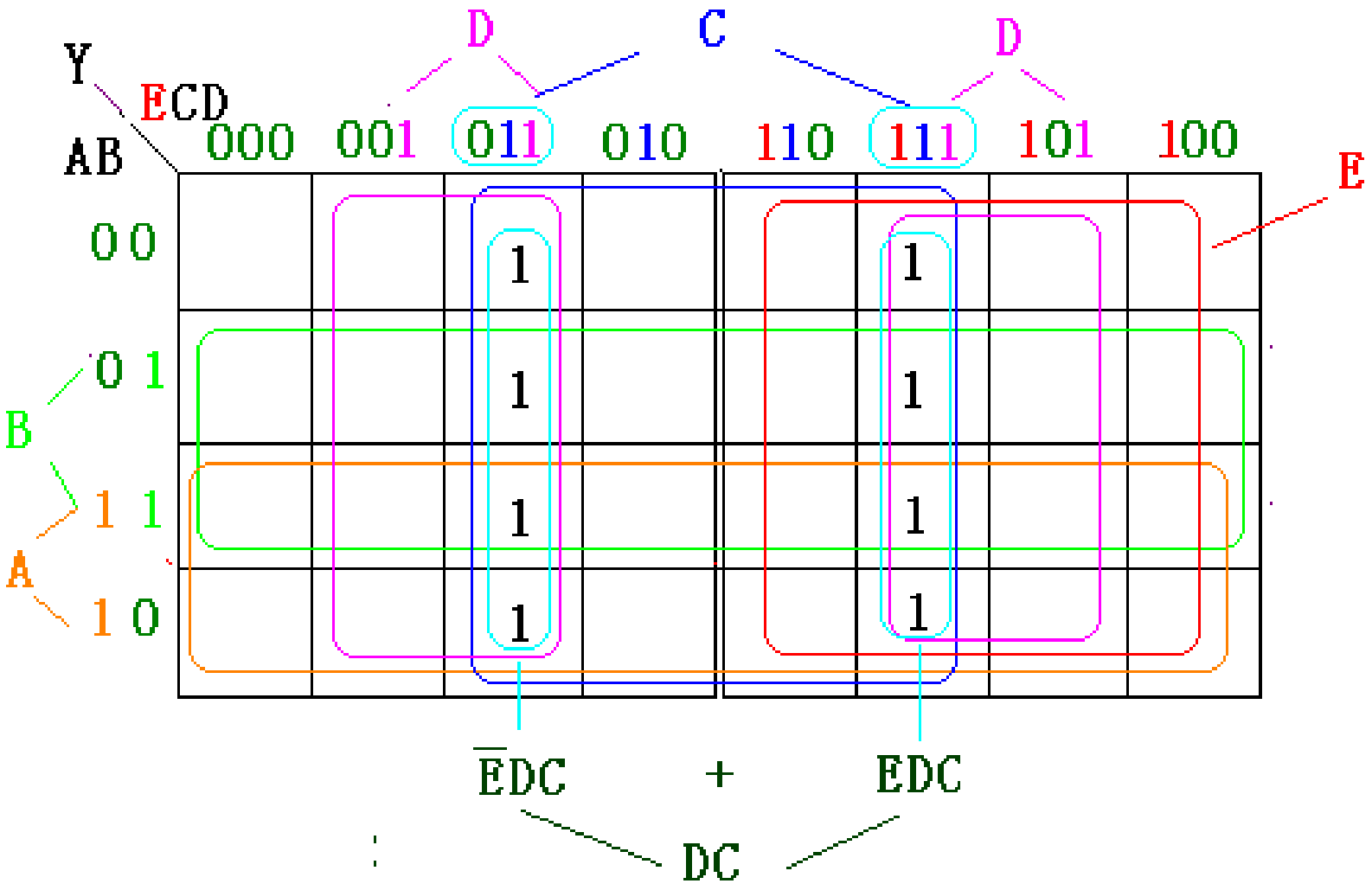
五、用卡诺图化简逻辑函数的依据

相邻两个最小项相或合并时可以消去一个取值不同的变量。

相邻列合并消去列变量。

相邻行合并消去行变量。




D
C
D
Y
ECD
AB
000
001
011
010
110
111
101
100
E
00
01
11
10
B
A
1
1
1
1
1
1
1
1
$\overline{E}DC$
+
EDC
DC

六、用卡诺图化简逻辑函数的步骤和规则

(1)以矩形圈形式合并2^n个函数值(为1)相同的卡诺图格，消去取值不同的变量，形成一个乘积项。

(2) 圈从大到小，直到所有函数值相同(为1)的格全部圈过。但每个圈中必须至少包含一个没有被其它圈包围的独立格。

(3)圈尽可能大，使乘积项的变量因子尽可能少。

圈尽可能少，使乘积项的个数尽可能少。

(4)所有乘积项之逻辑和为原函数(值为1)或反函数(值为0)的最简与或表达式。

七、具有无关项的逻辑函数表示方法

1、无关项（任意项、伪码）

对函数值没有影响的变量取值组合所对应的最小项，用符号Φ表示其函数值。（如BCD码中的伪码组合），用Φ_i表示，i取值同最小项。

2、具有无关项的逻辑函数最小项表达式

$$f(A、B、C.....)=\sum m_i+\sum \Phi_i$$

3、具有无关项的逻辑函数卡诺图

在无关项格中-填入Φ（或X、-、d），表示函数值任意。

4、具有无关项逻辑函数的化简

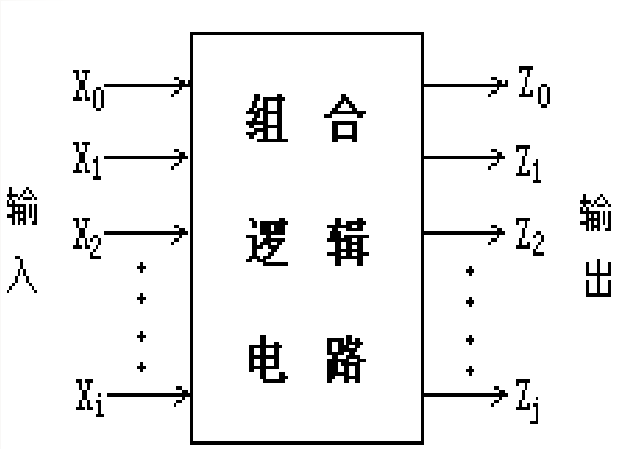
无关项可以根据合并圈扩大的化简要求任意取值“0”或“1”，但不必全部圈入。

数字电子技术基础

第三章　　组合逻辑电路

组合逻辑的电路结构：

信号从输入端逐级向输出传输，没有后级向前级的反馈。



$$Z1 = f1（X1，X2，…，Xi）$$
$$Z2 = f2（X1，X2，…，Xi）$$
$$\vdots$$
$$Zj = fj（X1，X2，…，Xi）$$

组合逻辑的电路特点：

任何时刻电路的输出状态只与当前输入信号的状态有关，与电路原来的输出状态无关，没有记忆功能。

3.1组合逻辑分析

任务：

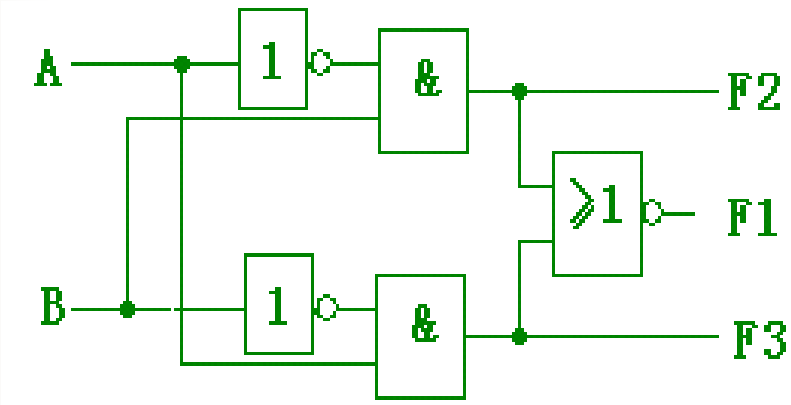
根据电路图分析其输入、输出关系，确定电路功能。

方法：

1、根据电路图从输入到输出逐级写逻辑表达式；化简后分析电路功能。

2、如果从表达式不能直接分析电路功能，可列真值表确定。

例：分析下图电路的三个输出各对两个输入的一位二进制数A、B 实现什么逻辑判断功能。



真值表

A B	F1	F2	F3
0 0	1	0	0
0 1	0	1	0
1 0	0	0	1
1 1	1	0	0

解：1、由图列表达式

2、列真值表

3、分析逻辑功能

$F1=\overline{\bar{A}B+A\bar{B}}$，$F2=\bar{A}B$，$F3=A\bar{B}$

输出为表示两个输入比较结果的开关量：

F1表示A=B；　F2表示A＜B；F3表示A＞B。

电路为一位二进制数比较器，输出A大于B、A小于B、A=B三种判断结果。

开关量：

表示特定功能的一个变量。当功能实现时变量为有效电平。

高电平有效：功能实现时变量为1，

低电平有效：功能实现时变量为0。

N种功能需要N个变量描述。

编码量：

以n个变量的一组特定组合（二进制码）共同表示一个功能。N种功能与n个变量的关系满足：$2^{n-1} < N \leq 2^n$。没有有效、无效之分。

例：一位二进制数比较器可以用2个变量的组合表示三种较结果。

3.2组合逻辑电路的设计

根据设计任务求实现逻辑功能的电路

3.2.1采用逻辑门设计组合逻辑电路的步骤:

1、分析任务要求，确定输入、输出变量及逻辑定义。

2、根据逻辑问题的因果关系写逻辑表达式或列函数真值表，写标准与或表达式。

3、化简逻辑函数得最简表达式或变换逻辑关系得满足设计要求的表达式形式。

4、根据表达式画逻辑图，并检查电路的驱动或时间延迟等是否符合工程要求。

3.2.3含无关项的组合逻辑电路设计

1、不拒伪码

利用对函数值没有影响的无关项化简函数使电路更简。（无关项对应的函数值可以为1或0）。变量输入为无关项时输出可能为 1或为0 。

2、拒伪码

无关项对应的函数值必须为0。变量输入为无关项时输出一定为0。

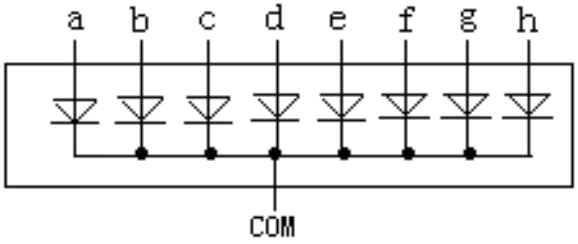
3.2.4多输出函数的组合逻辑电路设计

在化简函数时保留各函数的公共项部分，以使整个电路形式最简。

七段显示器的结构——由8个发光二极管构成

共阴显示器

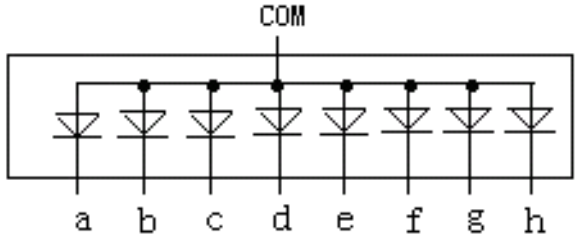
控制端接高电平，段点亮



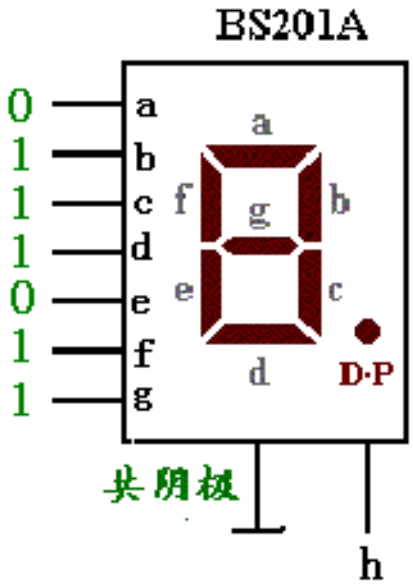
公阴极接低电平（0）

共阳显示器

公阳极接高电平（1）



控制端接低电平，段点亮



3.3常用中规模组合逻辑标准构件

集成电路规模的划分

小规模集成电路SSI—— 器件集成。如与非门等

中规模集成电路MSI—— 构件集成。如数据选择器、译码器、编码器、计数器、寄存器等。

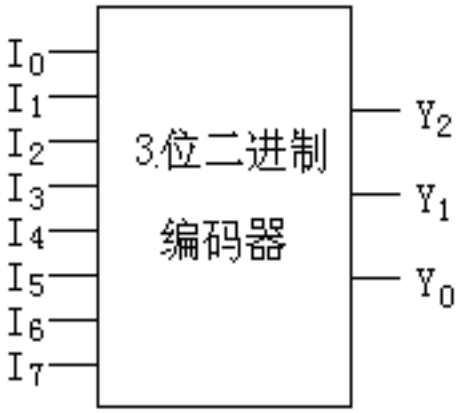
大规模集成电路LSI——子系统集成。LCD控制器、按键显示扫描管理器等。

超大规模集成电路VLSI——系统集成。

单片机、处理器（CPU）等。

三位二进制编码器功能表

输入								输出		
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	1	1	1	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	1	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0



$$Y_2=I_7+I_6+I_5+I_4$$

$$Y_1=I_7+I_6+I_3+I_2$$

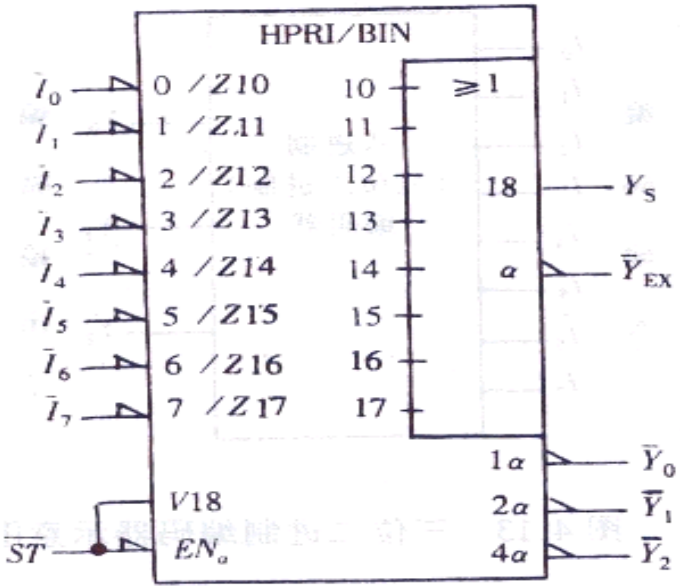
$$Y_0=I_7+I_5+I_3+I_1$$

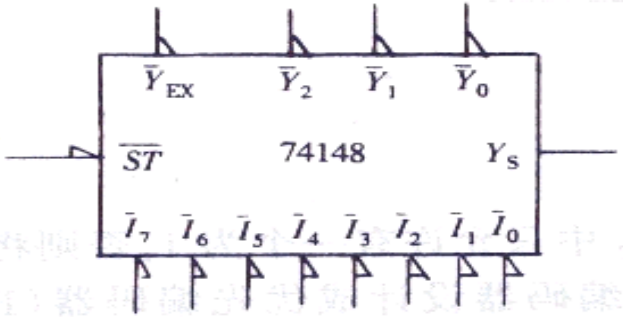
优先编码器

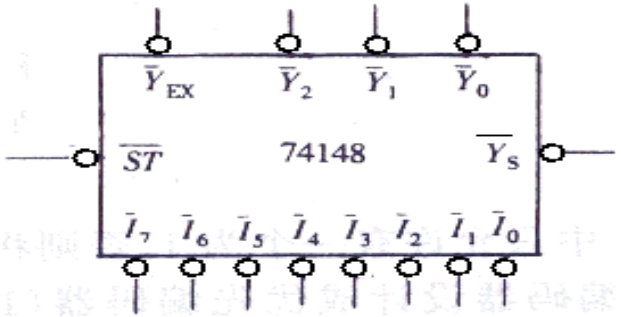
按输入开关量的标注大小规定其优先级别，允许多个输入同时有效，输出码与有效输入中级别最高的开关量对应。

1、集成8/3线优先编码器（74148）

逻辑符号







①集成8/3线优先编码器的端口和功能

8-线-3-线优先编码器74148的功能表

输入									输出				
$\overline{ST}$	$\overline{I}_0$	$\overline{I}_1$	$\overline{I}_2$	$\overline{I}_3$	$\overline{I}_4$	$\overline{I}_5$	$\overline{I}_6$	$\overline{I}_7$	$\overline{Y}_2$	$\overline{Y}_1$	$\overline{Y}_0$	$\overline{Y}_{EX}$	Y_S
H	×	×	×	×	×	×	×	×	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	×	×	×	×	×	×	×	L	L	L	L	L	H
L	×	×	×	×	×	×	L	H	L	L	H	L	H
L	×	×	×	×	×	L	H	H	L	H	L	L	H
L	×	×	×	×	L	H	H	H	L	H	H	L	H
L	×	×	×	L	H	H	H	H	H	L	L	L	H
L	×	×	L	H	H	H	H	H	H	L	H	L	H
L	×	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

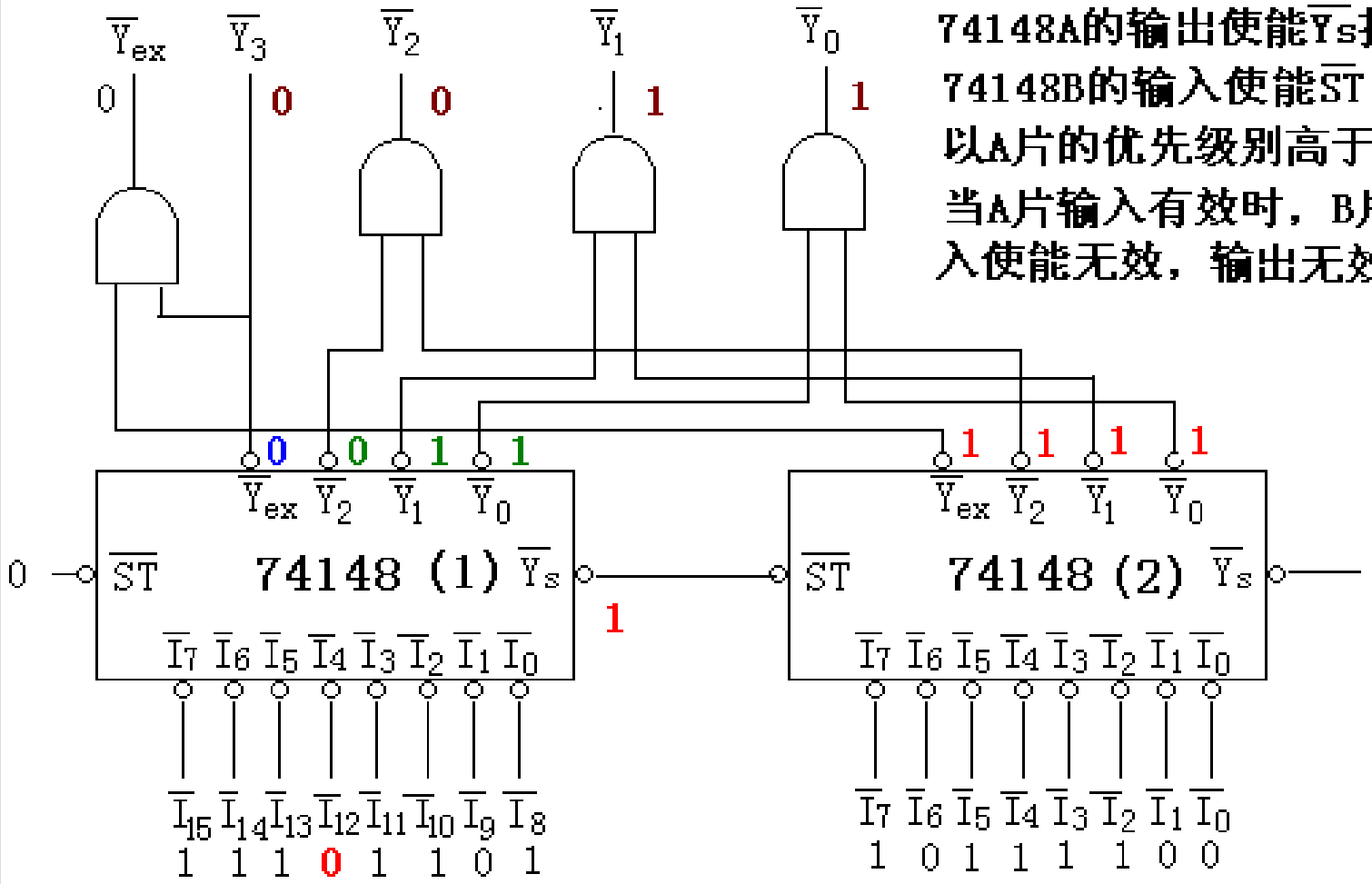
使能无效
无有效输入

使能有效，输入有效

所有输出无效
输出码无效，
输出使能有效

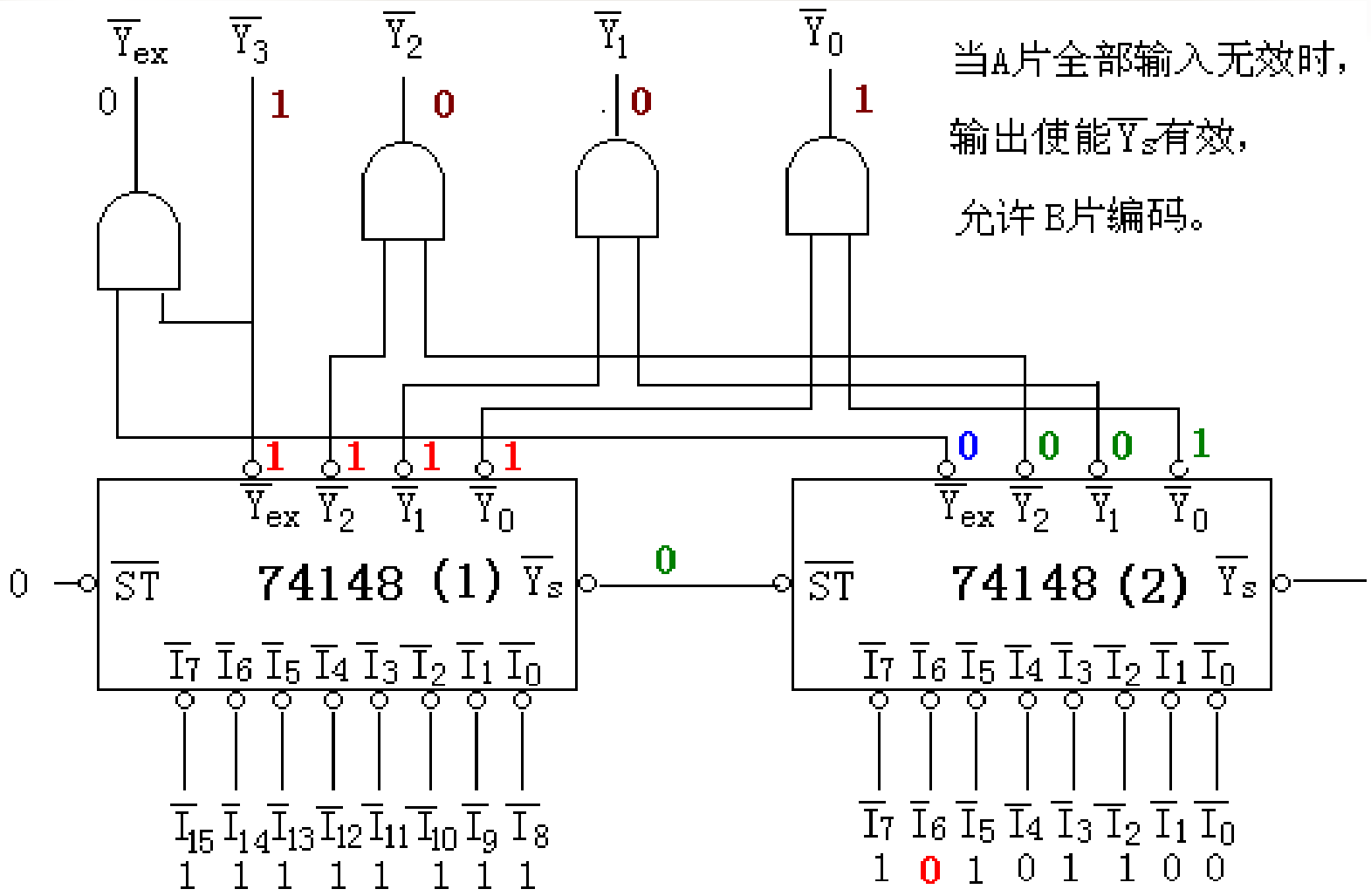
使能输出无效，输出码有效

②集成8/3线优先编码器的端口扩展



74148A的输出使能$\overline{Y}_S$控制74148B的输入使能$\overline{ST}$，所以A片的优先级别高于片B。当A片输入有效时，B片输入使能无效，输出无效码。

用两片74148扩展成16线/4线 编码器



用两片74148扩展成16线/4线 编码器

2、集成BCD码优先编码器74147

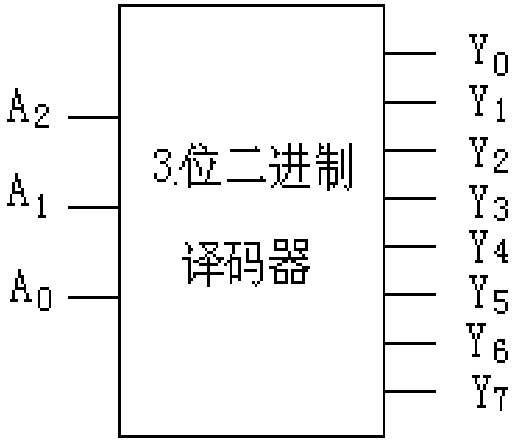
端口和功能：

表 3-3-4　　BCD 输出的 10 线-4 线优先编码器 74147 的功能表

输入									输出			
$\overline{I}_1$	$\overline{I}_2$	$\overline{I}_3$	$\overline{I}_4$	$\overline{I}_5$	$\overline{I}_6$	$\overline{I}_7$	$\overline{I}_8$	$\overline{I}_9$	$\overline{Y}_3$	$\overline{Y}_2$	$\overline{Y}_1$	$\overline{Y}_0$
H	H	H	H	H	H	H	H	H	H	H	H	H
×	×	×	×	×	×	×	×	L	L	H	H	L
×	×	×	×	×	×	×	L	H	L	H	H	H
×	×	×	×	×	×	L	H	H	H	L	L	L
×	×	×	×	×	L	H	H	H	H	L	L	H
×	×	×	×	L	H	H	H	H	H	L	H	L
×	×	×	L	H	H	H	H	H	H	L	H	H
×	×	L	H	H	H	H	H	H	H	H	L	L
×	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

3.3.2译码器

一、二进制译码器



1、结构：多输入、多输出

输入：使能控制开关量（选通）若干个，

n位二进制码A_0-A_{n-1}，

输出：N个开关量信号： Y_0-Y_{N-1} （$N=2^n$）。

2、功能：当使能控制有效时_（被选通），端口序号与输入的二进制码值相同的输出端为有效电平，指示了当前输入码，其他端口输出无效电平。

一组输入码只能使唯一的一个输出有效（电平与其他输出端不同）。

3、输出表达式：Y_i（A_{n-1}~A_0）=m_i （使能控制有效时），每个输出信号是输入变量最小项。

4、常用译码器型号：

74139（双2线-4线译码器）

2位码输入，4个开关量输出，一个低电平有效的使能$\overline{G}$

74138（3线-8线译码器）

3位码输入，8个开关量输出，三个使能控制：$EN=S_A\overline{S_B}\overline{S_C}$。

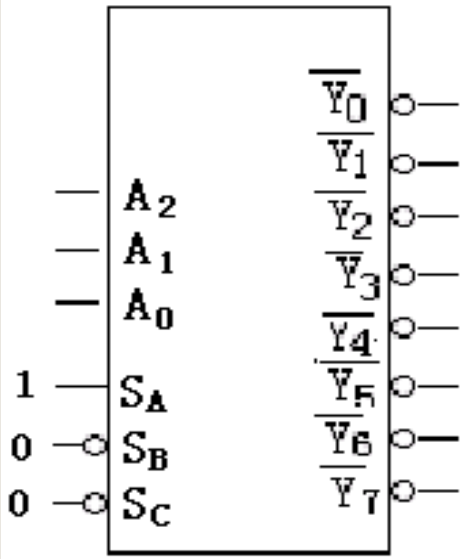
74154（4线-16线译码器）

4位码输入，16个开关量输出，两个使能$\overline{G_1}\overline{G_2}$同时为低电平有效。

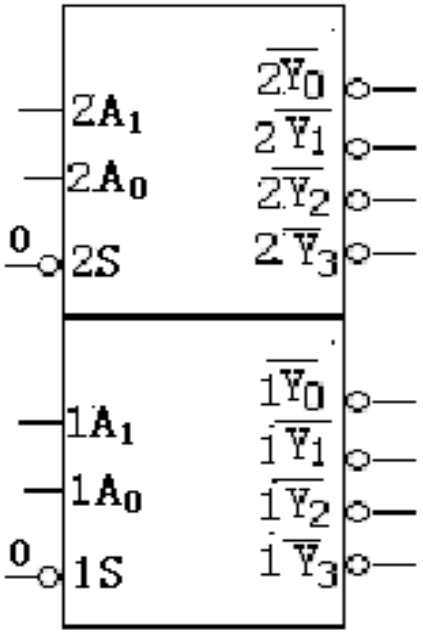
74145（4线-10线BCD码译码器）

一位BCD码输入，10个指示十进制数符的开关量输出

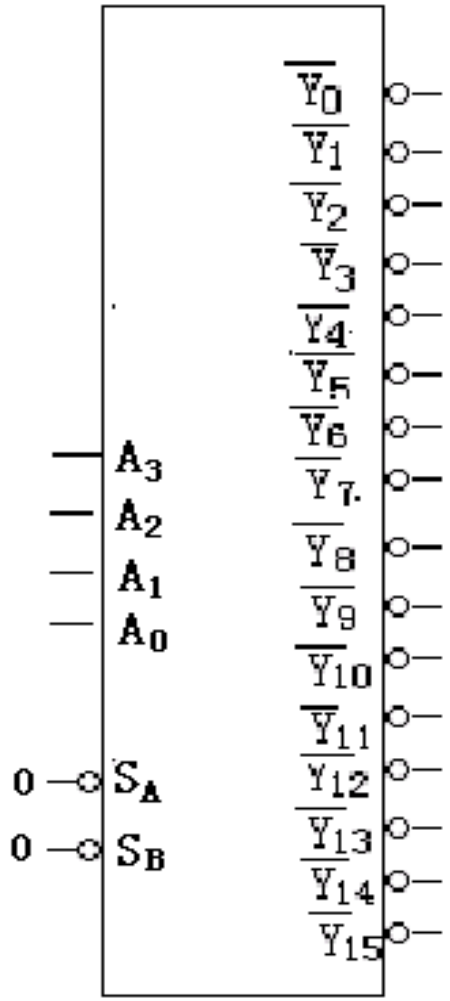
译码器逻辑符号



3线/8线译码器
74138



双2线/4线译码器
74139



4线/16线译码器
74154

5、74138功能表

各输出表达式

$$\overline{Y_i}(A_2A_1A_0) = S_A\overline{S}_B\overline{S}_C\,\overline{m_i}$$

当$S_A=1$，$S_B=S_C=0$时

$$\overline{Y}_0(A_2A_1A_0) = \overline{m}_0$$
$$\overline{Y}_1(A_2A_1A_0) = \overline{m}_1$$
$$\overline{Y}_2(A_2A_1A_0) = \overline{m}_2$$
$$\overline{Y}_3(A_2A_1A_0) = \overline{m}_3$$
$$\overline{Y}_4(A_2A_1A_0) = \overline{m}_4$$
$$\overline{Y}_5(A_2A_1A_0) = \overline{m}_5$$
$$\overline{Y}_6(A_2A_1A_0) = \overline{m}_6$$
$$\overline{Y}_7(A_2A_1A_0) = \overline{m}_7$$

表3-3-6　　3线-8线译码器74138功能表

输入					输出							
S_A	$\overline{S}_B+\overline{S}_C$	A_2	A_1	A_0	$\overline{Y}_0$	$\overline{Y}_1$	$\overline{Y}_2$	$\overline{Y}_3$	$\overline{Y}_4$	$\overline{Y}_5$	$\overline{Y}_6$	$\overline{Y}_7$
×	H	×	×	×	H	H	H	H	H	H	H	H
L	×	×	×	×	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

表3-3-6　3线-8线译码器74138功能表

输入					输出							
S_A	$\overline{S}_B+\overline{S}_C$	A_2	A_1	A_0	$\overline{Y}_0$	$\overline{Y}_1$	$\overline{Y}_2$	$\overline{Y}_3$	$\overline{Y}_4$	$\overline{Y}_5$	$\overline{Y}_6$	$\overline{Y}_7$
×	H	×	×	×	H	H	H	H	H	H	H	H
L	×	×	×	×	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

表1-3-6　三变量A、B、C的8个最小项及各自的取值表

序号	变量取值 A	B	C	m_0 ABC	m_1 ABC	m_2 ABC	m_3 ABC	m_4 ABC	m_5 ABC	m_6 ABC	m_7 ABC
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0
5	1	0	1	0	0	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	0	1

$\overline{Y}_i(A_2A_1A_0) = S_A\overline{S}_B\overline{S}_C\overline{m}_i$，当$S_A=1$，$S_B=S_C=0$时

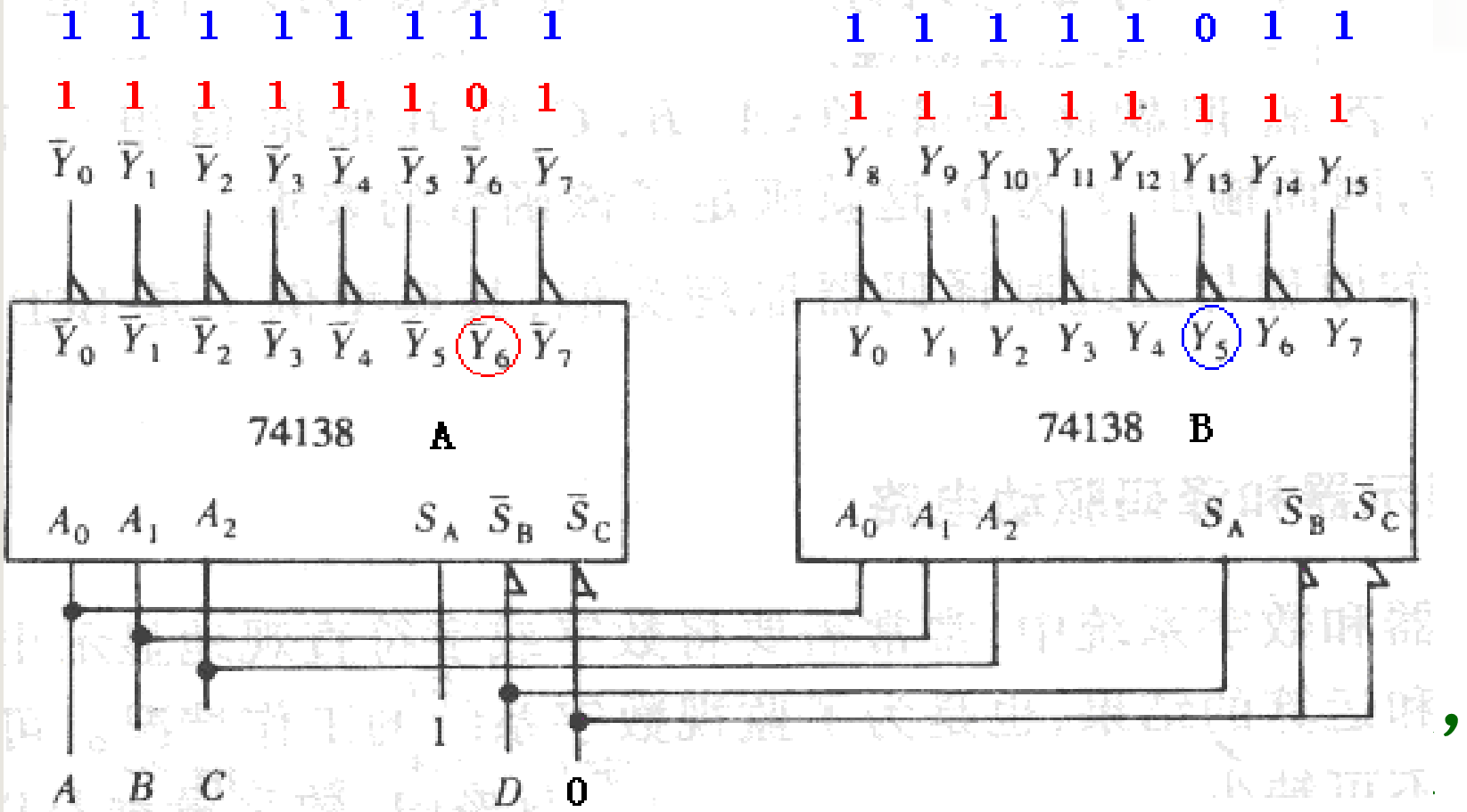
各输出表达式

$$\overline{Y}_0(A_2A_1A_0)=\overline{m}_0 \qquad \overline{Y}_4(A_2A_1A_0)=\overline{m}_4$$

$$\overline{Y}_1(A_2A_1A_0)=\overline{m}_1 \qquad \overline{Y}_5(A_2A_1A_0)=\overline{m}_5$$

$$\overline{Y}_2(A_2A_1A_0)=\overline{m}_2 \qquad \overline{Y}_6(A_2A_1A_0)=\overline{m}_6$$

$$\overline{Y}_3(A_2A_1A_0)=\overline{m}_3 \qquad \overline{Y}_7(A_2A_1A_0)=\overline{m}_7$$



D=0时，A片译码，B片禁止

D=1时，A片禁止，B片译码

利用使能控制端将两片74138扩展成4线-16线二进制译码器。

高位码D同时控制A片的$\overline{S}_B$（0有效）和B片的S_A（1有效）

（2）实现用标准与或表达式表示的组合逻辑函数，函数变量数与多一译码器的输入码位数相同。

方法：

①写函数各输出的标准与或表达式，并应用摩根定理转换成最小项的“与非”形式。

②译码器的使能接有效电平。

③函数变量按最小项编号的位序从地址码端输入。

④采用与非门将译码器输出序号与函数表达式中最小项序号相同的端口综合构成函数的输出端。N输出的函数需要N个与非门。

例：用74138和与非门设计双输出函数F_1、F_2

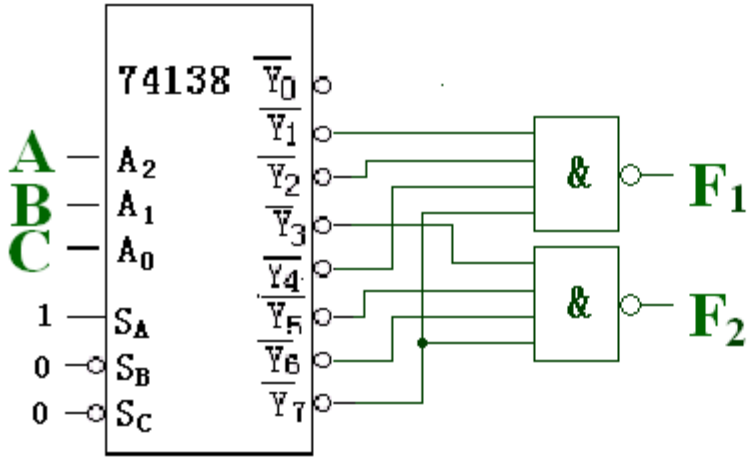
$F_1=\bar{A}\bar{B}C+\bar{A}B\bar{C}+A\bar{B}\bar{C}+ABC$

$F_2=\bar{A}BC+A\bar{B}C+AB\bar{C}+ABC$

解：

$F_1(A、B、C)=m_1+m_2+m_4+m_7=\overline{\overline{m_1}\ \overline{m_2}\ \overline{m_4}\ \overline{m_7}}$

$F_2(A、B、C)=m_3+m_5+m_6+m_7=\overline{\overline{m_3}\ \overline{m_5}\ \overline{m_6}\ \overline{m_7}}$



二、代码转换器

输入、输出都是二进制码，但编码形式不同。

BCD码/七段显示译码/驱动器

输入：一位BCD码（$\underline{A_3}$、$\underline{A_2}$、$\underline{A_1}$、A_0）；

三个控制信号LT、$\overline{BI}$、$\overline{RBI}$

均为低电平有效，控制优先级为：

BI：灭灯； LT：试灯； RBI：灭零。

输出：七个开关量段信号（Y_a、Y_b、Y_c、Y_d、Y_e、Y_f）

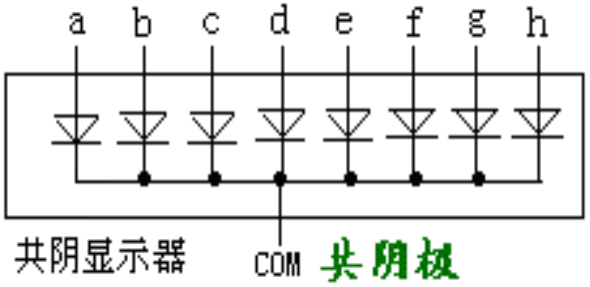
控制七段显示器的七个发光二极管显示与输入BCD码 对应的十进制数符。

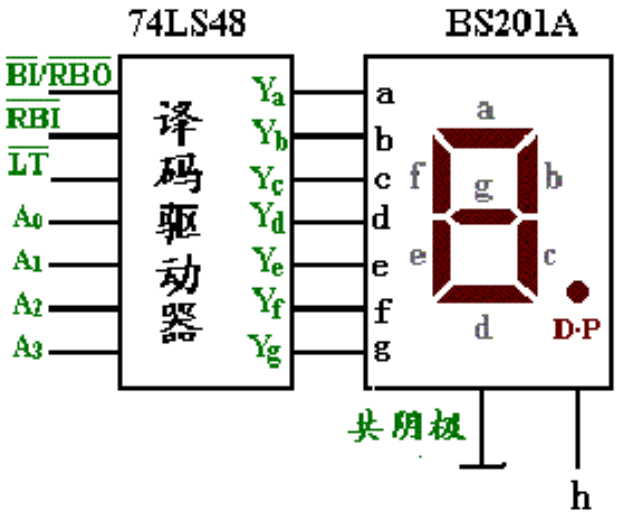
7447：驱动共阳显示器(LG5011BSR)，输出低电平有效

共阳——公共端高电平驱动、段信号低电平驱动

7448：驱动共阴显示器(BS201A)，输出高电平有效

共阴——公共端低电平驱动、段信号高电平驱动





7 段数字显示器 原理图

74LS48逻辑功能表

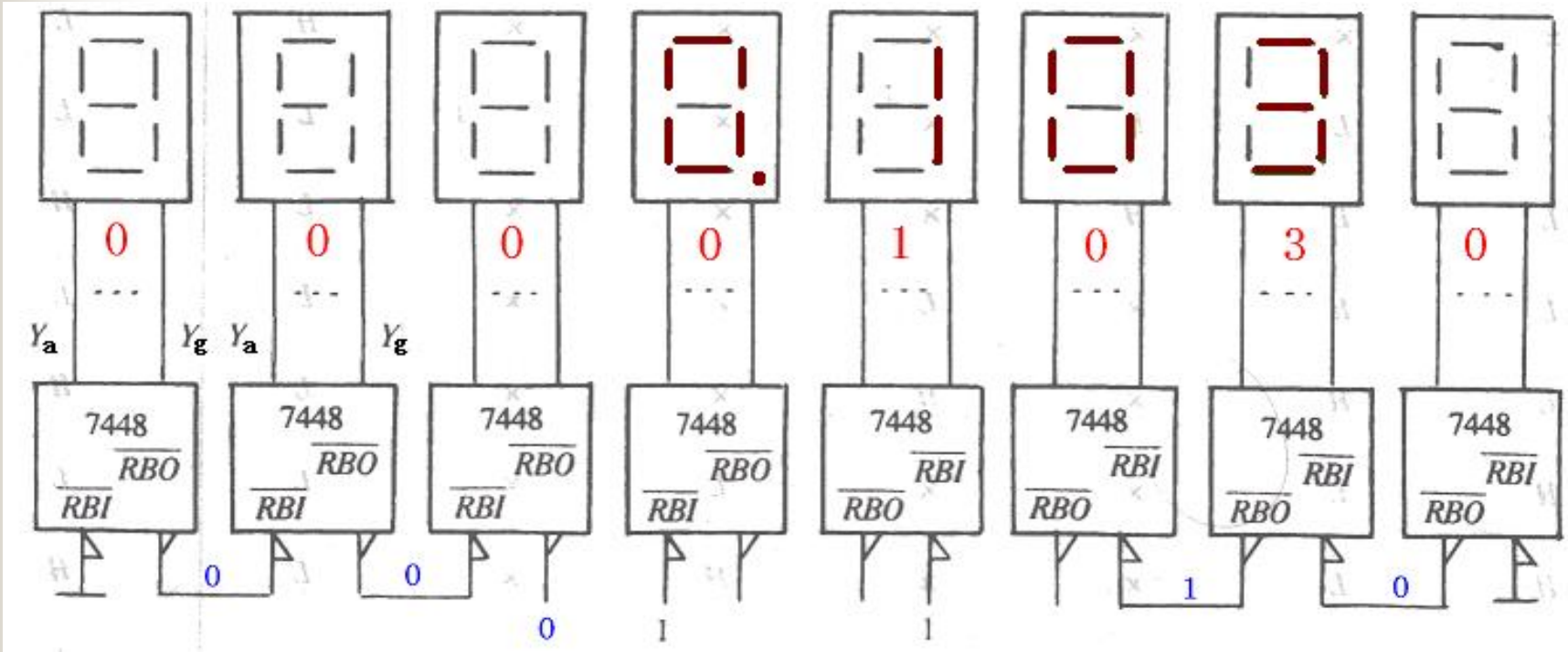
输入						in/out	输出							
$\overline{LT}$	$\overline{RBI}$	A_3	A_2	A_1	A_0	$\overline{BI/RBO}$	Y_a	Y_b	Y_c	Y_d	Y_e	Y_f	Y_g	
1	1	0	0	0	0	1	1	1	1	1	1	1	0	显零
1	×	0	0	0	1	1	0	1	1	0	0	0	0	1
1	×	0	0	1	0	1	1	1	0	1	1	0	1	2
1	×	0	0	1	1	1	1	1	1	1	0	0	1	3
1	×	0	1	0	0	1	0	1	1	0	0	1	1	4
1	×	0	1	0	1	1	1	0	1	1	0	1	1	5
1	×	0	1	1	0	1	0	0	1	1	1	1	1	6
1	×	0	1	1	1	1	1	1	1	0	0	0	0	7
1	×	1	0	0	0	1	1	1	1	1	1	1	1	8
1	×	1	0	0	1	1	1	1	1	0	0	1	1	9
1	×	1	0	1	0	1	0	0	0	1	1	0	1	c
1	×	1	0	1	1	1	0	0	1	1	0	0	1	3
1	×	1	1	0	0	1	0	1	0	0	0	1	1	
1	×	1	1	0	1	1	1	0	0	1	0	1	1	F
1	×	1	1	1	0	1	0	0	0	1	1	1	1	L
1	×	1	1	1	1	1	0	0	0	0	0	0	0	
×	×	×	×	×	×	0	0	0	0	0	0	0	0	灭显
1	0	0	0	0	0	0	0	0	0	0	0	0	0	灭零
0	×	×	×	×	×	1	1	1	1	1	1	1	1	试灯

灭零控制功能的8位数码显示（4位整数、4位小数）

当RBI=0且输入BCD码为0000时, 灭零。

整数部分的高位和小数部分的低位0灭显。

整数部分最低位和小数部分最高位的0必须显示，RBI=1。



3.3.3加法器

功能：采用逻辑运算关系实现二进制运算。

一、半加器

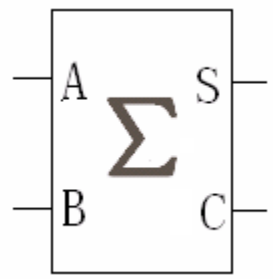
功能：实现两个一位二进制数的加运算。

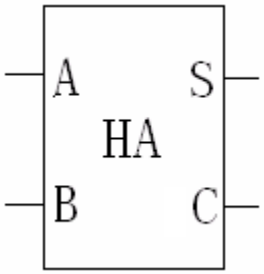
输入：两个二进制加数A、B

输出：A加B的和S和进位输出Co

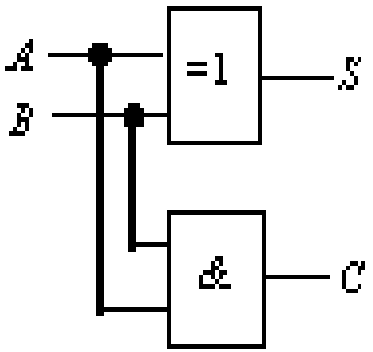
输出函数式：$S=A\oplus B$ ； $C=AB$

逻辑符号：





逻辑图



半加器真值表

输入 A	输入 B	输出 S	输出 C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

二、全加器

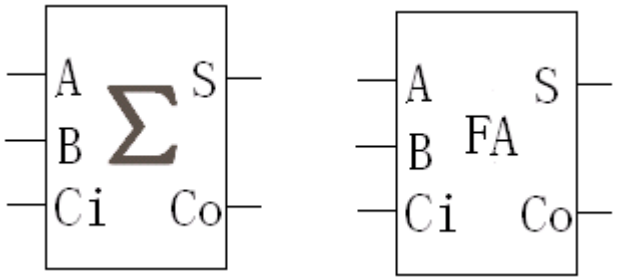
功能：实现三个一位二进制数的加运算。

输入：两个二进制加数输入A、B及低位的进位输入Ci

输出：A加B加Ci的和S和进位输出Co

$S=A \oplus B \oplus Ci$ ； $Co=AB+BCi+ACi$

逻辑符号



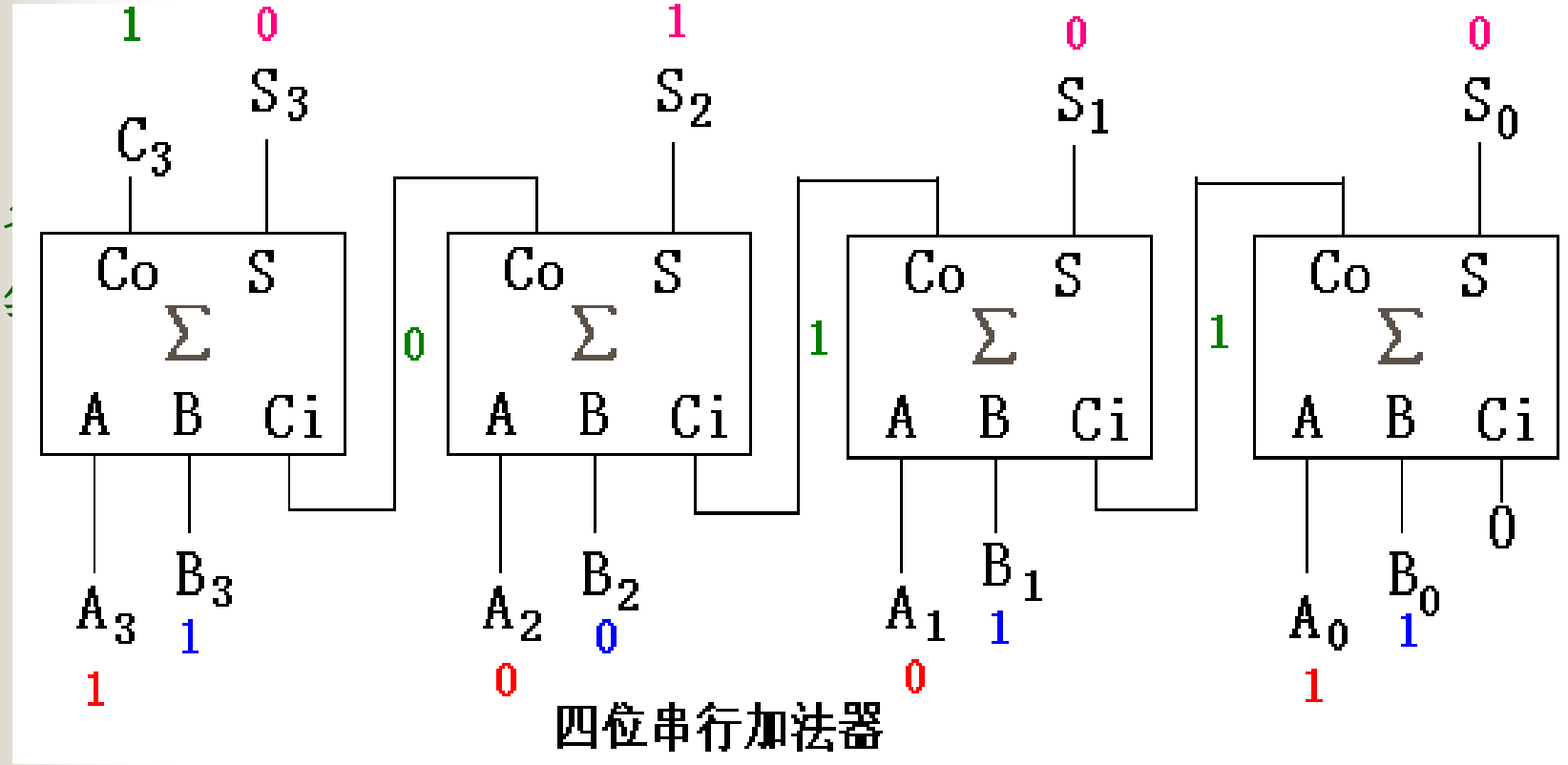
全加器真值表

输入			输出	
A	B	C_i	S	C_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

二、多位加法器

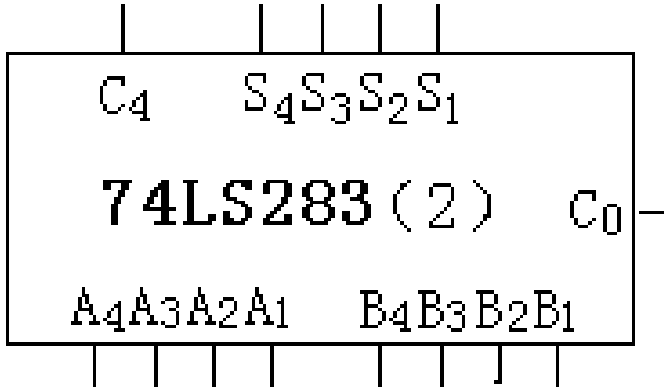
两个n位二进制数A（A_{n-1}~A_0）、B（B_{n-1}~B_0）的加法运算，输出加运算的和S（S_{n-1}~S_0） 及最高位的进位C_{n-1}。

1、串行进位加法器



2、集成4位超前并行进位加法器

A、B为两个4位的
二进制加数
S为A加B加C_0的
4位二进制和，
C_4为A加B加C_0的最高位进位



$S_1=A_1 \oplus B_1 \oplus C_0$ ； 各位进位：

$S_2=A_2 \oplus B_2 \oplus C_1$ ；

$S_3=A_3 \oplus B_3 \oplus C_2$ ；

$S_4=A_4 \oplus B_4 \oplus C_3$ ；

$C_1=A_1B_1+(A_1 \oplus B_1)C_0=G_1+P_1C_0$

$C_2=A_2B_2+(A_2 \oplus B_2)C_1=G_2+P_2C_1$
$=G_2+P_2G_1+P_2P_1C_0$

$C_3=A_3B_3+(A_3 \oplus B_3)C_2=G_3+P_3C_2$
$=G_3+P_3G_2+P_3P_2P_1G_1+P_3P_2P_1C_0$

进位产生项：$Gi=AiBi$ ，进位传递项：$Pi=Ai \oplus Bi$

例：用4位加法器设计一个代码转换电路。当控制信号X=0时，输入8421BCD码，输出余3码，当X=1时，输入余3码，输出8421BCD码。

解：

输入为8421BCD码时加3（0011）输出余3码；

输入为余3码时减3输出8421BCD码，

减3可以用加-3的补码1101实现。

用信号X对控制代码转换：

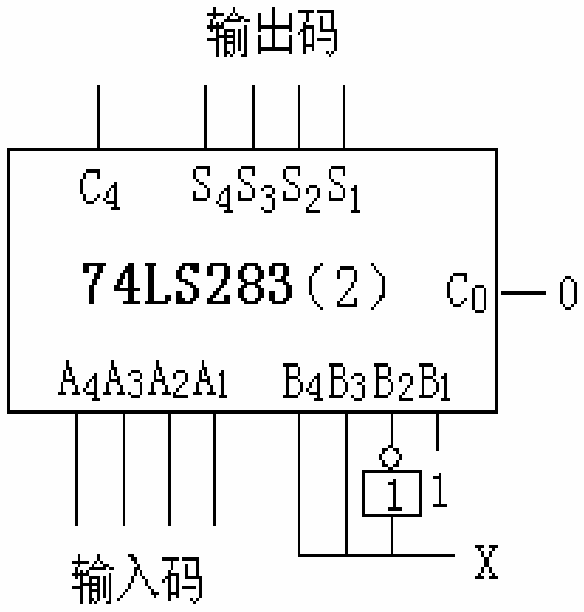
当X=0时，输入8421BCD码加0011后输出余3码；

当X=1时，输入余3码加1101后输出8421BCD码。

显然，加法器A输入待转换码，

B输入 $B_4=B_3=X$，$B_2=\overline{X}$，$B_1=1$，$C_0=0$；

或输入 $B_4=B_3=X$，$B_2=\overline{X}$，$B_1=0$，$C_0=1$。



3.3.4数据选择器和数据分配器

一、数据选择器

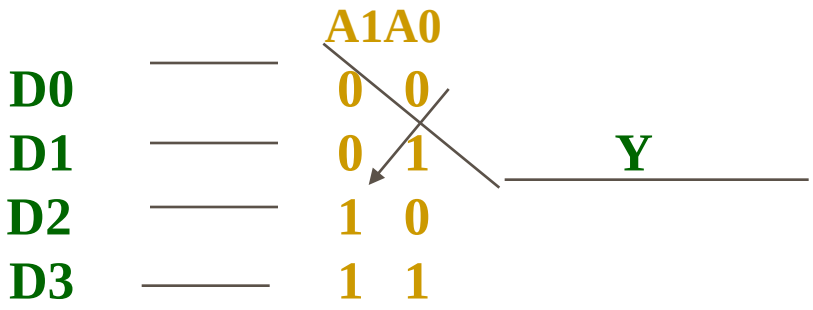
结构：多输入、单输出

输入端：使能控制（选通） 1个：ST

选择控制 n位：A_{n-1}~A_0

数据输入 2^n个 ： D_{m-1}~D_0 ， m= 2^n

功能：当使能有效时（被选通），根据选择信号从多路数据中选择一路输出。



功能表

选 择		输 出
A_1	A_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

双四选一MUX 74153

两位控制码A_0、A_1选择四个数据输入D_0~D_3中的一个到输出Y。

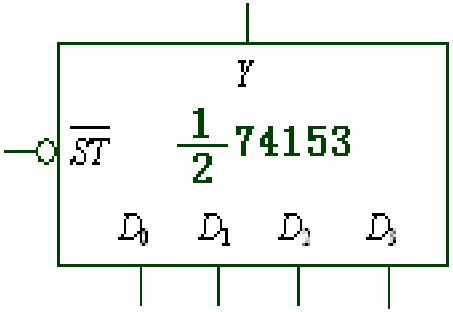
四选一数据选择器功能表

使能	选择输入		数据输入				输出
$\overline{ST}$	A_1	A_0	D_0	D_1	D_2	D_3	Y
H	×	×	×	×	×	×	L
L	L	L	L	×	×	×	L
L	L	L	H	×	×	×	H
L	L	H	×	L	×	×	L
L	L	H	×	H	×	×	H
L	H	L	×	×	L	×	L
L	H	L	×	×	H	×	H
L	H	H	×	×	×	L	L
L	H	H	×	×	×	H	H

输出函数表达式：

$$Y=\overline{ST}(\bar{A}_1\bar{A}_0D_0+\bar{A}_1A_0D_1+A_1\bar{A}_0D_2+A_1A_0D_3)$$

当$\overline{ST}$=0时，$Y(A_1,A_0)=\sum m_iD_i$



八选一MUX 74151

三位控制码A_2、A_1、A_0选择八个数据输入D_0~D_7中的一个到输出Y，或反相输出到$\overline{Y}$。

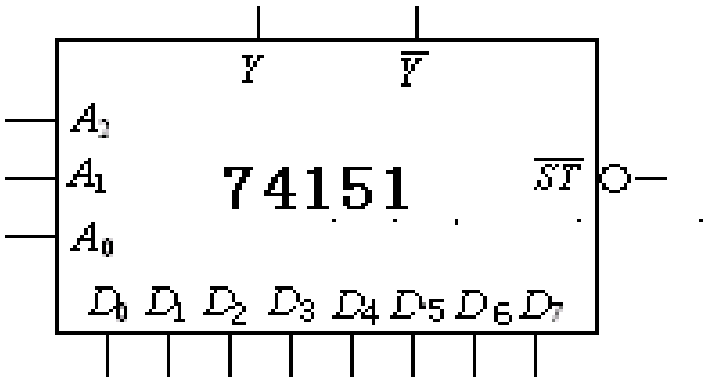
八选一数据选择器
74151功能表

使能	选 择			输 出	
$\overline{ST}$	A_2	A_1	A_0	Y	$\overline{Y}$
H	×	×	×	L	H
L	L	L	L	D_0	$\overline{D}_0$
L	L	L	H	D_1	$\overline{D}_1$
L	L	H	L	D_2	$\overline{D}_2$
L	L	H	H	D_3	$\overline{D}_3$
L	H	L	L	D_4	$\overline{D}_4$
L	H	L	H	D_5	$\overline{D}_5$
L	H	H	L	D_6	$\overline{D}_6$
L	H	H	H	D_7	$\overline{D}_7$

输出函数表达式：

$$Y(A2,A1,A0)=\overline{ST}\sum m_iD_i$$

逻辑符号：



数据选择器应用

1、信号选择控制

2、实现单输出组合逻辑函数（函数发生器）。

当使能有效($\overline{ST}$="0")，函数变量从选择控制端输入，输出可写成函数变量最小项和对应数据输入相与的或项。

方法：

（1）写函数的标准与或表达式。

（2）数据选择器的使能接有效电平。

（3）根据数据选择器的控制输入端数选择函数的变量数，并按最小项编号的位序从控制端输入

（4）比较函数的标准与或表达式和数据选择器的输出表达式，确定各Di的值。

当函数变量数与选择器控制码位数相同时，选择器各数据输入Di等于对应最小项变量组合的函数值。

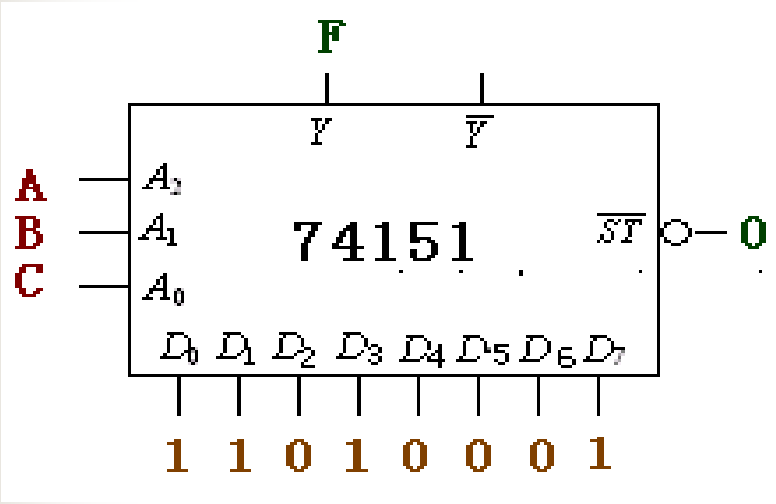
例： $F=\overline{A}\overline{B}\overline{C}+\overline{A}\overline{B}C+\overline{A}B\overline{C}+ABC$

$F（A、B、C）$

$= m_0 + m_1 + m_3 + m_7$

$=\sum m（0,1,3,7）$

变量取值	函数值	选择数据
A B C	F	Di
0 0 0	1	D_0
0 0 1	1	D_1
0 1 0	0	D_2
0 1 1	1	D_3
1 0 0	0	D_4
1 0 1	0	D_5
1 1 0	0	D_6
1 1 1	1	D_7



当函数变量数比选择器控制码位数多时，选择器各数据输入Di等于多余变量的组合。

例： $F=\overline{A}\overline{B}\overline{C}+\overline{A}\overline{B}C+\overline{A}B\overline{C}+ABC$

3变量函数用四选一数据选择器74153实现。

解：选择变量B、C从控制端输入，Di是A的函数。

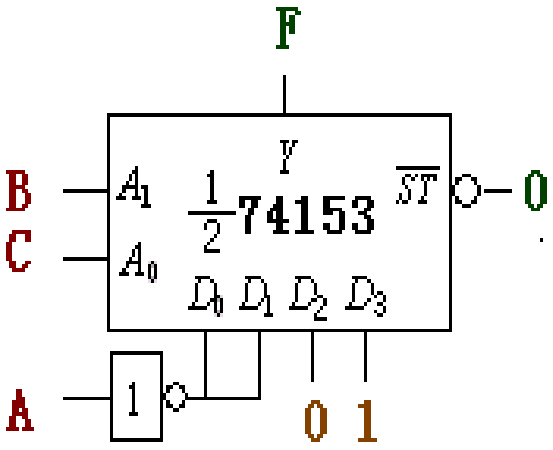
$F(B,C)=\overline{A}m_0+\overline{A}m_1+(\overline{A}+A)m_3$

$=\overline{A}m_0+\overline{A}m_1+m_3$

所以：$D_0=D_1=\overline{A}$，

$D_2=0$，$D_3=1$。

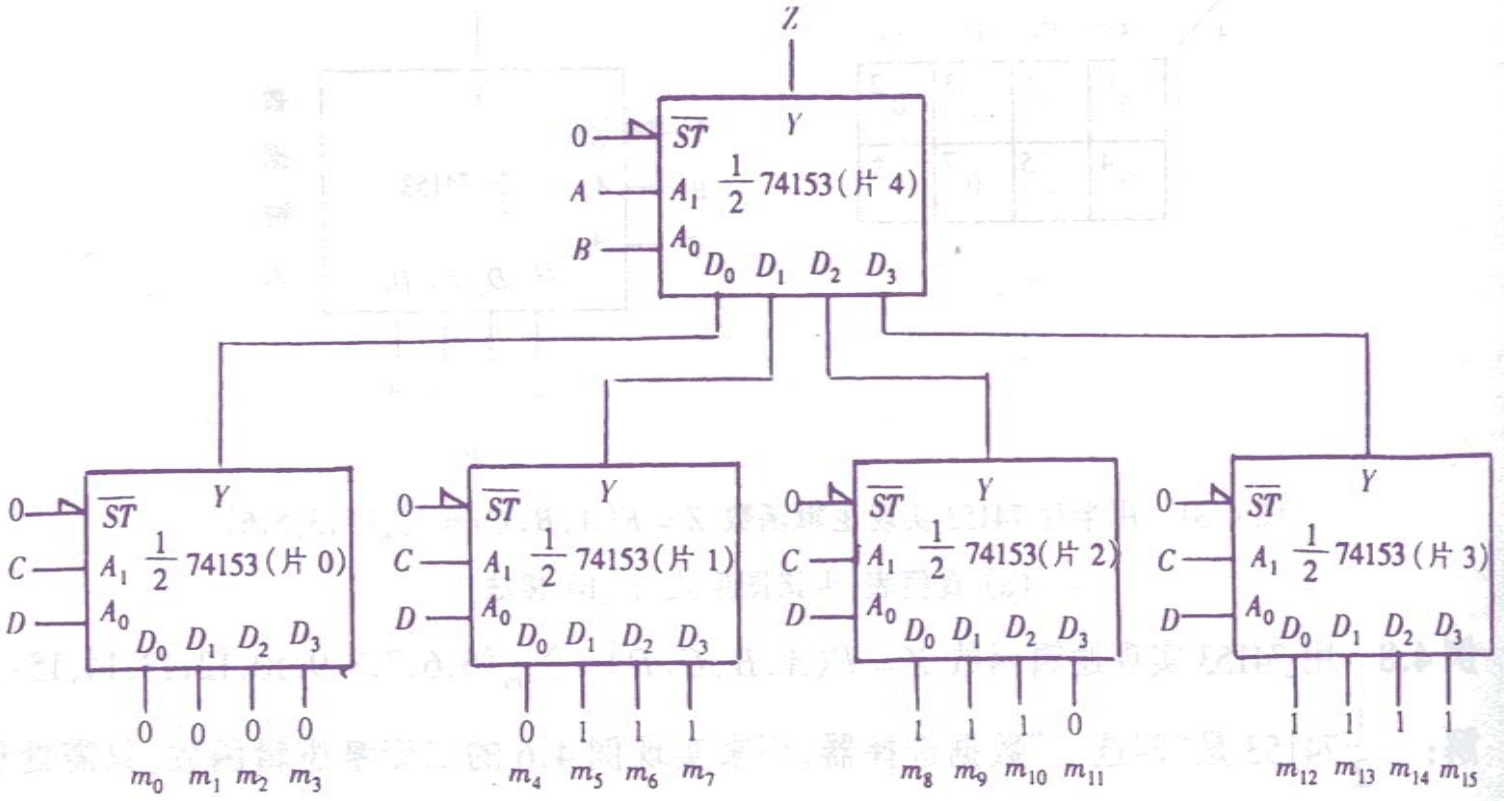
F　A \ BC	D_0 00	D_1 01	D_3 11	D_2 10
0	1	1	1	0
1	0	0	1	0



当函数变量数比选择器控制码位数多时，也可扩展控制码位数。

例:用四选一数据选择器扩展成十六选一选择器实现四变量逻辑函数:Z(A,B,C,D)=∑m(5,6,7,8,9,10,12,13,14,15)

两级选择四选一构成十六选一



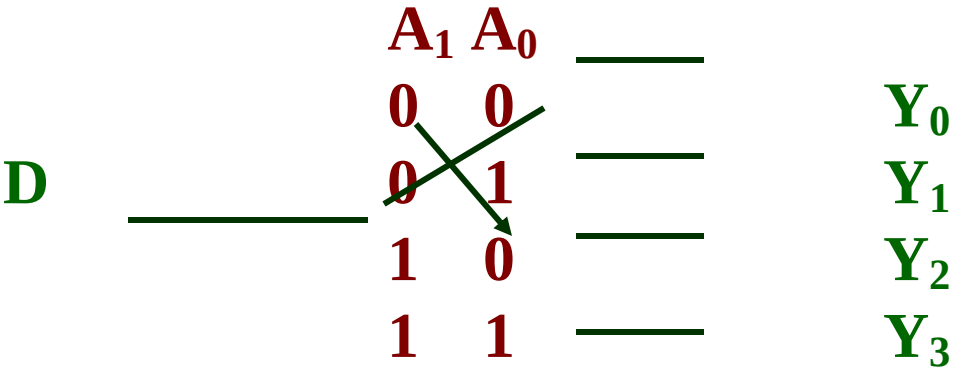
二、数据分配器

结构：单数据输入、多数据输出

输入端：数据输入 1个D，选择控制n个（A_{n-1}~A_0），

输出端： 2^n个，Y_0~Y_{m-1}， $m=2^n$ 。

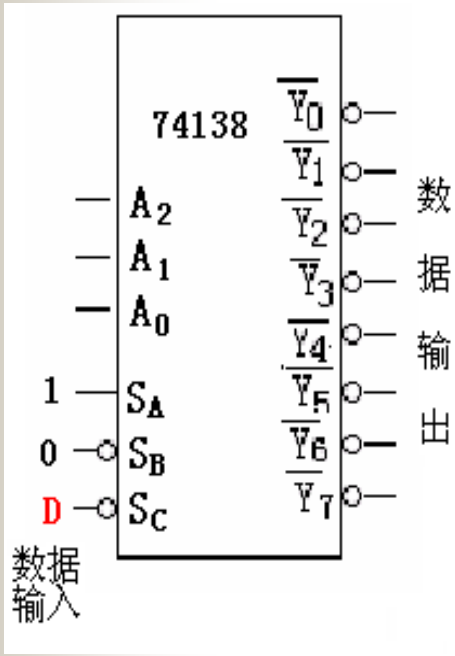
功能：当使能有效时，根据选择控制信号将输入数据D分配给多路输出Y_0~Y_{m-1}中的一路。



二进制译码器用作数据分配器：

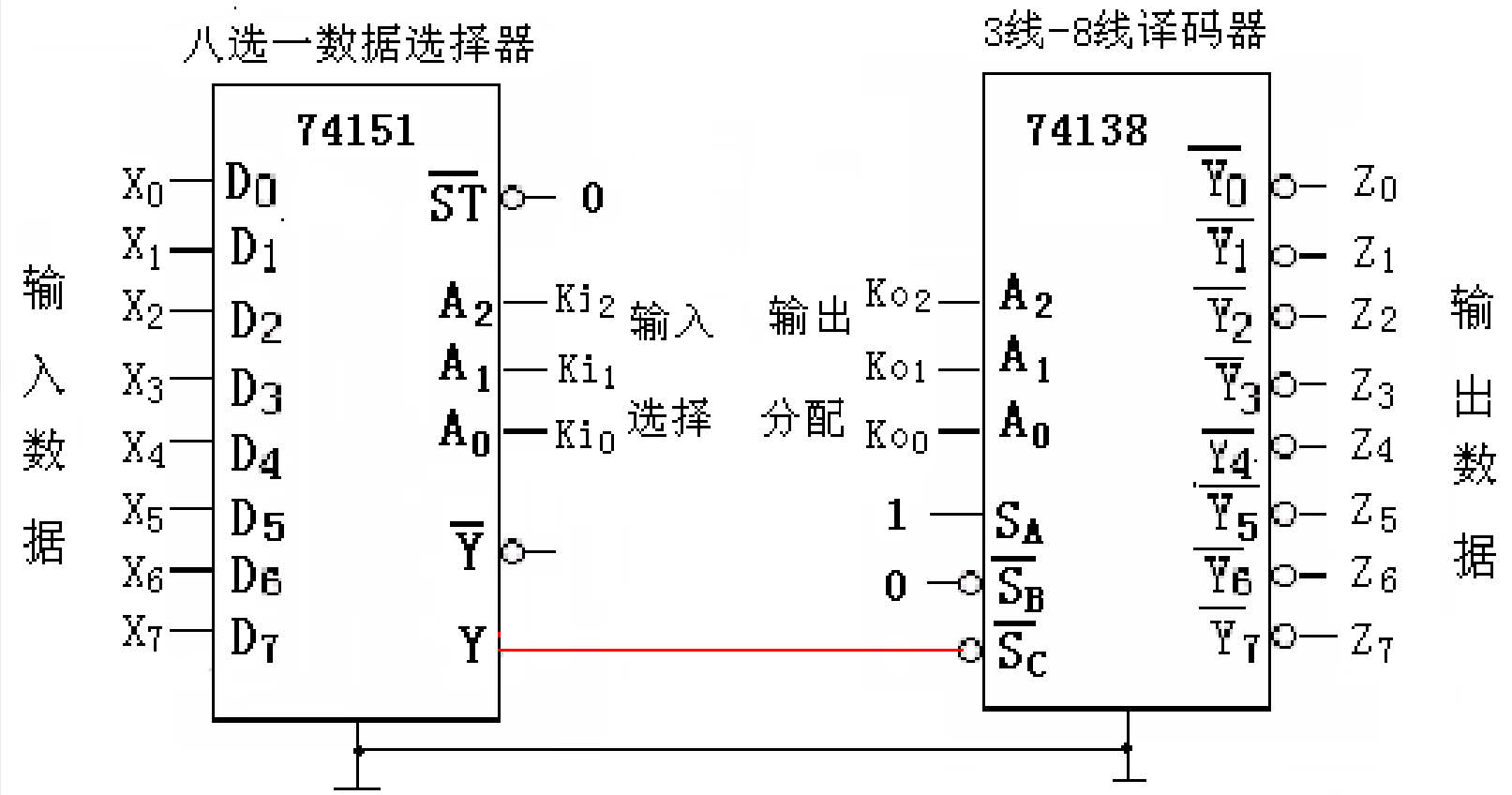
数据从使能端S输入，二进制码输入端作为分配选择控制，译码输出端为数据输出通道。

用3线-8线译码器构成8通道数据分配器



输入						输出							
S_A $\overline{S_B}$		$\overline{S_C}$	A_2	A_1	A_0	$\overline{Y_0}$	$\overline{Y_1}$	$\overline{Y_2}$	$\overline{Y_3}$	$\overline{Y_4}$	$\overline{Y_5}$	$\overline{Y_6}$	$\overline{Y_7}$
H	L	L	L	L	L	L	H	H	H	H	H	H	H
H	L	H	L	L	L	H	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
H	L	H	L	L	H	H	H	H	H	H	H	H	H
H	L	L	L	H	L	H	H	L	H	H	H	H	H
H	L	H	L	H	L	H	H	H	H	H	H	H	H
H	L	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	H	H	H	H	H	H	H	H	H	H
H	L	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	H	L	L	H	H	H	H	H	H	H	H
H	L	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	H	H	H
H	L	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	L	H	H	H	H	H	H	H	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L
H	L	H	H	H	H	H	H	H	H	H	H	H	H

多路信号采用一条信号线分时传送



例：$Ki_2=0, Ki_1=1, Ki_0=0$，
$Ko_2=1, Ko_1=0, Ko_0=1$

将X_2的数据传送到Z_5，其余$Zi=1$。

3.3.5数据比较器

功能：采用逻辑运算关系比较两个二进制数A、B的大小，输出表示比较结果的（A>B）、（A<B）、（A=B）三个开关量。

一、1位数字比较器的逻辑运算关系：

$(A>B)=A\overline{B}$；$(A<B)=\overline{A}B$；$(A=B)=A\odot B$

二、多位数字比较的方法：

当高位不同时，得比较结果；当高位相同时，顺位序比较低位；当两个数的所有位一一对应相同时，A=B

三、集成4位数字比较器7485

比较对象输入：a3~a0、b3~b0；

级联扩展输入：（a>b)、(a<b)、(a=b)

当a3~a0与b3~b0各位一一对应相同时，三个输出分别为：

（A>B）= （a>b) 、（A<B）= (a<b) 、（A=B）= (a=b)

高电平有效

三、4位集成数字比较器 74LS85 功能表

比较对象输入				级联输入			比较结果输出		
a_3b_3	a_2b_2	a_1b_1	a_0b_0	a>b	a<b	a=b	A>B	A<B	A=B
$a_3>b_3$	×	×	×	×	×	×	1	0	0
$a_3<b_3$	×	×	×	×	×	×	0	1	0
$a_3=b_3$	$a_2>b_2$	×	×	×	×	×	1	0	0
$a_3=b_3$	$a_2<b_2$	×	×	×	×	×	0	1	0
$a_3=b_3$	$a_2=b_2$	$a_1>b_1$	×	×	×	×	1	0	0
$a_3=b_3$	$a_2=b_2$	$a_1<b_1$	×	×	×	×	0	1	0
$a_3=b_3$	$a_2=b_2$	$a_1=b_1$	$a_0>b_0$	×	×	×	1	0	0
$a_3=b_3$	$a_2=b_2$	$a_1=b_1$	$a_0<b_0$	×	×	×	0	1	0
$a_3=b_3$	$a_2=b_2$	$a_1=b_1$	$a_0=b_0$	1	0	0	1	0	0
$a_3=b_3$	$a_2=b_2$	$a_1=b_1$	$a_0=b_0$	0	1	0	0	1	0
$a_3=b_3$	$a_2=b_2$	$a_1=b_1$	$a_0=b_0$	0	0	1	0	0	1

$$(A>B)=a_3\overline{b}_3+(a_3\odot b_3)\,a_2\overline{b}_2+(a_3\odot b_3)\,(a_2\odot b_2)\,a_1\overline{b}_1+(a_3\odot b_3)\,(a_2\odot b_2)\,(a_1\odot b_1)\,a_0\overline{b}_0+(a_3\odot b_3)\,(a_2\odot b_2)\,(a_1\odot b_1)\,(a_0\odot b_0)\,(a>b)$$

$$(A>B)=\overline{a}_3b_3+(a_3\odot b_3)\,\overline{a}_2b_2+(a_3\odot b_3)\,(a_2\odot b_2)\,\overline{a}_1b_1+(a_3\odot b_3)\,(a_2\odot b_2)\,(a_1\odot b_1)\,\overline{a}_0b_0+(a_3\odot b_3)\,(a_2\odot b_2)\,(a_1\odot b_1)\,(a_0\odot b_0)\,(a>b)$$

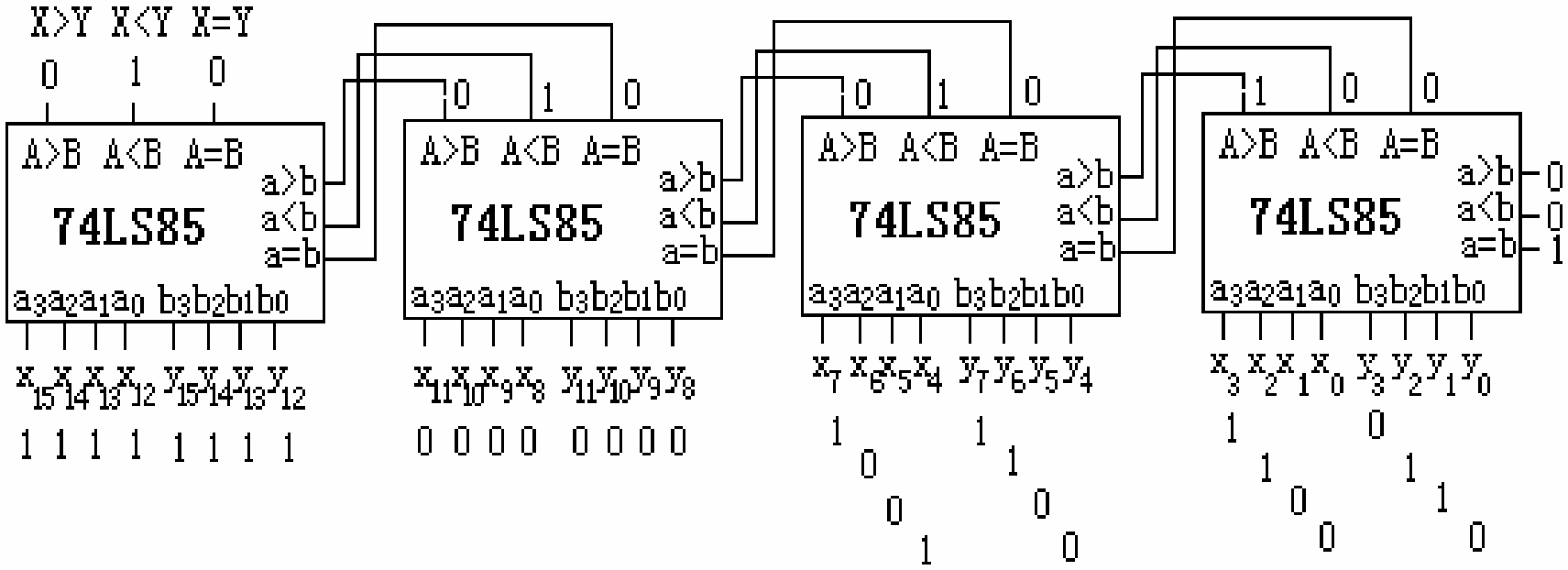
$$(A=B)=(a_3\odot b_3)\,(a_2\odot b_2)\,(a_1\odot b_1)\,(a_0\odot b_0)\,(a=b)$$

四、集成数字比较器的位数扩展

当需要比较的二进制数位大于 4 位时，采用多片集成比较器级联扩展位数。比较对象按位序分成四位一组输入各片集成比较器，构成**串行级联**。

比较结果从最高位序组的比较器输出，其它各片比较器的输出分别连接相邻高位组比较器的级联扩展输入。当高位组的输入相等时，传递低位组的比较结果，最低位组级联输入为(A>B)=0，(A<B)=0，(A=B)=1。速度较低，比较时间由数据位数决定。

例：四片74LS85串行级联成16位数据比较器



3.4 组合逻辑中的竞争冒险

电路输出信号与输入信号不符合应有的逻辑关系。

例：$A\overline{A}\neq 0$，$A+\overline{A}\neq 1$，

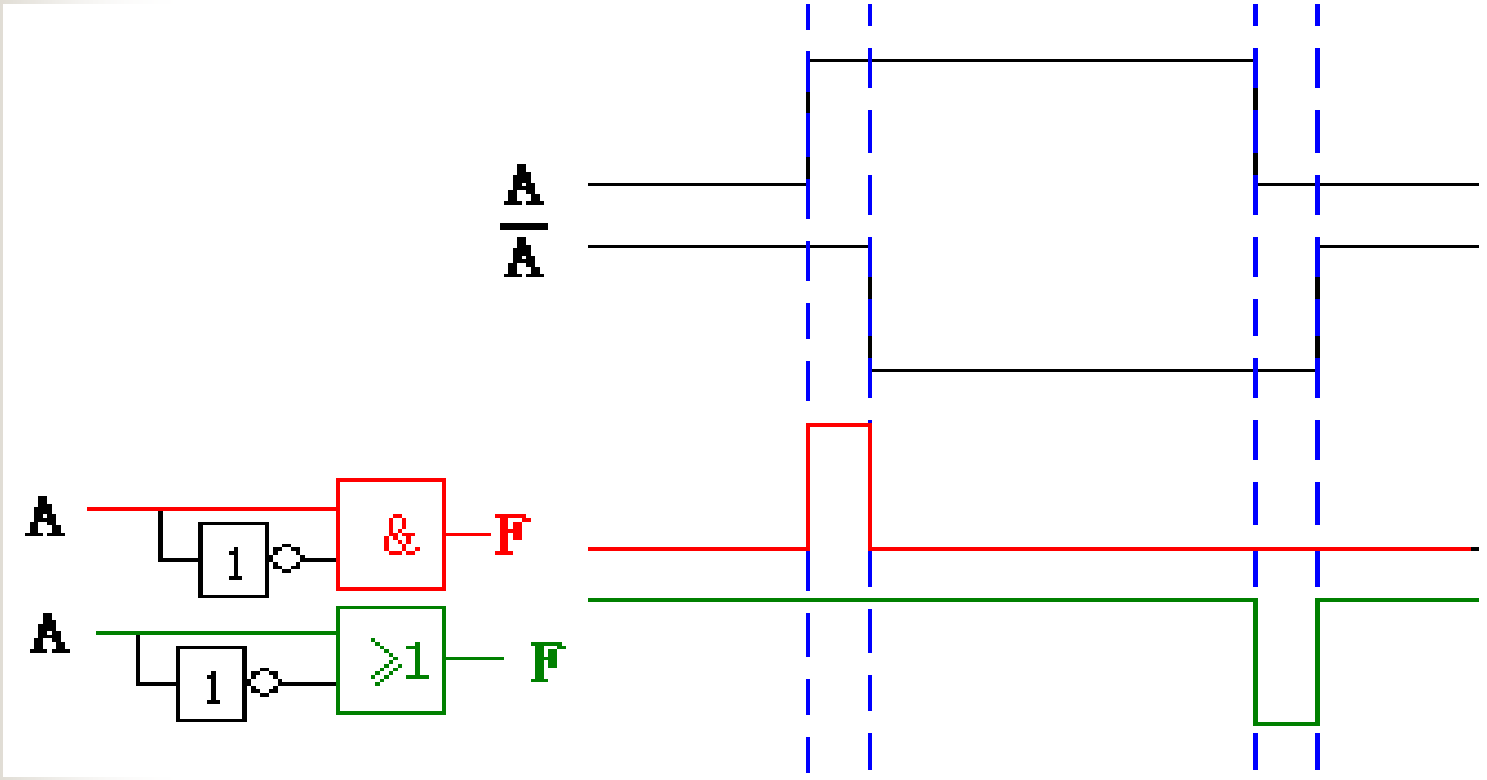
3.4.1竞争冒险的概念及其产生的原因

竞争——由于信号通过逻辑门有传输时间延迟，同时输入电路的信号通过不同途径到达同一个门的时间有先后。

冒险——由于竞争原因造成逻辑门错误输出干扰脉冲的现象。

$F=A\overline{A}$，在A信号的上升沿(0->1)产生正脉冲冒险

$F=A+\overline{A}$，在A信号的下降沿(1->0)产生负脉冲冒险



3.4.2消除竞争冒险的方法

1、增加封锁脉冲

输入信号改变时，封锁信号有效，逻辑门输出不能改变；

输入信号稳定后，封锁信号无效，允许逻辑门输出改变。

封锁与门的脉冲为0，封锁或门的脉冲为1。

2、增加选通脉冲

输入信号改变时，选通信号无效，封锁逻辑门；

输入信号稳定后，选通信号有效，允许逻辑门输出改变。

选通与门的脉冲为1，选通或门的脉冲为0。

需要考虑封锁信号、选通信号与输入信号的时序关系。

3、接滤波电容

利用电容的充放电作用消除冒险产生的窄脉冲（容量约几百皮法），但对正常信号脉冲波形也有影响。

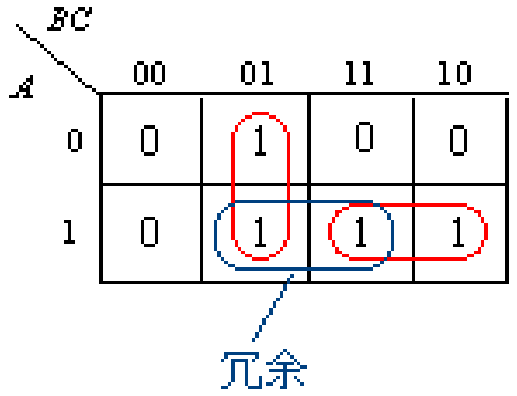
4、修改逻辑设计

若卡诺图中乘积项相邻 (圈相切) ,当相邻项取值不同的信号变化时将存在竞争冒险。(取值相同的信号满足逻辑值)

可改变电路,增加冗余项（包含相切圈的相邻最小项）代表的逻辑门，屏蔽互补信号的影响 。

例： $F=AB+\overline{B}C$

- 当A="1"且C="1"时， F=B+$\overline{B}$。
- 在B信号的下降沿 ， 由于$\overline{B}$滞后于B，使F="0"， 产生负脉冲冒险。



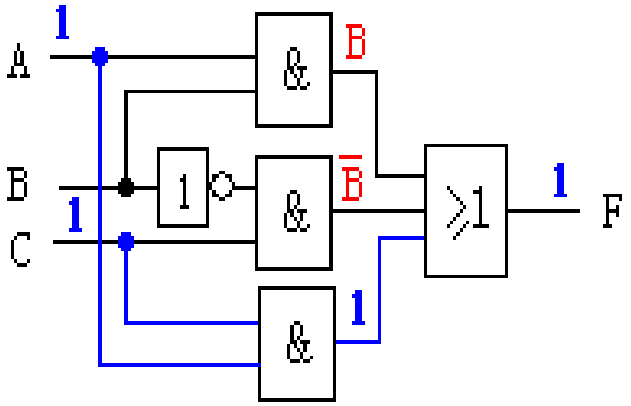
增加冗余项AC，使：

$F=AB+\overline{B}C=AB+\overline{B}C+AC$

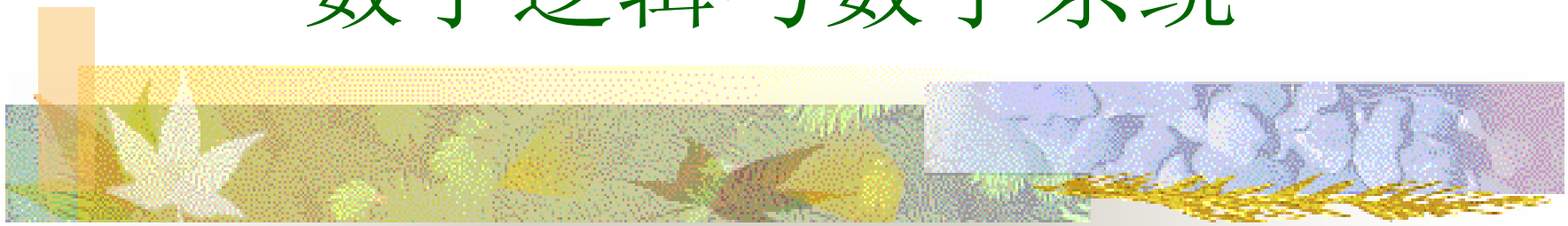
当A="1"且C="1"时，

F=B+$\overline{B}$+1="1"，

封锁或门,消除冒险。



数字逻辑与数字系统



第四章　　触发器

时序电路的特点：

电路的输出不仅与当时的输入有关，而且与电路原来的输出状态（输入控制历程）有关。

时序电路的结构和基本元件：

电路中有反馈路径，基本元件为能够记忆一位二值信号“1”或“0” 的双稳态触发器。

双稳态触发器的基本特性：

1、具有两个互补的输出端：Q、$\overline{Q}$，输出信号总是相反。

2、具有两个稳定的工作状态：

复位状态（Q=“0”）和置位状态（Q=“1”）。

双稳态触发器的特性方程（当控制条件满足时，触发器新的输出与激励输入X及原输出的逻辑关系）

$$Q^{n+1}=f(X, Q^n)$$

次态Q^{n+1}——触发器变化后的新状态；

现态Q^n ——触发器变化前的原状态

当控制条件满足时，触发器的新状态可以是：

置位：$Q^{n+1}=$“1”

复位：$Q^{n+1}=$“0”

保持：$Q^{n+1}=Q^n$（与原来的状态相同）

翻转：$Q^{n+1}=\overline{Q}^n$（与原来的状态相反）

双稳态触发器的触发方式（触发器状态变化时间的控制条件）：

直接触发：没有触发控制约束，激励变化时触发器状态立即变化。

电平触发：触发控制为开关电平信号E，E为有效电平时，触发器状态根据激励信号改变。

边沿触发：触发控制为时钟脉冲信号CP（Clock Pulse），触发器状态只在CP的有效沿（0->1上升沿或1->0下降沿）瞬间变化。

双稳态触发器的激励类型：

根据激励输入信号的名称定义：RS、D、JK、T和T'。

双稳态触发器的电路结构：

基本、同步、主从、维持阻塞等。

4.1 RS触发器

RS触发器具有两个开关特性的激励输入端R、S：

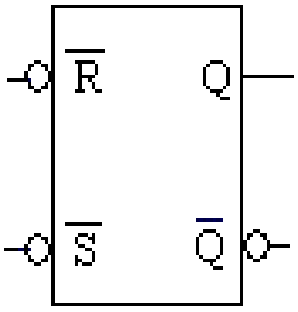
R的有效电平使触发器复位（Reset），Q=“0”；

S的有效电平使触发器置位（Set），Q=“1”。

R和S无效时触发器状态不变。

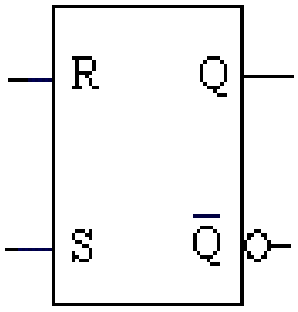
4.1.1直接触发的基本RS触发器—当R或S有效时触发器立即复位或置位。

低电平有效



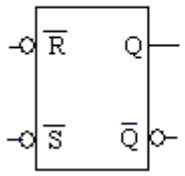
$\overline{R}$	$\overline{S}$	Q^{n+1}	功能
0	1	0	复位
1	0	1	置位
1	1	Q^n	保持
0	0	X	非法

高电平有效

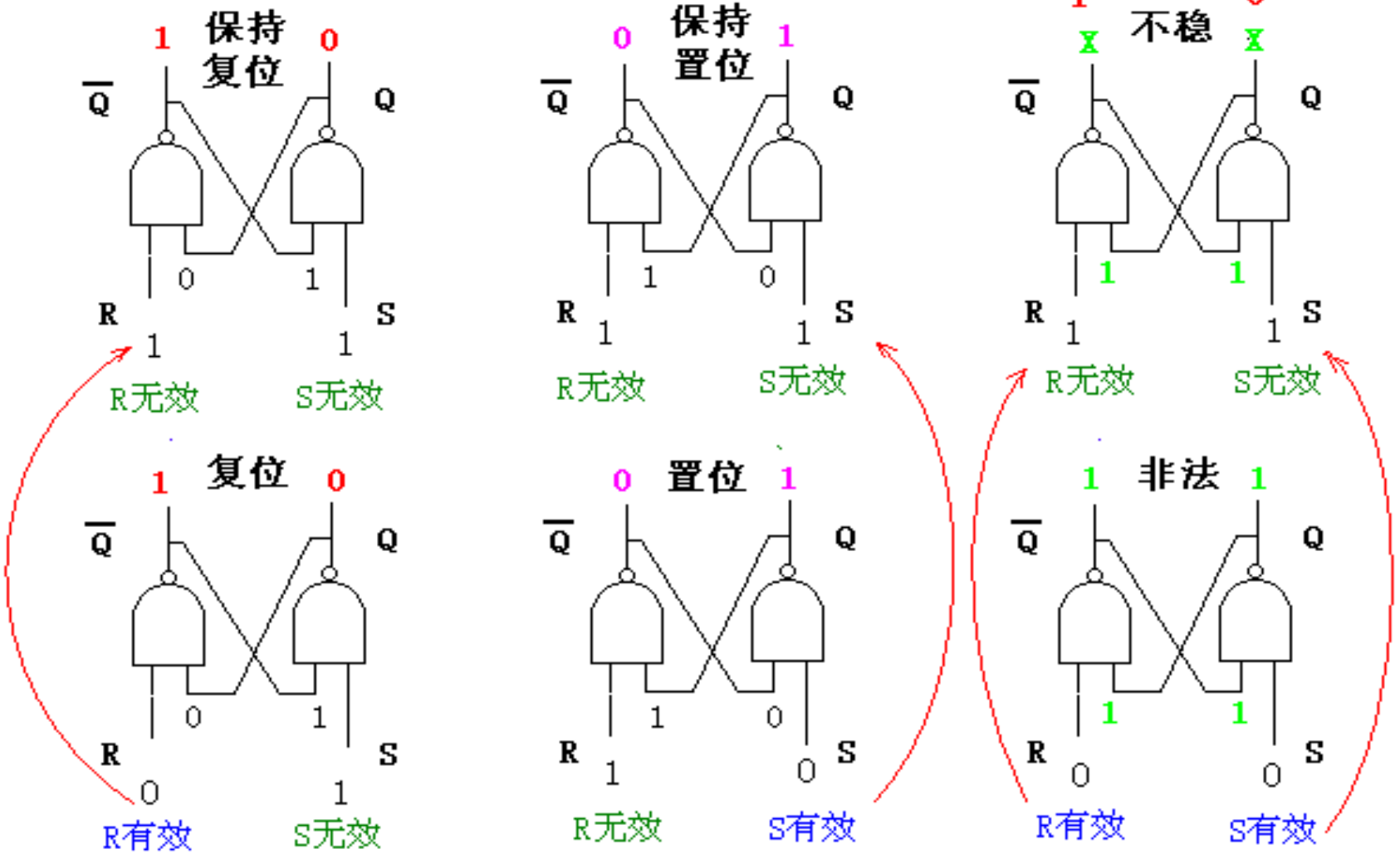


R	S	Q^{n+1}	功能
0	1	1	置位
1	0	0	复位
0	0	Q^n	保持
1	1	X	非法

低电平有效



$\overline{R}$	$\overline{S}$	Q^{n+1}	功能
0	1	0	复位
1	0	1	置位
1	1	Q^n	保持
0	0	X	非法

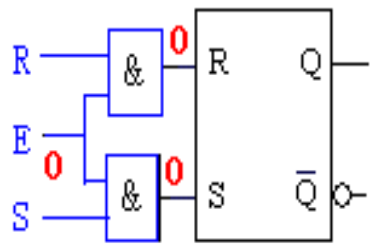


低电平有效的基本RS触发器的工作特性

4.1.2 由电平E控制的同步RS触发器

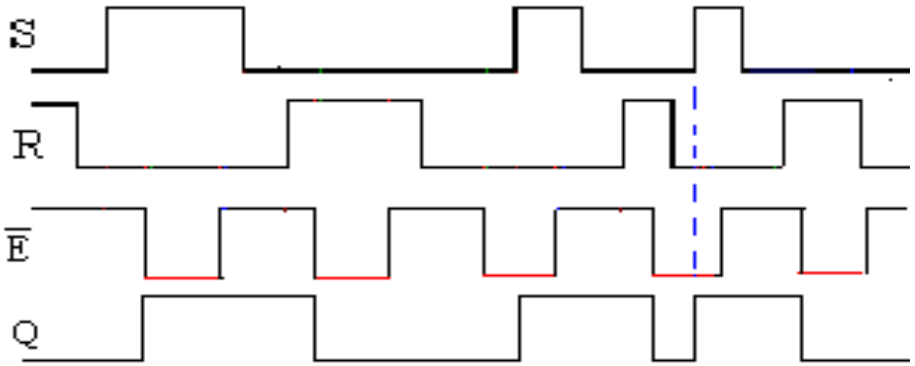
当E为无效电平时， RS触发器的状态不能改变。

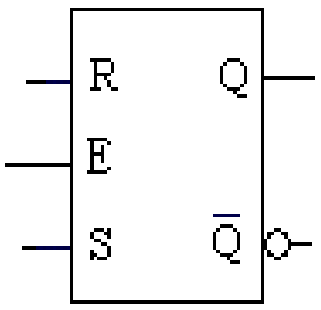
当E为有效电平时，允许激励输入R、S控制触发器状态改变。



同步RS触发器电路结构

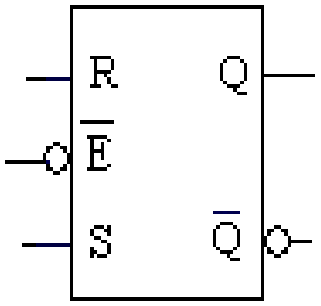
R	S	Q^{n+1}	功能
0	1	1	置位
1	0	0	复位
0	0	Q^n	保持
1	1	X	非法





使能高电平有效

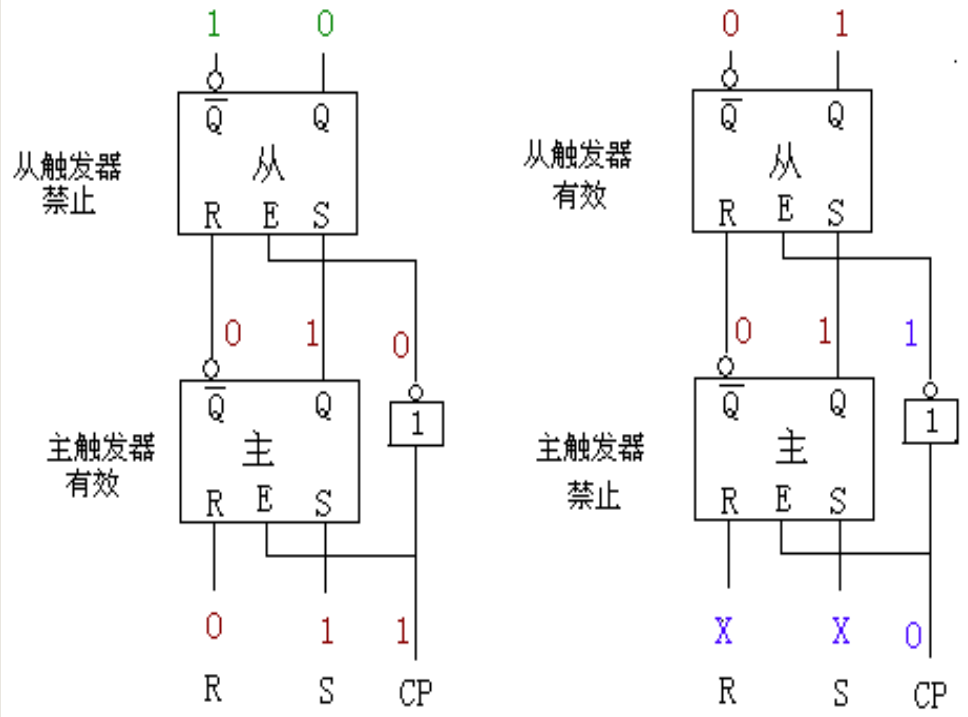
E	R	S	Q^{n+1}	功能
0	X	X	Q	保持
1	0	1	1	置位
1	1	0	0	复位
1	0	0	Q^n	保持
1	1	1	X	非法



使能低电平有效

$\overline{E}$	R	S	Q^{n+1}	功能
1	X	X	Q	保持
0	0	1	1	置位
0	1	0	0	复位
0	0	0	Q^n	保持
0	1	1	X	非法

4.1.3、负边沿控制的主从RS触发器



主从RS触发器的工作原理

主、从两个触发器的触发使能信号互补，两者分时工作，一个工作时，另一个被封锁。

1、当 CP=“1”期间，主触发器工作，输出受激励输入 S、R 的控制改变状态，但从触发器使能无效，所以输出保持不变。

2、当 CP 从“1”变为“0”，主触发器使能被禁止，从触发器工作，电路输出改变状态。

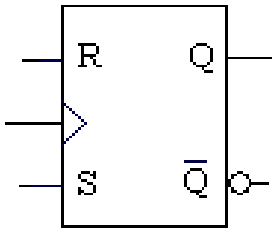
由于从触发器的激励不能再次改变，主从RS触发器的状态仅在 触发信号的有效沿改变一次

触发信号CP为脉冲Clock Puls

边沿控制RS触发器的逻辑符号和功能表

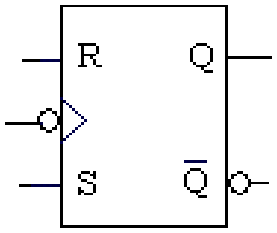
触发器的次态仅在时钟脉冲有效边沿时产生，由CP有效边沿前瞬间的RS信号控制。

正边沿RS触发器



E	R	S	Q^{n+1}	功能
↑	X	X	Q	不变
↑	0	1	1	置位
↑	1	0	0	复位
↑	0	0	Q^n	保持
↑	1	1	X	非法

负边沿RS触发器



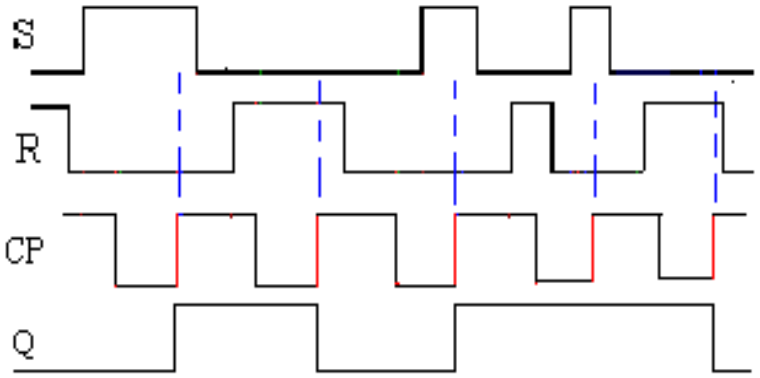
$\overline{E}$	R	S	Q^{n+1}	功能
↓	X	X	Q	不变
↓	0	1	1	置位
↓	1	0	0	复位
↓	0	0	Q^n	保持
↓	1	1	X	非法

三角表示脉冲边沿有效
小圈表示负边沿有效

高电平有效
RS触发器

特性方程 $\begin{cases} Q^{n+1} = S + \overline{R}Q^n \\ SR=0 \end{cases}$

约束条件:
R、S不能同时有效

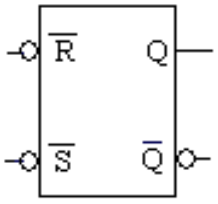


正边沿高电平有效RS触发器的时序波形

RS触发器的触发和激励比较

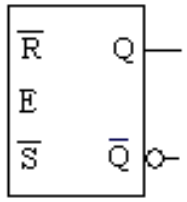
激励RS低电平有效

直接触发



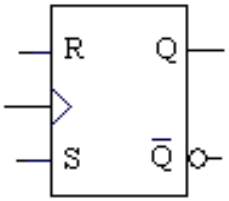
$\overline{R}$	$\overline{S}$	Q^{n+1}	功能
0	1	0	复位
1	0	1	置位
1	1	Q^n	保持
0	0	X	非法

高电平触发



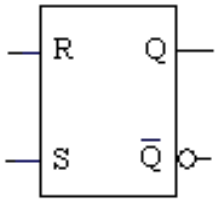
E	R	S	Q^{n+1}	功能
0	X	X	Q	保持
1	0	1	0	复位
1	1	0	1	置位
1	1	1	Q^n	保持
1	0	0	X	非法

正边沿触发

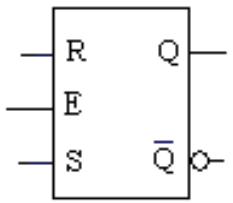


E	R	S	Q^{n+1}	功能
↑	X	X	Q	不变
↑	0	1	0	复位
↑	1	0	1	置位
↑	1	1	Q^n	保持
↑	0	0	X	非法

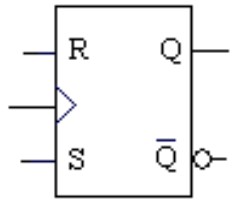
激励RS高电平有效



R	S	Q^{n+1}	功能
0	1	1	置位
1	0	0	复位
0	0	Q^n	保持
1	1	X	非法

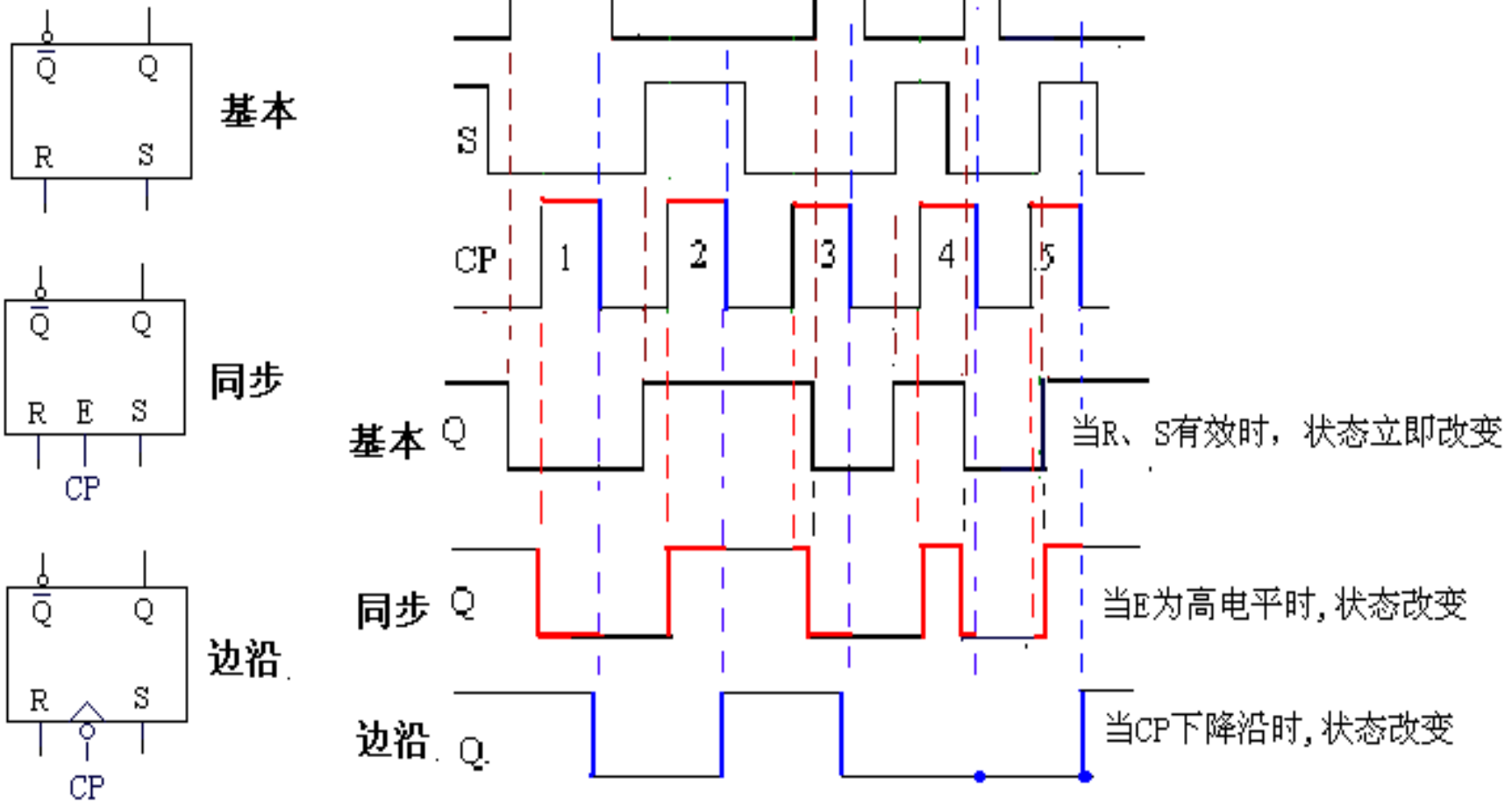


E	R	S	Q^{n+1}	功能
0	X	X	Q	保持
1	0	1	1	置位
1	1	0	0	复位
1	0	0	Q^n	保持
1	1	1	X	非法



$\overline{E}$	R	S	Q^{n+1}	功能
↑	X	X	Q	不变
↑	0	1	1	置位
↑	1	0	0	复位
↑	0	0	Q^n	保持
↑	1	1	X	非法

特性方程：	$Q^{n+1}=\overline{S}+RQ^n$	$Q^{n+1}=S+\overline{R}Q^n$	当触发条件满足时，特性方程成立
约束条件：	$S+R=1$	$S \cdot R=0$	



高电平有效的RS触发器的触发特性比较

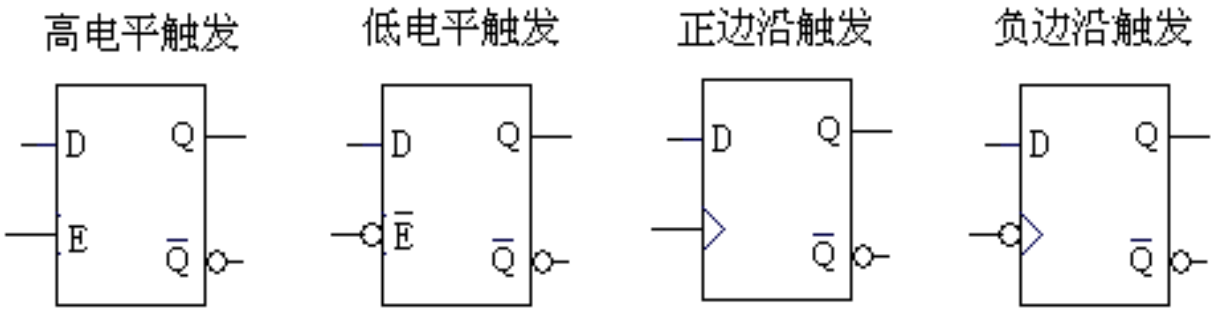
4.2其他激励功能的触发器

4.2.1 D触发器——当触发有效时，触发器状态与D相同，没有约束条件。

当触发条件满足时：　　特征方程:$Q^{n+1}=D$

常用集成D触发器:

同步D触发器　　边沿D触发器。



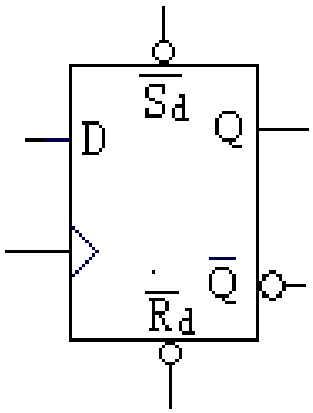
集成触发器的直接复位、置位功能

不受触发信号CP控制，立即影响触发器的状态，用于触发器的初始状态设置。

当触发器受触发信号CP控制时，直接控制输入Rd、Sd必须为无效电平。（例）

直接复位端Rd，直接置位端Sd
一般为低电平有效。

正边沿D触发器的逻辑符号和功能表



$\overline{R}_d$	$\overline{S}_d$	CP	D	Q^{n+1}	功能
0	1	X	X	0	复位
1	0	X	X	1	置位
0	0	X	X	X	非法
1	1	↑	0	0	触发
1	1	↑	1	1	

D触发器的典型应用

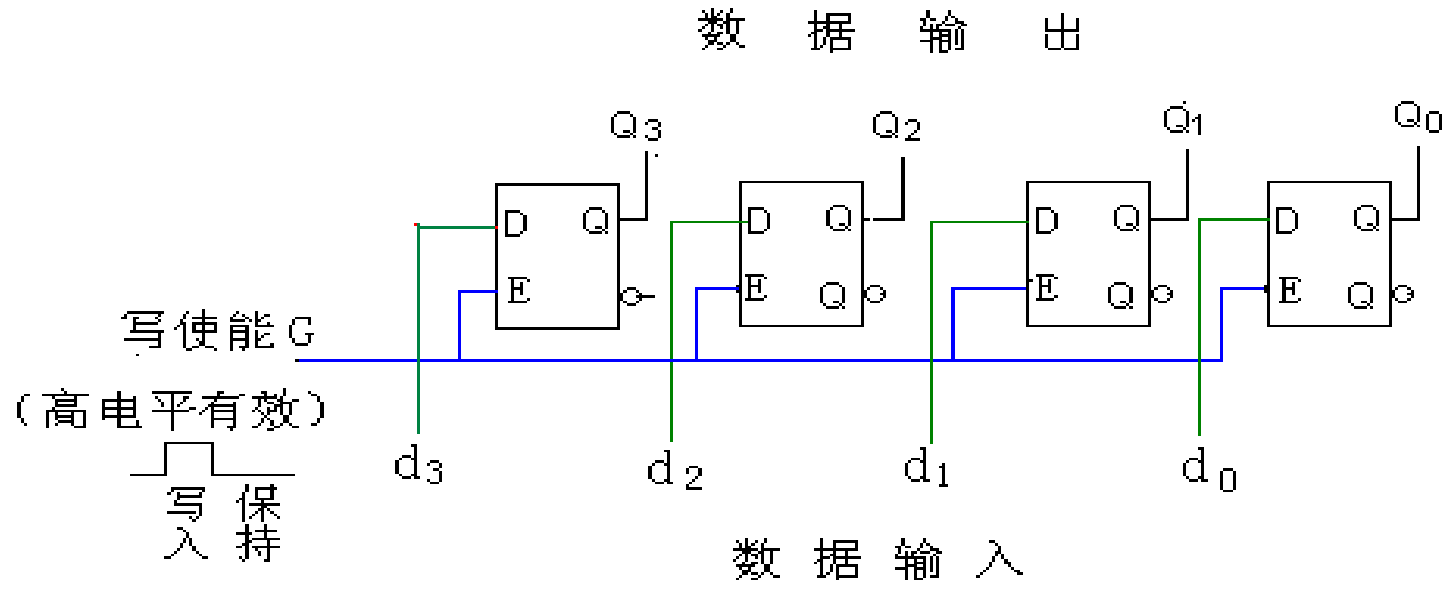
（5.2.1）锁存器、寄存器和移位寄存器

一、1个D触发器可以记忆1位二进制数，由同一个写使能信号共同控制的n个D触发器一次可记忆n位二进制数（一般n=4或8），根据触发方式不同称为锁存器或寄存器。

二、锁存器由同步D触发器构成，寄存器由边沿D触发器构成。

1、锁存器——由多个同步D触发器构成，适用于数据信号滞后于写使能信号有效的场合。

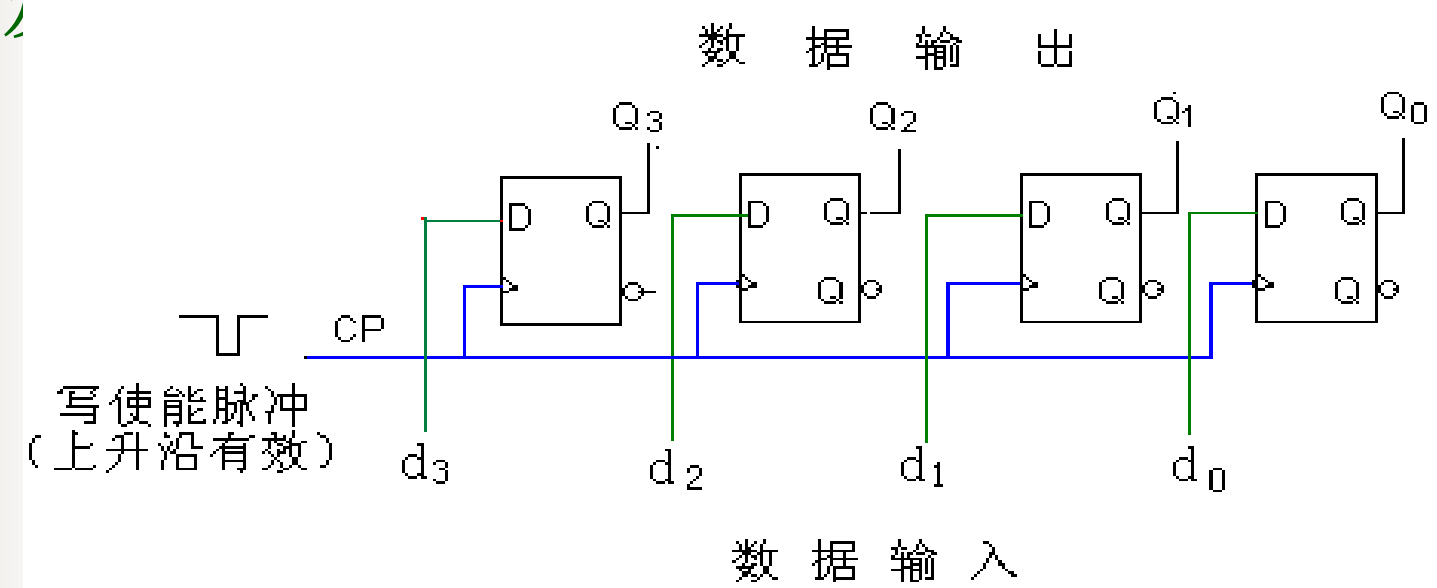
写使能信号G 为电平信号，当使能G有效时，输入端的数据Di被写入相应位的D触发器，Qi=Di；当使能G为无效电平时，触发器数据被锁存，Qi保持原来的状态，与输入端的数据Di无关。



由同步D触发器组成的4位锁存器

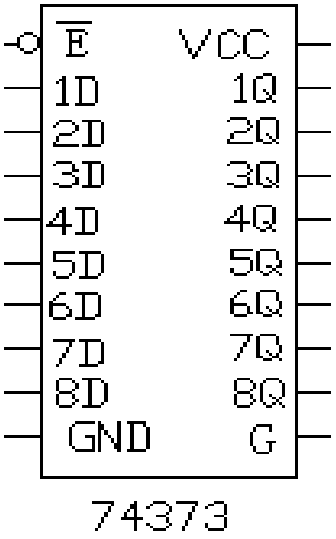
2、寄存器——由多个边沿触发器构成，适用于数据信号超前于写使能信号有效的场合。

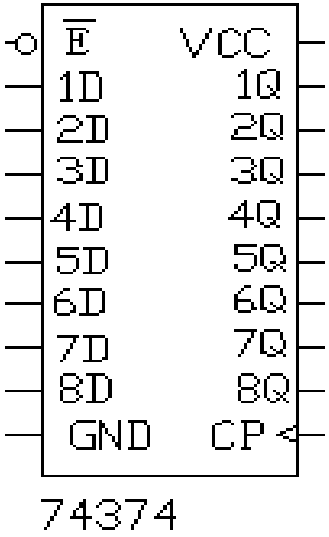
写使能信号CP为脉冲信号，当使能CP为有效边沿时，输入端的数据Di被写入相应位的D触发器，Qi=Di；否则，D触发器输出保持原来的状态，与输入端的数据无关。



由边沿D触发器组成的4位寄存器

集成8位锁存器74LS373和集成8位寄存器74LS374的逻辑符号





集成8位锁存器74LS373：
写使能高电平有效，输出使能E低电平有效。

集成三态输出8位寄存器74LS374：
写脉冲CP上升沿有效，输出使能E低电平有效。

例4-3：单脉冲产生电路。

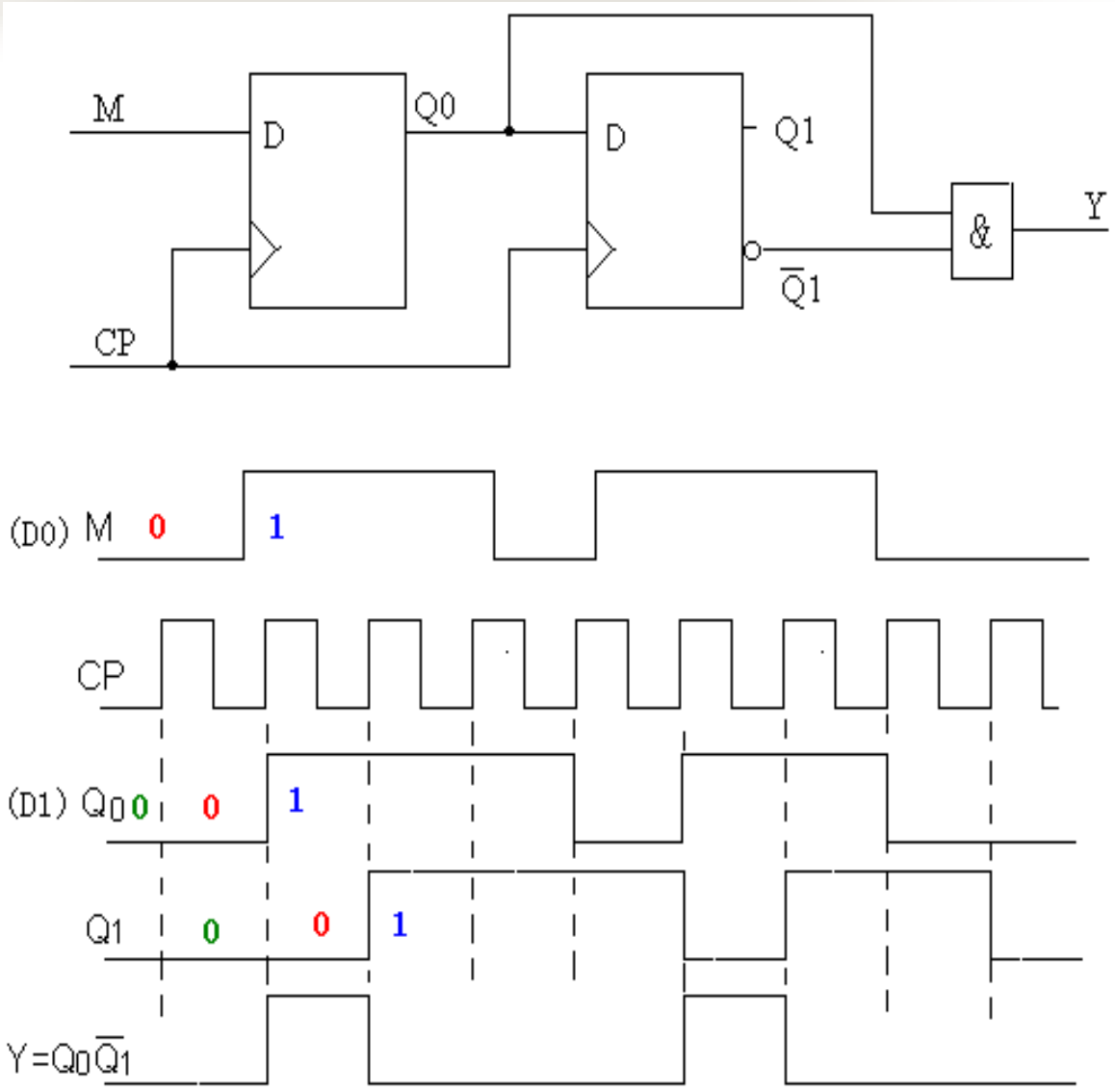
$D_0=M$

$D_1=Q_0$

$Q_0^{n+1}=D_0=M$

$Q_1^{n+1}=D_1=Q_0^n$

将脉冲宽度（高电平1的时间）大于时钟周期的输入M信号转换成脉冲宽度恒定为一个CP周期的输出信号Y。



4.2.2 边沿JK触发器

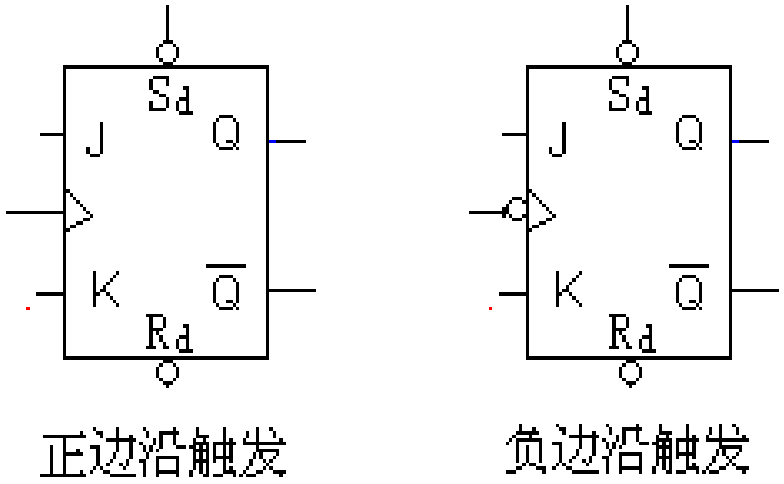
有两个编码量的激励输入J、K，当触发有效时，可以控制触发器状态分别为

置位（Q^{n+1}=“1”）、复位（Q^{n+1}=“0”），

保持（$Q^{n+1}= Q^n$）、翻转（$Q^{n+1}= Q^n$）。

特性方程:$Q^{n+1}=J\overline{Q}^n+\overline{K}Q^n$

逻辑符号



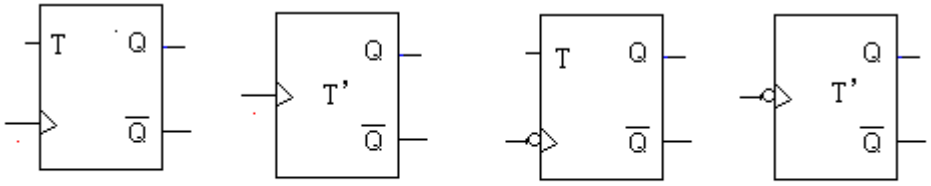
正边沿触发　　负边沿触发

触发功能表

J	K	Q^{n+1}	功能
0	0	Q^n	保持
0	1	0	复位
1	0	1	置位
1	1	$\overline{Q}^n$	翻转

4.2.3 T触发器和T’（计数型）触发器

1、T触发器-----特征方程：$Q^{n+1}=T\overline{Q}^n+T\overline{Q}^n$



正边沿触发　　负边沿触发

2、T’触发器------特征方程：　$Q^{n+1}=\overline{Q}^n$

没有激励输入，触发有效时,状态总是翻转，类似用一位二进制码累计触发脉冲的个数。

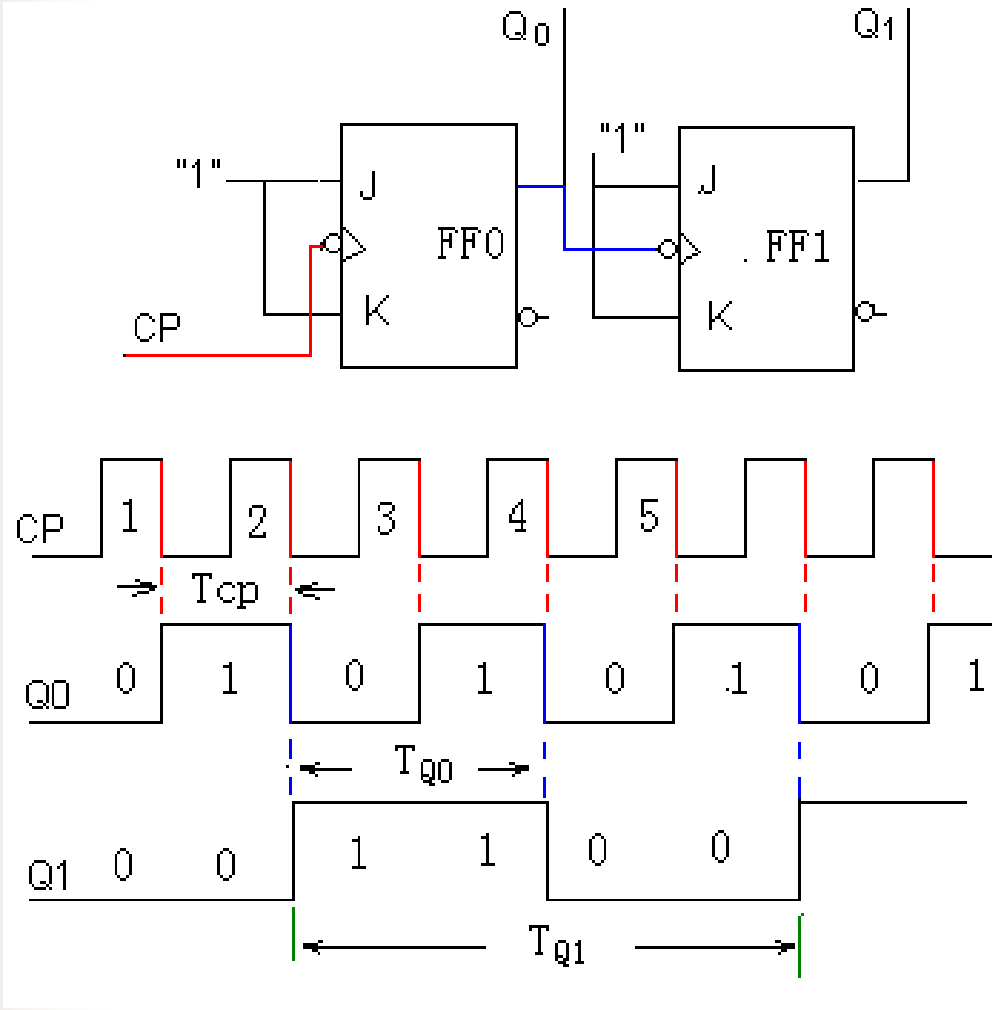
当激励J=K=1　时，JK触发器具有计数特性。

当激励$D=\overline{Q}$ 时，D触发器具有计数特性。

当激励T=1 时，T触发器具有计数特性。

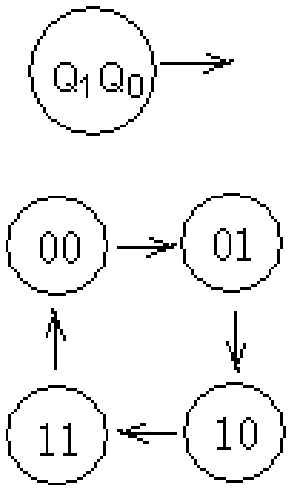
计数型触发器的状态输出信号周期是触发脉冲周期的一倍。具有二分频功能。

例4-6 两个JK触发器连成计数型T'触发器,分析电路.



触发器

状态转换图：



功能：

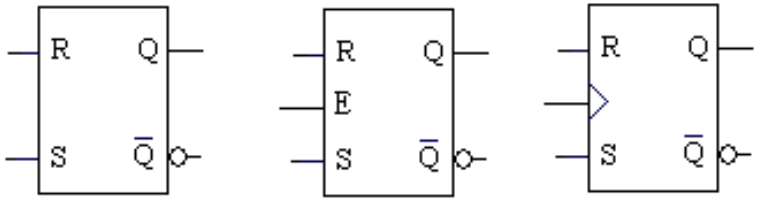
四进制加计数器

（模4）

2位二进制加计数器

RS触发器

激励RS高电平有效



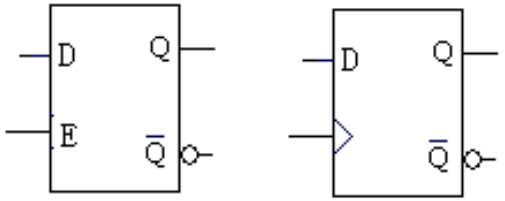
特性方程：

$$Q^{n+1}=S+\overline{R}Q^{n}$$

约束条件：$S \cdot R = 0$

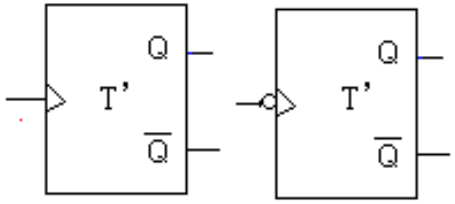
D触发器

高电平触发　　正边沿触发



$$Q^{n+1}=D$$

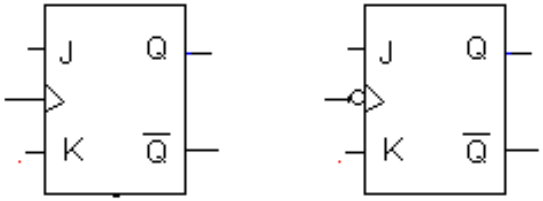
T'触发器



$$Q^{n+1}=\overline{Q}^{n}$$

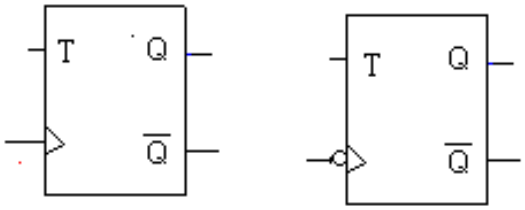
JK触发器

正边沿触发　　负边沿触发



$$Q^{n+1}=J\overline{Q}^{n}+\overline{K}Q^{n}$$

T触发器



$$Q^{n+1}=T\overline{Q}^{n}+\overline{T}Q^{n}$$

触发器的控制特性:

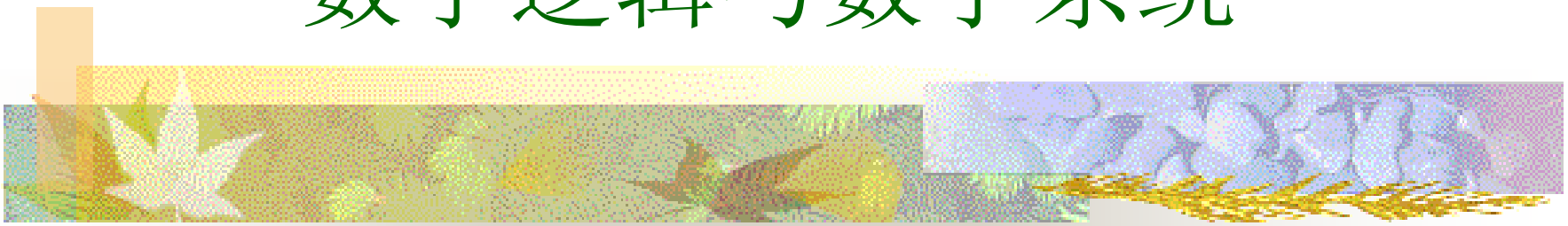
1、直接置位S_D、复位R_D控制功能最优先，控制信号有效时触发器立即被置位（Q=1）或复位（Q=0）。

2、当R_D、S_D无效、触发控制（CP）无效时触发器状态不变。

3、当R_D、S_D无效、触发控制CP有效时触发器次态受激励控制：

$$Q^{n+1}=D \qquad Q^{n+1}=J\bar{Q}^n+\bar{K}Q^n$$

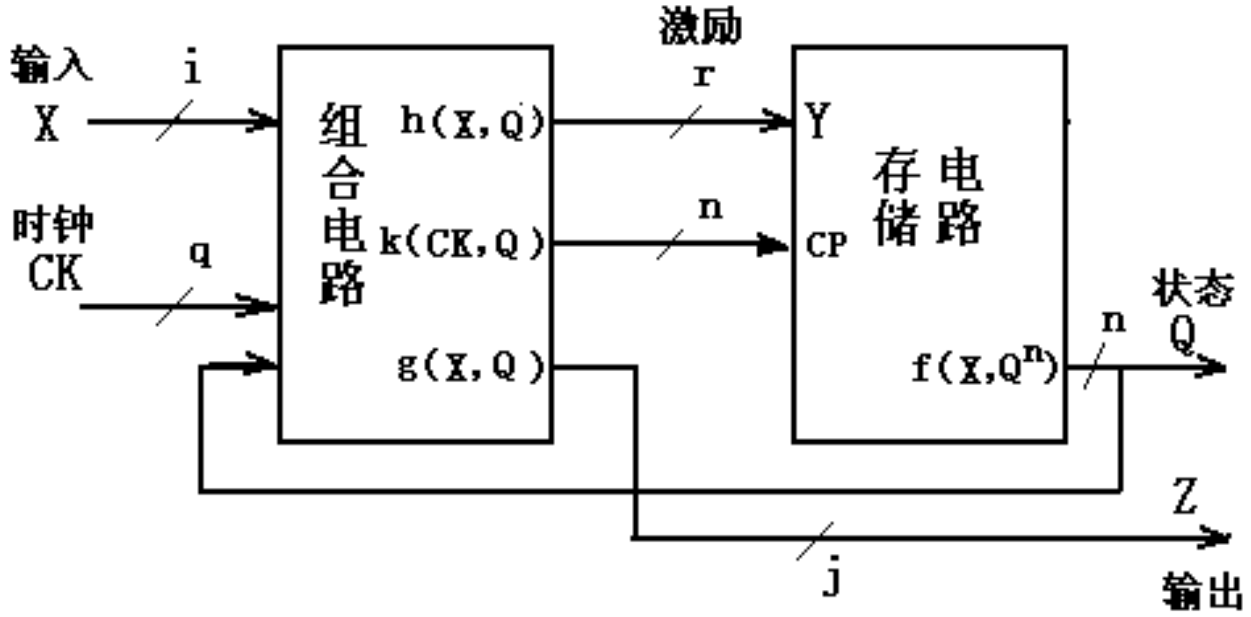
数字逻辑与数字系统



第五章　　时序逻辑电路

时序逻辑电路的基本结构

由触发器和控制激励和输出的组合逻辑电路构成。



时序逻辑电路的分类：

根据触发控制方式分类：

同步时序电路——电路中所有触发器由同一时钟触发。

异步时序电路——电路中至少有一个触发器的触发时钟与其它触发器不同。

根据输出控制方式分类：

米利（mealy）型时序电路——输出Z受触发器状态Q和外部输入X控制。Z=f（X，Q）

莫尔(moore)型时序电路——输出Z仅受触发器状态Q控制，与外部输入X无关。Z=f（Q）

时序逻辑电路的描述方法：

- 时钟方程：　CP=*k*（CK，Q）
- 激励方程：　Y=*h*（X，Q）
- 次态方程：　$Q^{n+1}=f$（x、Q^n）
- 输出方程：　Z=*g*（X，Q）
-
- 状态转换表和状态卡诺图：
  输入、现态（函数变量）与次态、输出（函数值）的关系。

- 时序波形图：
  输入与输出数字信号的时序对应关系图。

状态转换图：

状态转换图是以拓扑图形式描述时序电路的转换关系。

(1) 电路的每个状态用一个圈表示，圈中填入状态符Si或状态码值，

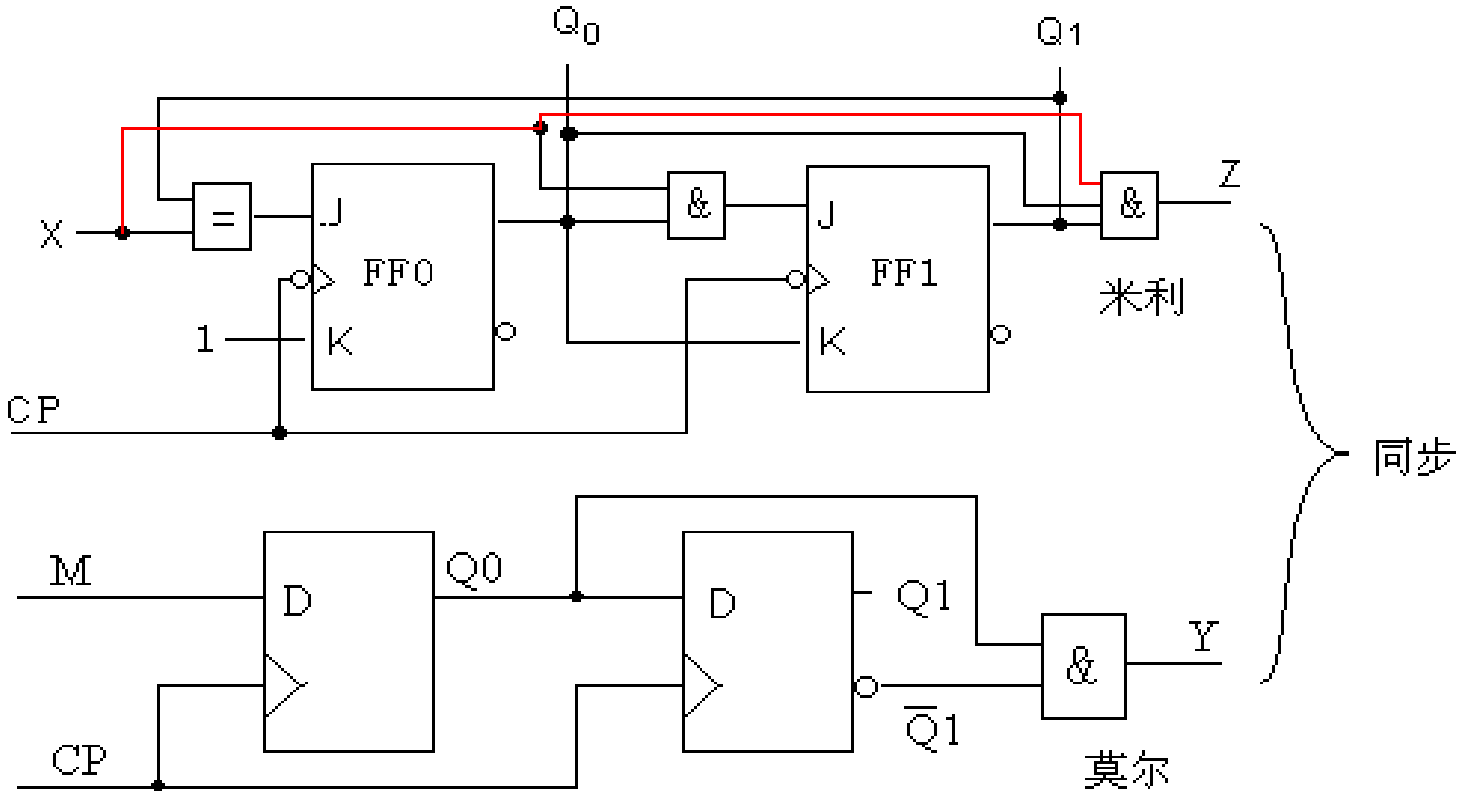
(2)圈外用箭头表示状态转换关系，箭头从某现态指向其次态，

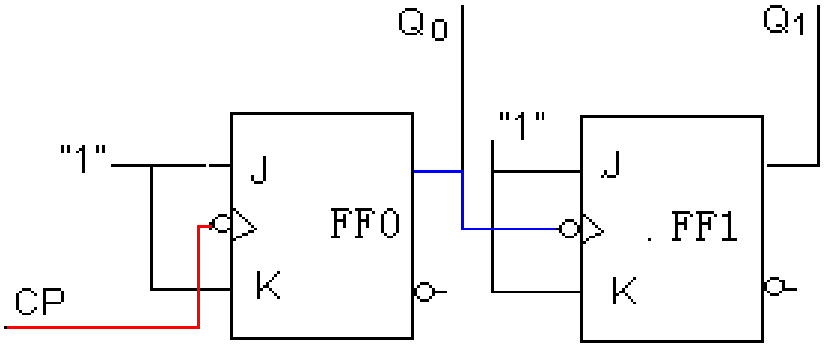
(3)箭头旁标出控制该状态转换的控制条件X。

(4)输出

Mealy：输出与输入一起标在箭头旁。用斜杠区分。

Moore：输出Z标在状态圈内，用斜杠区别于状态。


Q0
Q1
X
=
J
FF0
K
1
CP
&
J
FF1
K
&
Z
米利
同步
M
D
Q0
D
Q1
Q1
&
Y
CP
莫尔


Q0
Q1
"1"
J
FF0
K
CP
"1"
J
FF1
K

异步

二进制计数器--------

功能：累计时钟脉冲的个数、分频、定时、产生节拍脉冲

特点：状态码随CP周期性循环，循环个数称为“模”M。

分类：

- 1、按计数器按计数体制
- 若n个触发器构成的计数器具有模$M=2^n$、且状态码变化有自然二进制数序的特点，就称为n位二进制计数器；若模$M<2^n$、或状态码变化不符合二进制数序特点称M（模）进制计数器，最常用的是“模”为10的十进制计数器。
- 2、按状态码值的增减趋势分
- 计数器状态码变化有数序特点、且呈递增趋势变化的称加计数器；呈递减趋势变化的称减计数器；在信号控制下既可递增也可递减计数的称可逆计数器。

3、按计数脉冲引入方式分

- 计数脉冲直接控制计数器电路中所有触发器的时钟触发端CP，称同步计数器；否则就称异步计数器。

计数器的自启动能力：

1、若n个触发器构成的计数器的模M小于2^n，则有2^n-M个无效状态存在。

2、计数器在正常运行时的状态周期性循环，不可能出现无效状态码。但在电路上电（合上电源）瞬间，计数器的状态是随机的，可能出现无效状态码。

3、如果计数器处于无效状态时，随着计数脉冲输入能够转入有效状态循环，则表示计数器具有自启动能力，否则电路没有自启动能力，将陷于无效状态的死循环。

5.3 时序逻辑电路的分析方法:

根据电路图分析状态转换规律和输出，确定电路功能。

分析时序逻辑电路的一般步骤

1、由电路连接关系写逻辑函数式:

（1）各触发器的时钟控制方程（同步时序电路可以不列）

$CP=f_0$（CP，Q）

（2）电路的输出方程　$Z=f_1$（X、Q）

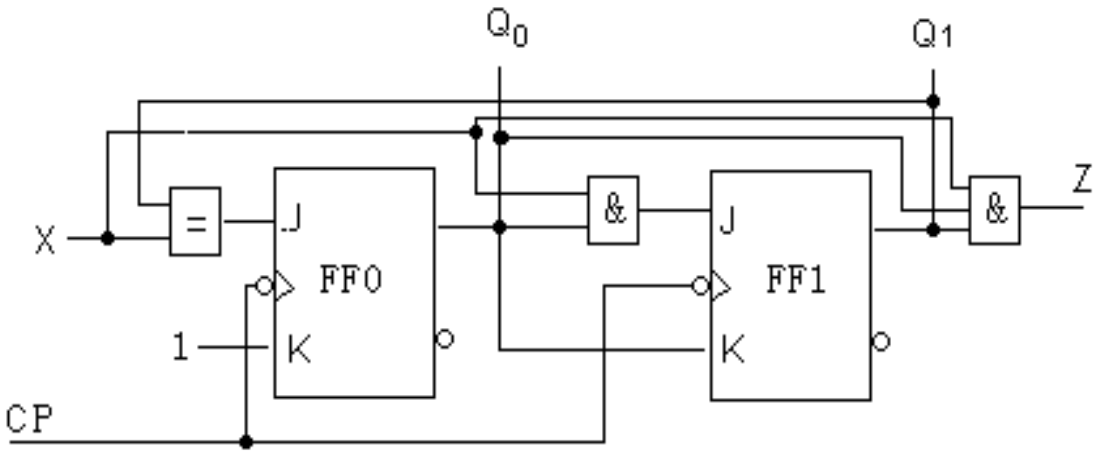
（3）各触发器的输入驱动（激励）方程

Y（J、K、D、T、R、S）$=f_2$（X、Q）

2、将驱动方程代入相应触发器的特性方程，得电路的状态方程:$Q_i^{n+1}=f_3$（X、Q^n）

3、根据状态方程和输出方程列电路的状态表，画出状态转换图或时序波形图。

4、分析电路功能及自启动能力。



状态转换表

输入	现态			次态			输出
X	Q_1^n	Q_0^n	PS	Q_1^{n+1}	Q_0^{n+1}	NS	Z
0	0	0	A	0	0	A	0
0	0	1	B	0	0	A	0
0	1	0	C	1	1	D	0
0	1	1	D	0	0	A	0
1	0	0	A	0	1	B	0
1	0	1	B	1	0	C	0
1	1	0	C	1	0	C	0
1	1	1	D	0	0	A	1

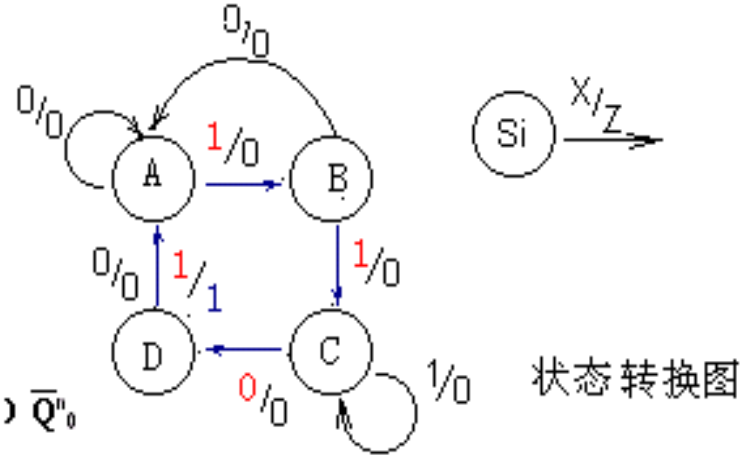
输出方程　$Z = Q_1 Q_0 X$

激励方程　$J_1 = X Q_0$ ；$K_1 = Q_0$

$J_0 = X \oplus Q_1$ ；$K_0 = 1$

状态方程　$Q^{n+1}_1 = X Q^n_0 \overline{Q}^n_1 + \overline{Q}^n_0 Q^n_1$

$Q^{n+1}_0 = X \overline{Q}^n_1 \overline{Q}^n_0 + \overline{X} Q^n_1 \overline{Q}^n_0 = (X \oplus Q^n_1)\, \overline{Q}^n_0$



状态转换图

功能：序列信号检测电路（米利型），
当输入X的连续信号序列为
“1101”时，输出Z=1。

例：输入X: 0101101010111010
状态S: ABABCDAABABCCCDAA
输出Z: 0000001000000010

异步时序电路分析

至少有一个触发器的时钟不是由计数脉冲控制，而是由其他触发器的输出控制。所以，不是所有的计数脉冲都能使该触发器发生变化，仅当其触发条件满足时才能受其激励控制，否则状态保持不变。

因此，时钟不受计数脉冲控制的触发器必须列其时钟控制的逻辑方程，将次态方程转换为与电路输入脉冲同步的状态方程。

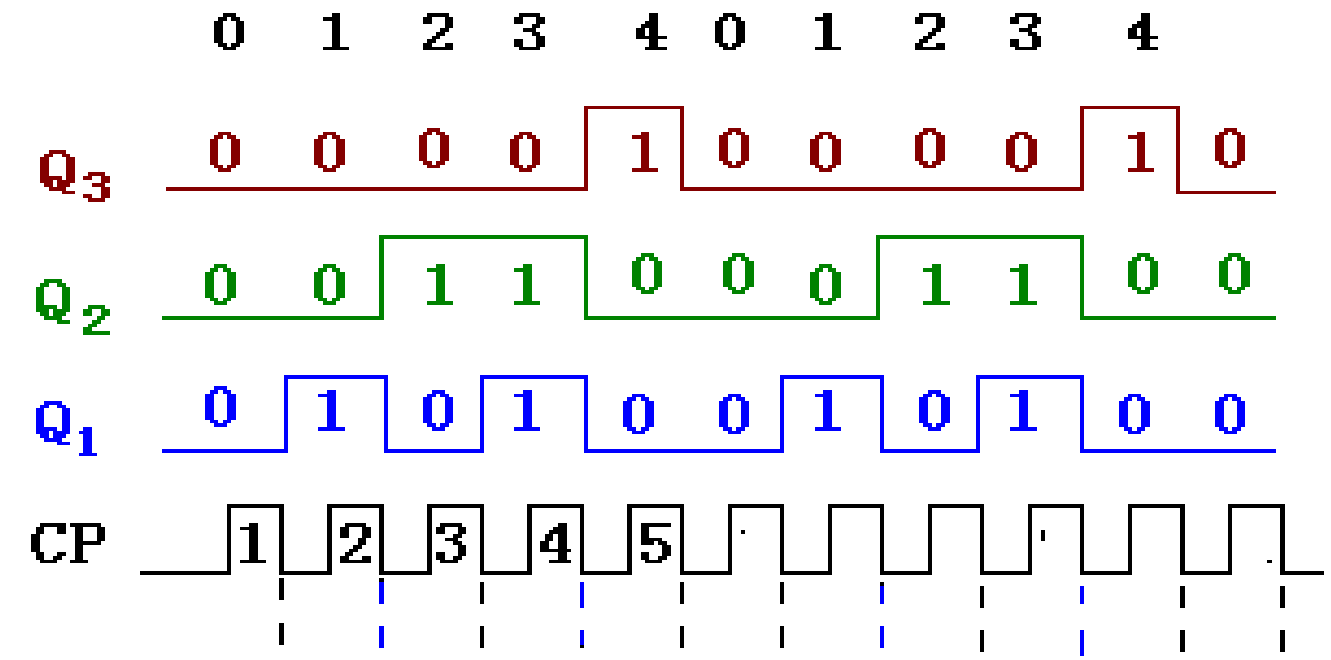
负边沿JK触发器：$Q^{n+1}=(J\overline{Q}^{n}+\overline{K}Q^{n})CP\downarrow+Q^{n}\overline{CP\downarrow}$

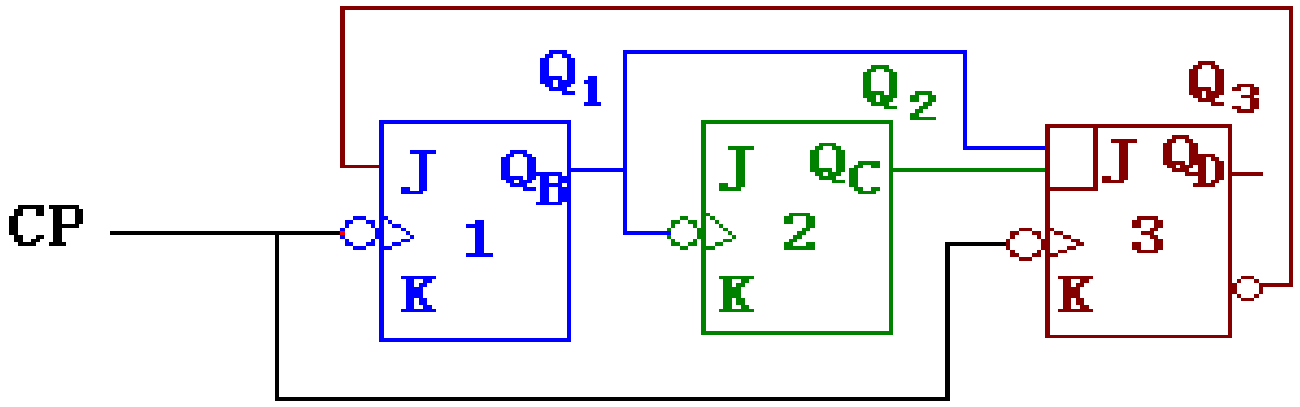
正边沿D触发器：$Q^{n+1}=DCP\uparrow+Q^{n}\overline{CP\uparrow}$

$CP\downarrow=Q_X^{n}\overline{Q_X^{n+1}}$　　$CP\uparrow=\overline{Q}_X^{n}Q_X^{n+1}$

Q_X为CP所连触发器的输出

异步五进制计数器


0 1 2 3 4 0 1 2 3 4
Q3 0 0 0 0 1 0 0 0 0 1 0
Q2 0 0 1 1 0 0 0 1 1 0 0
Q1 0 1 0 1 0 0 1 0 1 0 0
CP 1 2 3 4 5


Q1
Q2
Q3
CP
J
QB
K
1
J
QC
K
2
J
QD
K
3

5.4同步时序电路的设计

5.4.1 设计步骤

1、根据功能要求确定所需输入变量、输出变量以及状态的个数，画出原始状态图或原始状态表。

2、化简状态（合并等价状态）列出最简状态转换表（或图）

3、 确定触发器个数、类型及状态编码值——赋于每个状态一组二进制码代入状态表得各触发器的状态转移表。

4、根据状态转移表或次态卡诺图列各触发器的次态方程和输出方程。

5、将次态方程与触发器的特性方程比较，得各触发器的激励驱动方程。

6、根据各触发器的激励方程和输出方程画逻辑电路图。

7、若有无效状态存在，分析自启动能力。不能自启动的修改电路。

原始状态表化简

- 状态——记忆事件。

- 等价状态——两个不同的状态在输入相同时输出相同、次态满足下列条件之一：
- （1）相同。
- （2）互为对方次态。
- 比如A的次态是B，B的次态是A，则状态A、B等价。
- （3）分别是互相等价的状态。
- 比如A的次态是C，B的次态是E，若状态C、E等价，则A、B等价。
- 相互等价的状态记忆的是同一个事件，可以合并为一个状态。使电路简化

例7

试用负边沿JK触发器设计“110”序列脉冲检测器。电路有一个串行信号输入端X和一个检测状态输出端Z。电路原理如图所示，当X连续输入的三个信号是“110”时，输出Z为“1”。

解：因为要求检测的序列是三个连续信号。所以，有两种方法可以实现检测要求。

（1）存储电路只“记忆” X的前两个连续信号的状态（共有4种可能的序列），再根据最新输入的X对3个连续信号作出判断产生输出Z，为米利型电路。

当信号满足序列时输出Z立即为1,不受CP控制。

（2）存储电路“记忆”X的三个连续信号（共8种可能的序列），输出Z不受输入X的控制，仅由电路状态决定，电路为莫尔型。

当信号满足序列且CP有效后输出Z才为1，与CP同步。

米利型电路实现

例：110序列信号检测

前两个连续信号序列	状态符号
0 0	S_0
0 1	S_1
1 0	S_2
1 1	S_3

原始状态表

S_i^n	(X=0) S_i^{n+1} / Z	(X=1) S_i^{n+1} / Z
S_0	S_0 / 0	S_1 / 0
S_1	S_2 / 0	S_3 / 0
S_2	S_0 / 0	S_1 / 0
S_3	S_2 / 1	S_3 / 0

等价

最简状态表

记忆

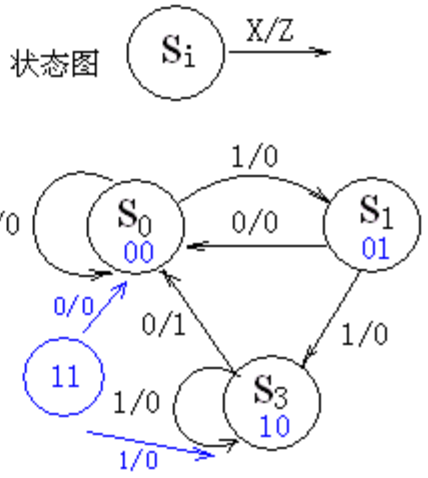
初始状态

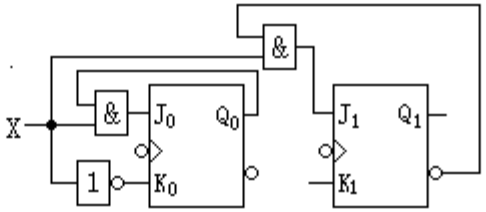
"1"有效

"11"有效

S_i^n	(X=0) S_i^{n+1} /Z	(X=1) S_i^{n+1} /Z
S_0	S_0 / 0	S_1 / 0
S_1	S_0 / 0	S_3 / 0
S_3	S_0 / 1	S_3 / 0

状态图





计数器编码表

状态符	Q_1Q_0
S_0	0 0
S_1	0 1
S_3	1 0

状态转移表

输入	现 态		次 态		输出
X	Q_1^n	Q_0^n	Q_1^{n+1}	Q_0^{n+1}	Z
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	1
1	0	0	0	1	0
1	0	1	1	0	0
1	1	0	1	0	0

触发器的次态方程

$$Q_1^{n+1} = XQ_0^n\overline{Q}_1^n + XQ_1^n$$

$$Q_0^{n+1} = X\overline{Q}_1^n\overline{Q}_0^n$$

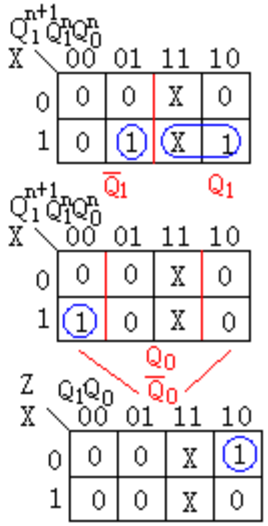
激励方程

$$J_1 = XQ_0,\ K_1 = \overline{X}$$

$$J_0 = X\overline{Q}_1,\ K_0 = 1$$

输出方程

$$Z = \overline{X}\,Q_1^n\overline{Q}_0^n$$

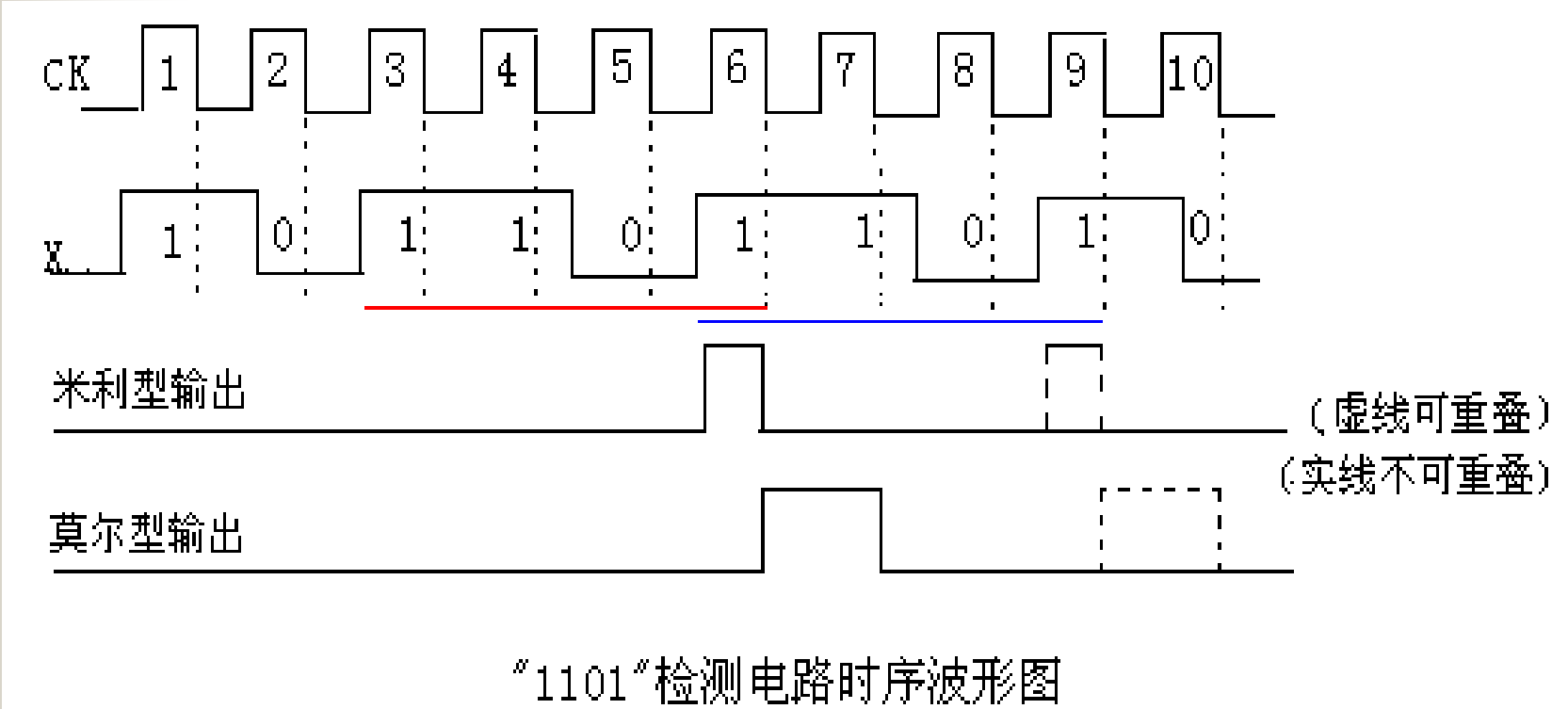


输入序列：X 0 1 0 1 1 0 1 0 1 0 1 1 1 0 1 0

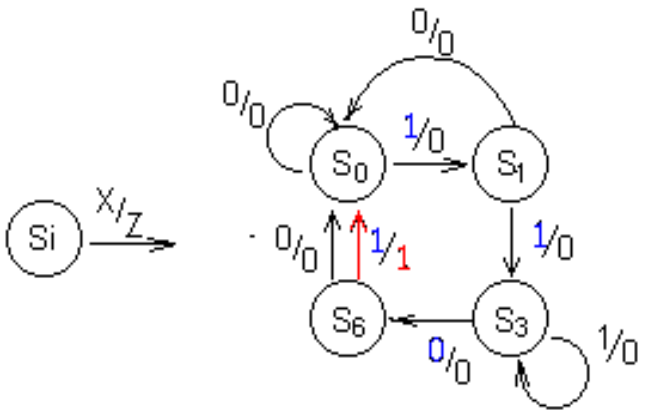
状态：S_i $S_0S_1S_0S_1S_3S_0S_1S_0S_1S_0S_1S_3S_3S_0S_1S_0$

输出：Z 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0

例:试用负边沿JK触发器设计一个“1101”序列检测器。当X连续输入的四个信号是“1101”时，输出Z为“1”。
设检测序列可以重叠，即前一序列的末位“1”可以作为下一序列的首位信号“1”。



“1101”检测电路时序波形图



米利型“1101”不可重复检测电路状态图

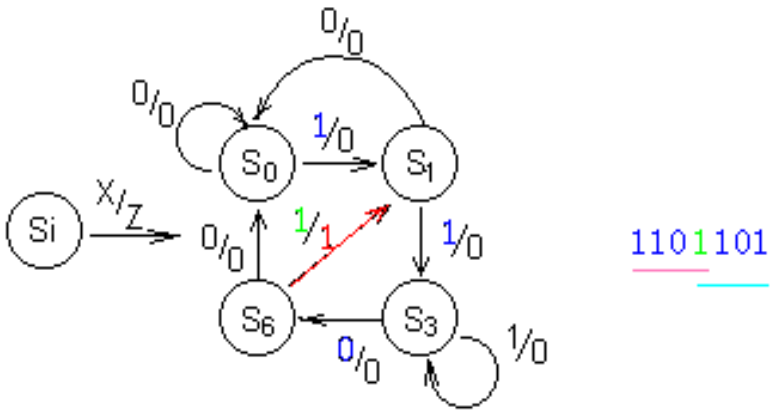


图5-4-4 米利型“1101”可重复检测电路状态图

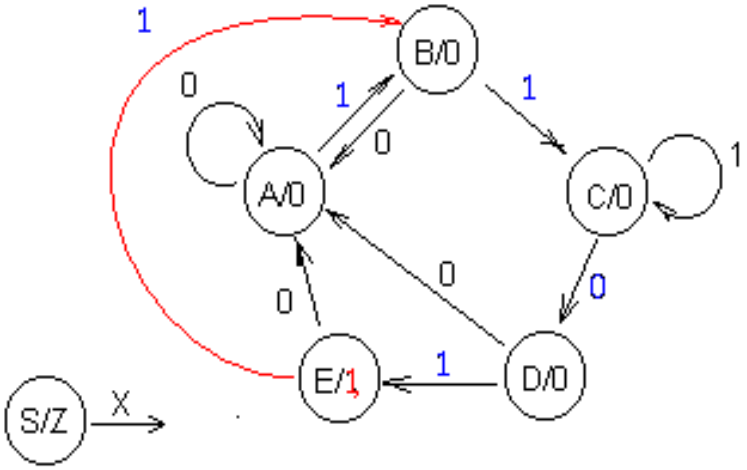
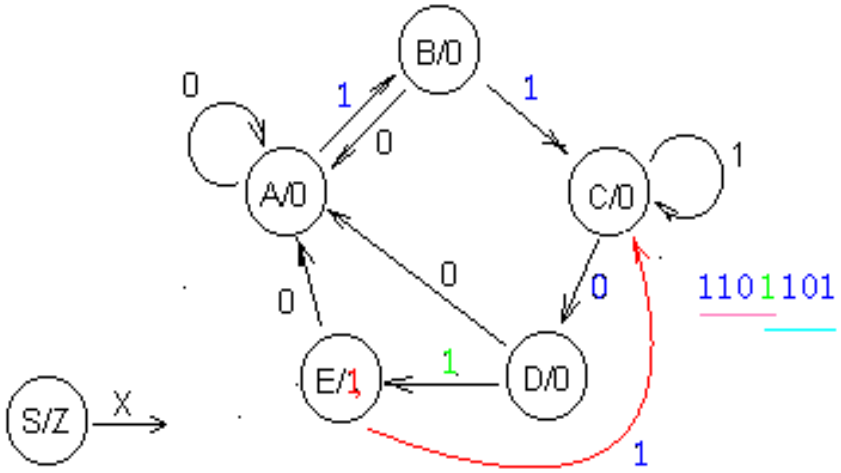


图5-4-8 莫尔型“1101”不可重复检测电路状态图



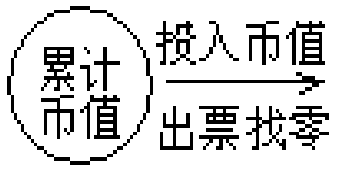
莫尔型“1101”可重复检测电路状态图

车票自动售票系统状态定义

控制要求：

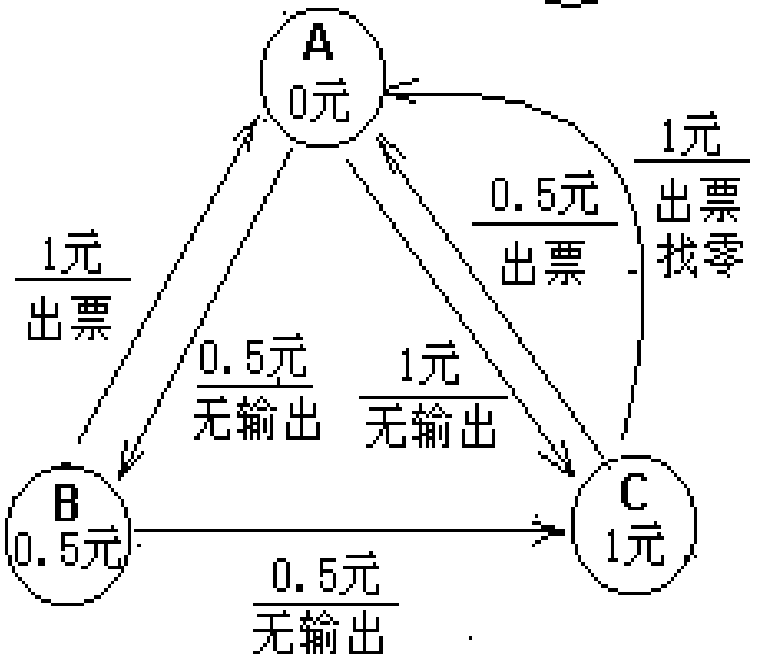
售出车票面额为1.5元，可投硬币值为0.5元和1元两种。

当投入币值达到车票面额时出票，超过面额时找零。



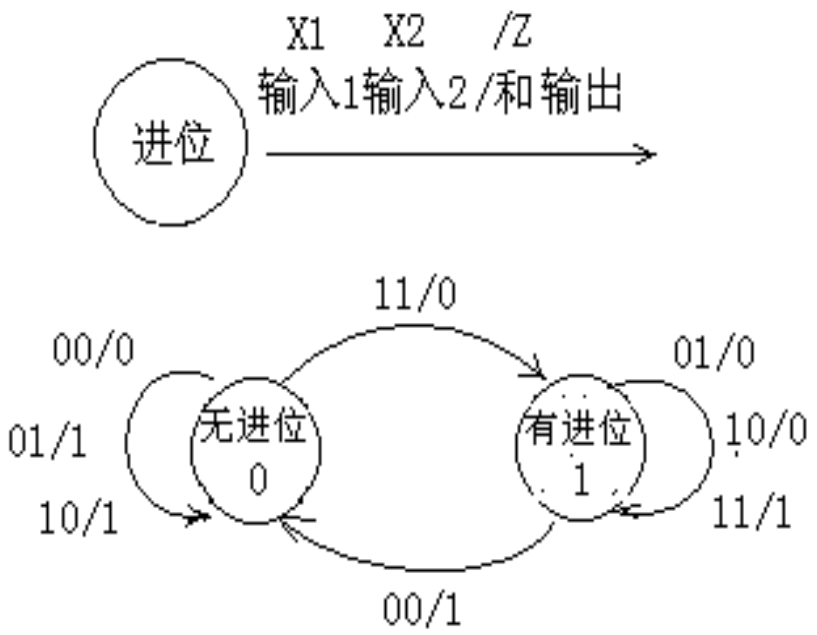
米利型电路：

用三个状态记忆已经投入的累计币值(0元、0.5元、1元)，根据新投入的币值决定是否出票、是否找零以及新的累计币值 。



例9：串行加法器对两个串行（输入位序先低后高）二进制数逐位相加。电路需要记忆每次相加后的进位输出，以便与后一拍输入的高一位序的加数相加。所以需要两个状态分别表示有进位(1)或无进位(0)。

用米利型电路实现的状态转换图：



状态表

	X1X2=00		X1X2=01		X1X2=10		X1X2=11	
PS	NS	Z	NS	Z	NS	Z	NS	Z
0	0	0	0	1	0	1	1	0
1	0	1	1	0	1	0	1	1

每位的和输出=

本位的输入1加输入2加前一位的进位状态

第六章　数字系统

数字系统——含有控制部件并能按控制信息有序操作的逻辑系统.

6.1.1 数字系统的基本模型

一、基本功能——能够传输、存储、处理信息。

信息传输：

1、并行传输——一组n位的信息在n条信号线上同时传递。

2、串行传输——一组n位的信息在一条信号线上依位序定时逐位传递。

信息存储：

采用动态存储器或静态存储器保存处理过程需要的信息。

信息处理：

按控制要求对信息进行算术运算或逻辑运算。

二、基本结构——输入、输出部件；控制、处理、存储部件

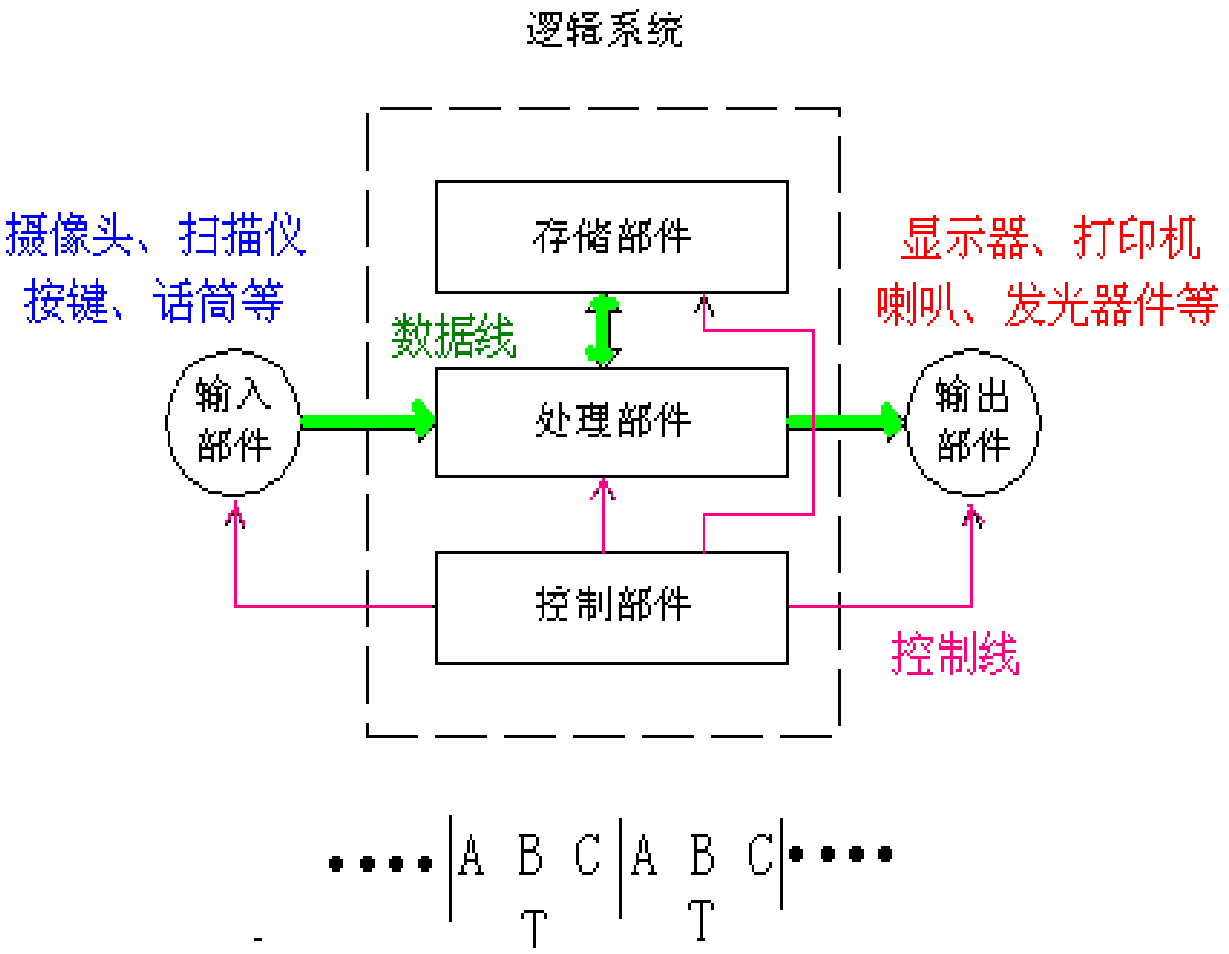
在数字系统中，执行部件（存储、处理、输入、输出部件）在控制部件发出的信息控制下顺序、循环操作。

一个执行周期T有若干个状态（比如三个）：

取数A

处理B

存数C



6.1.2数字系统与逻辑功能部件的区别

一、功能不同

数字系统含有控制器，能按控制信息管理逻辑功能部件（子系统）有序操作。

逻辑功能部件能够完成某一具体的逻辑任务。

二、设计方法不同

逻辑功能部件的设计方法是：按任务要求列真值表或状态表，化简后得各输出函数逻辑表达式，完成电路设计。

数字系统的设计方法是：按控制任务要求划分子系统，再分别设计各子系统和控制器的具体逻辑结构（由上而下）；

最后整合所有部件检查设计是否满足要求（由下而上）。

构成最小数字系统的基本子系统（必需的功能部件）：
控制器、寄存器、存储器、运算处理单元、数据总线。

6.2.1算术逻辑运算单元ALU(Arithmetic Logical Unit)

功能:由控制信息控制对输入的二进制数据信息进行算术运算或逻辑处理操作。

- n位控制信息码，可以实现2^n种运算功能。
- 如74LS381的控制码有3位：M2、M1、M0，可以对输入的两组4位二进制码A、B实现8种操作：

- M2M1M0： 000 、 001 、 010、 011、 100、 101、 110、 111
- 输出 F=： 清零、 B减A、A减B、A加B、A异或B、A或B、A与B、预置A

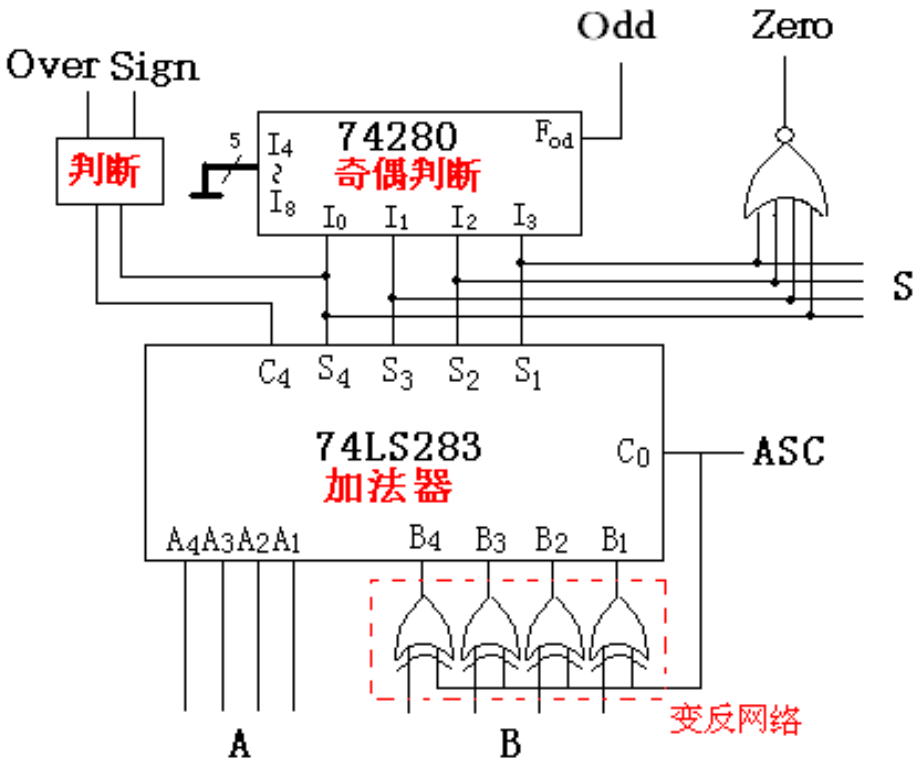
例：在一位二进制命令ASC的控制下，4位加法器74283采用二进制补码对输入的两组4位二进制数A、B进行加或减的运算操作，由逻辑判断电路输出4个运算结果状态标志并用触发器保存：

Over——溢出标志，ALU输出数值大于4位二进制码时为1；

Sign——符号标志，ALU输出为负数时为1；

Zero——全零标志，ALU输出全0时为1；

Odd——奇偶标志，ALU输出码中1的个数是奇（偶）数时为1。



A直接输入到加法器，B由控 制信号ASC通过异或门输入加法器。

当ASC=0时，异或门输出B的原码，加法器实现A加B；

当ASC=1时，异或门输出B的反码，加法器以加补码方式实现A减B。

$A+\overline{B}+1=A-B$

ASC

功能:存储处理信息

通用寄存器：一般用来暂存参与ALU运算的数据或运算结果以及作为程序堆栈（保存子程序返回地址等）、间址寄存器（存放存储器或寄存器的访问地址）等。数据传输的速度较存储器快，。

寄存器堆（组）由若干寄存器（2^n，如64、128、256等）组成，写数据时由地址码控制数据分配器将输入数据分配到指定的寄存器；读数据时由地址码控制数据选择器从指定寄存器中读出数据输出。

双端口寄存器堆有两个数据输出端口，可以由两组地址码选择两个寄存器的数据同时输出。

特殊功能寄存器：存放ALU的状态标志、功能部件（串口管理、中断管理、堆栈管理等）的控制命令或状态等。

6.3数据通路

6.3.1总线——多源信号的共同传输通路。

1、总线功能——允许多个信息源分时传送给多个目标。

单向总线：信息的传输方向是固定从源流向负载。

双向总线：总线上的信息传输方向可以改变，即总线各端的部件既可以是信息源又可以是传输目标。

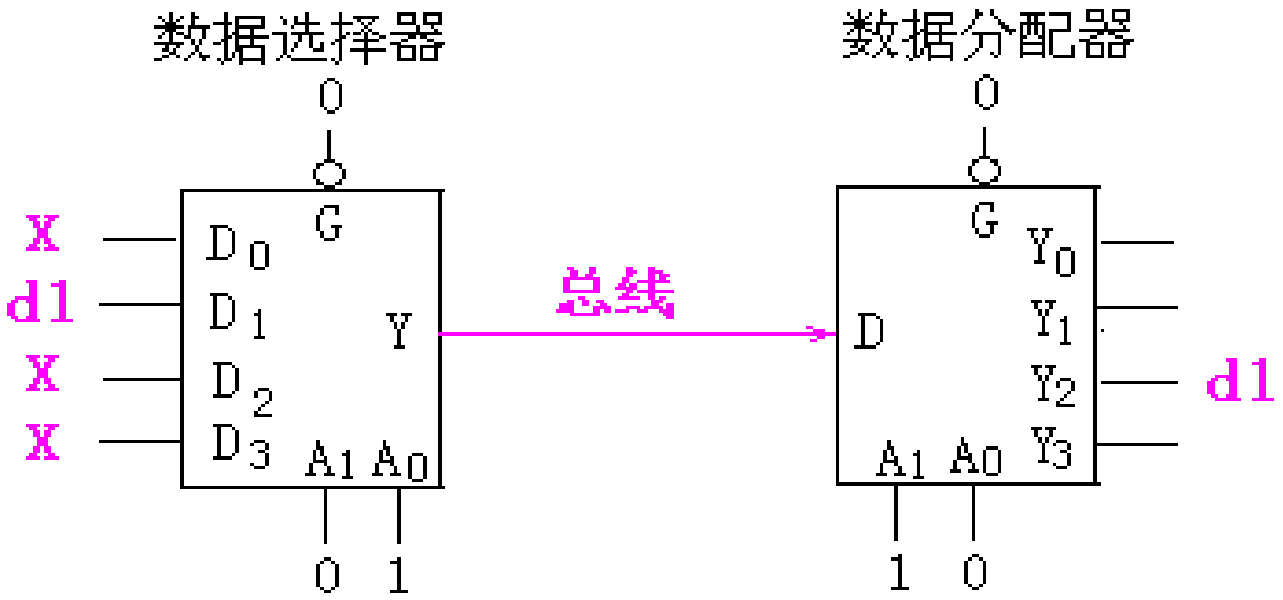
2、总线逻辑结构——

信息源部件的输出可以通过数据选择器或三态门挂在总线上，任意瞬时选择信号或使能信号只能允许（有效）一个信号源使用总线传输数据。

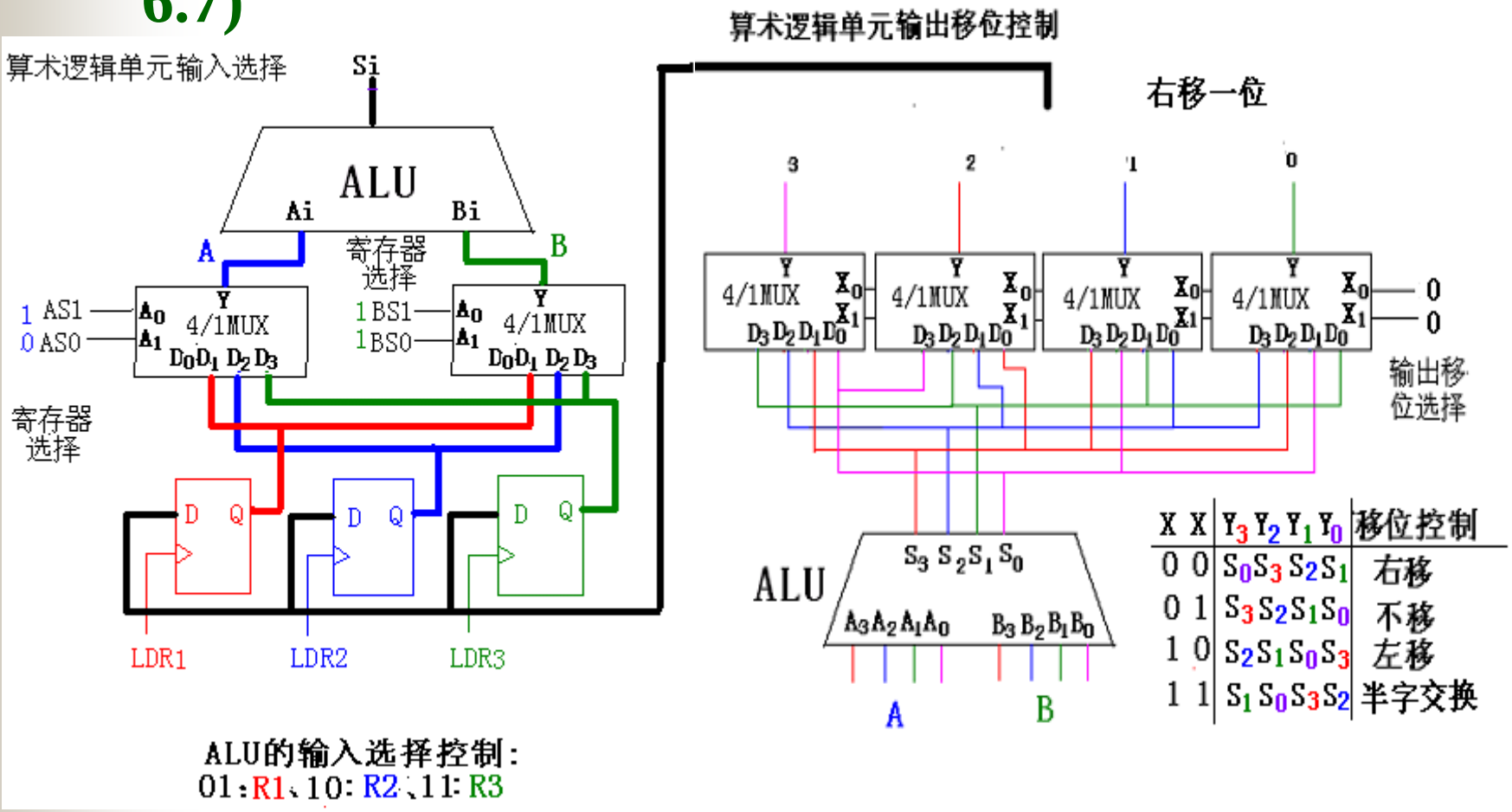
双向总线的上各部件的输出必需通过三态门挂上总线。

目标部件的输入可以由输出分配器或使能信号控制，有选择地接收总线上传输的数据。

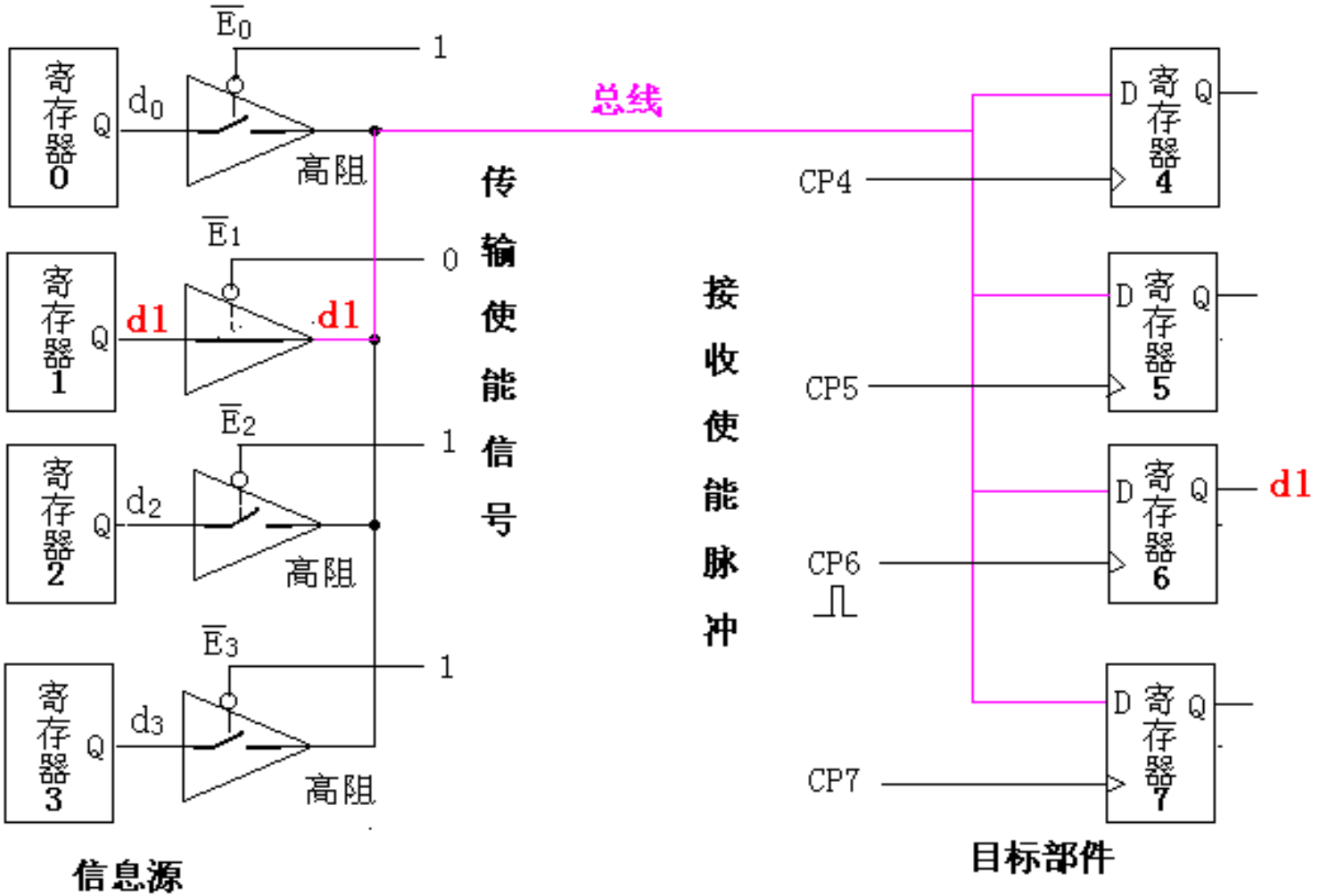
1）多路选择器、分配器构成的单向总线



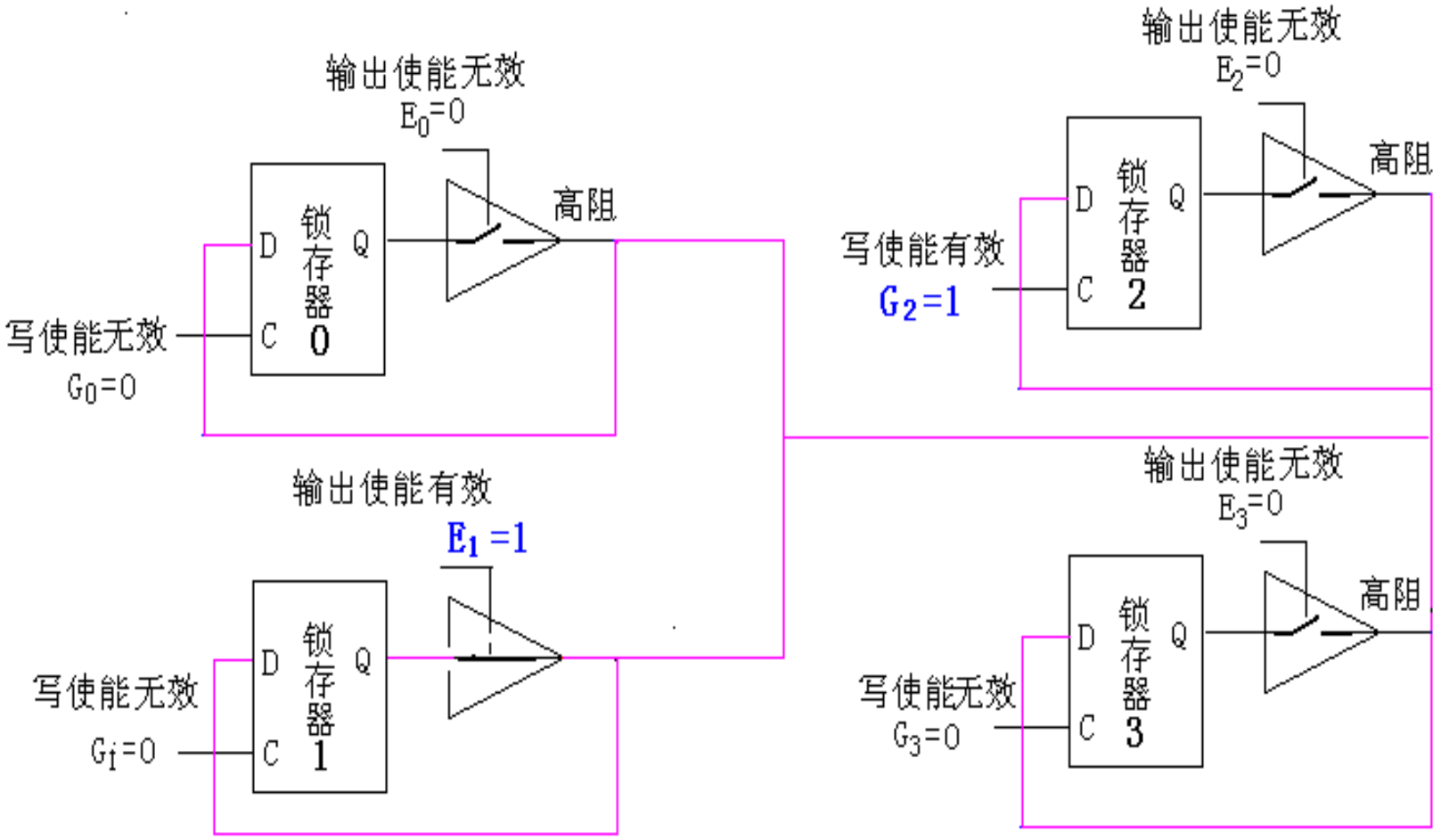
具体应用:用多路选择器构成的数据总线结构(图6.7)



2) 三态门构成的单向总线



3)一位4终端双向总线电路结构——4个锁存器的数



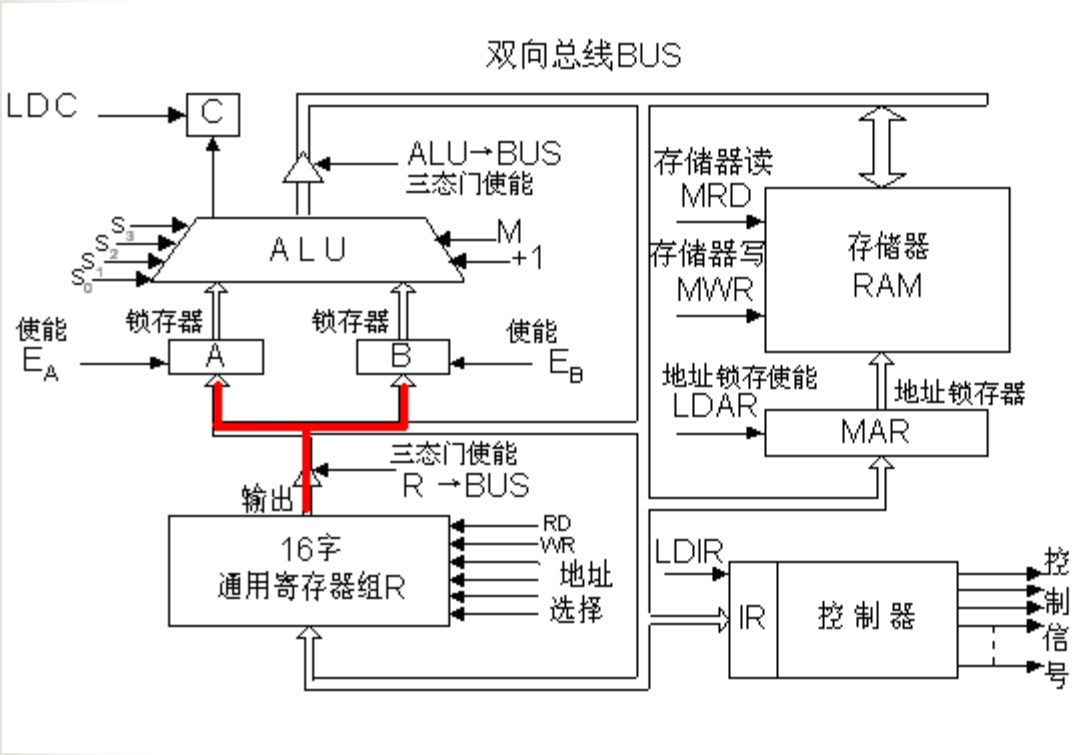
双向总线锁存器1的信息传向锁存器2

6.3.2数据通路

数据通路——子系统通过总线联结形成的数据传输通路

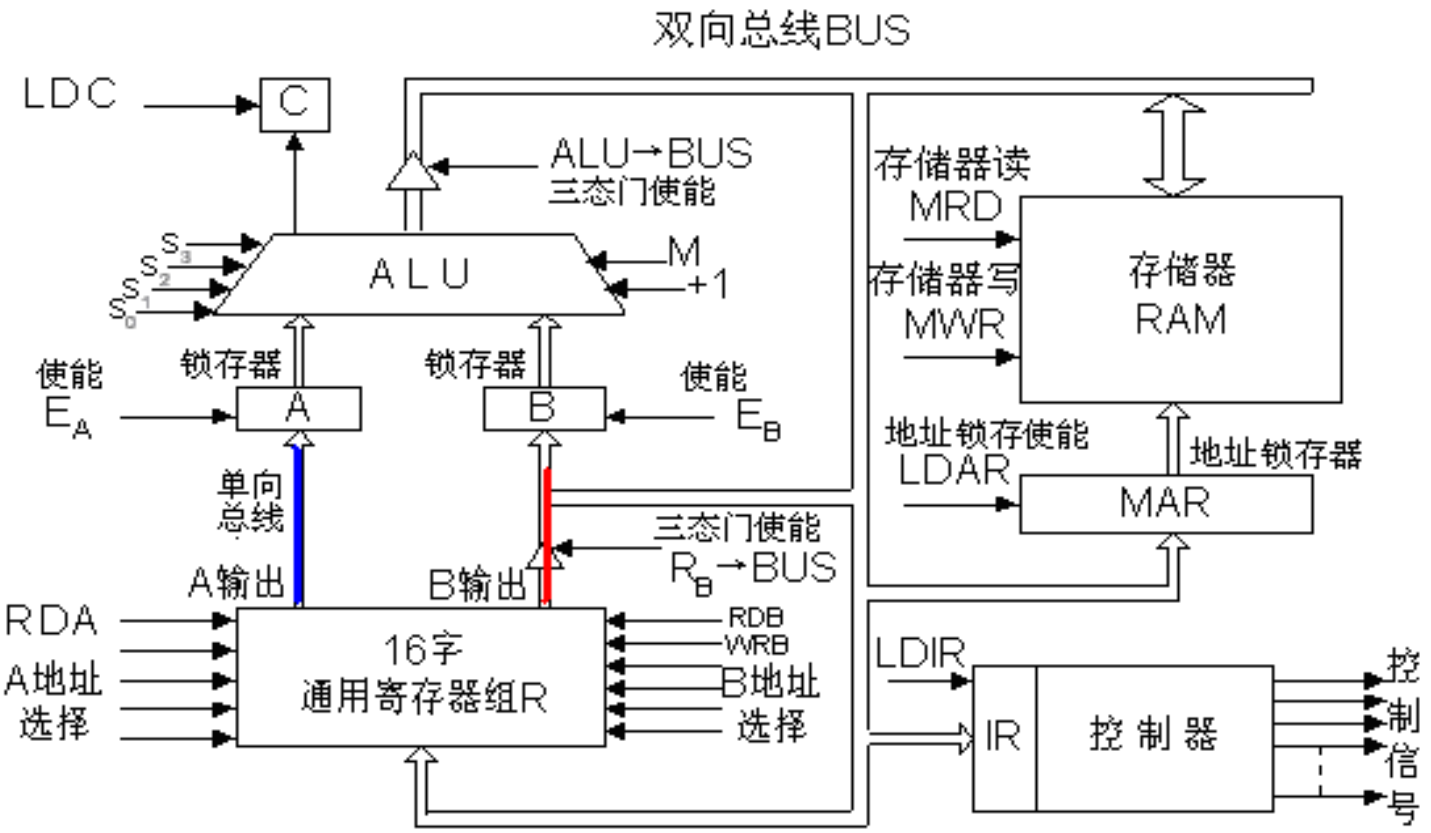
若总线较少，各子系统（信号源、目标部件）必须分别通过相同的路径传输，数据传输速度下降。

若总线增加，多个数据可以通过不同的路径同时传输，速度加快，但总线间的数据交互控制复杂。



ALU的两个输入分两次从通用寄存器输出，保存在锁存器A、B中同时处理。

通用寄存器改成双输出，增加一条通用寄存器到锁存器A的单向数据总线（例1）。数字系统中有一个ALU，一个存储器、一个16字的寄存器堆，两个锁存器A、B和一个控制器。

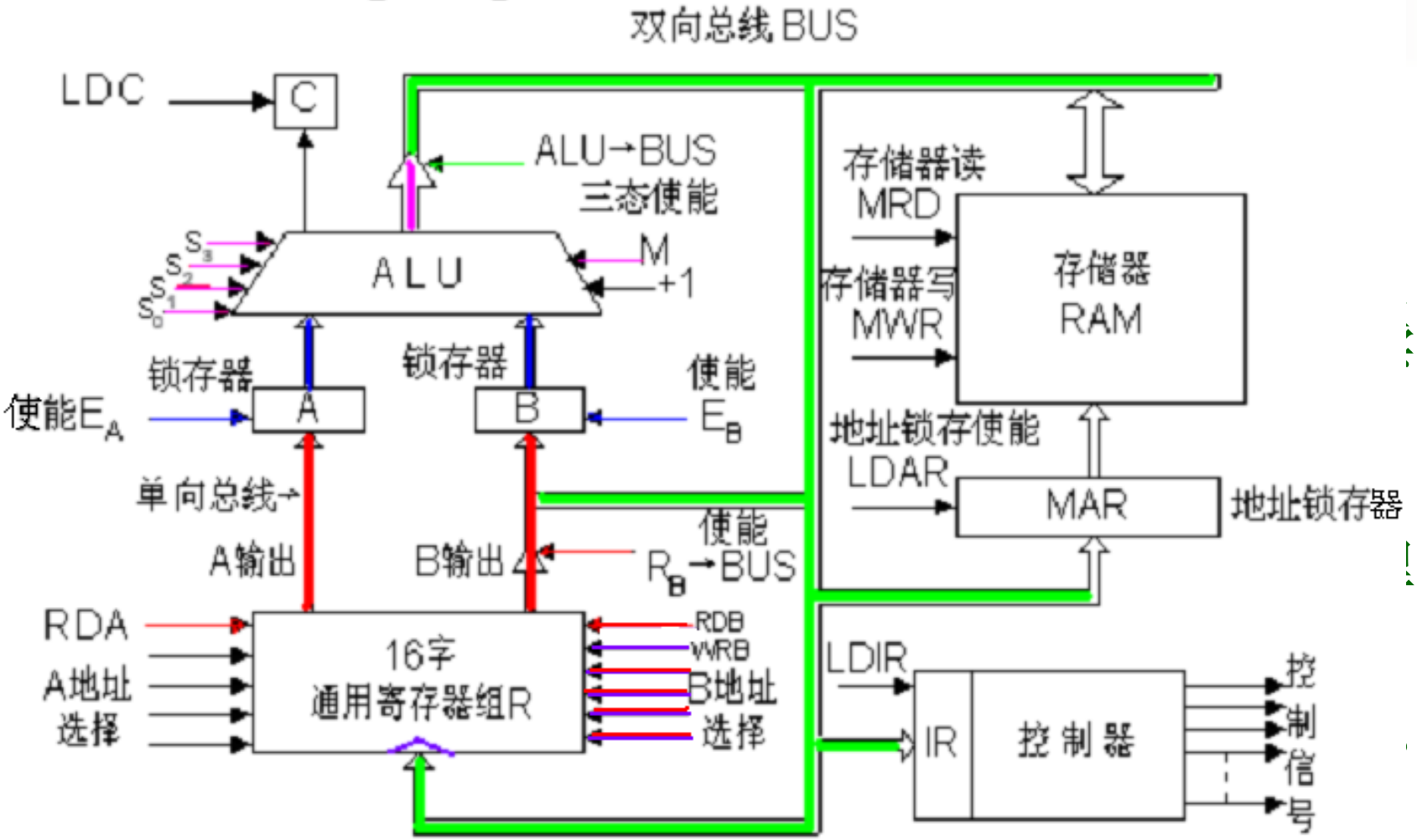


1、ALU的输出，三态门控制信号ALU->BUS；
2、存储器的双向数据端口（三态），读出控制MRD，写入控制MWR；
3、存储器的地址锁存器输入端口，锁存控制LDAR；
4、控制器的指令锁存器，锁存控制LDIR；
5、锁存器B的输入端口，接收控制E_B；
6、双端口寄存器堆B端口输出（Rj），读出控制RDB和三态门控制 R_B->BUS，寄存器数据写入控制WRB。

寄存器堆的16个字的信息通过A 端口（Ri）的单向总线传输给锁存器A，输出控制RDA，锁存器A 的接收控制E_A。

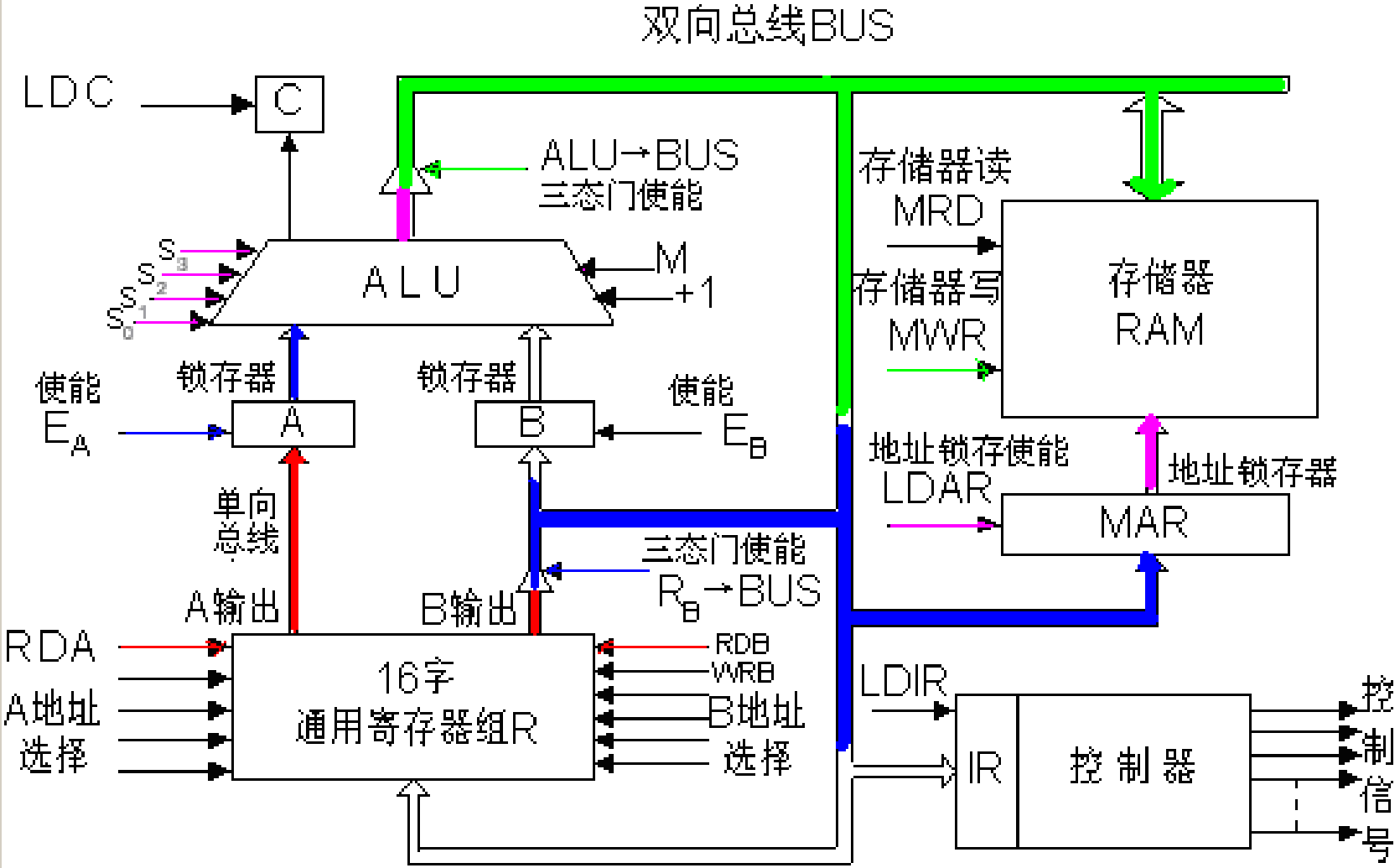
锁存器A、B的输出直接连到ALU的两组输入，不需控制信号。

Ri+Rj=>Rj



1 → 2 → 3 → 4 → 5

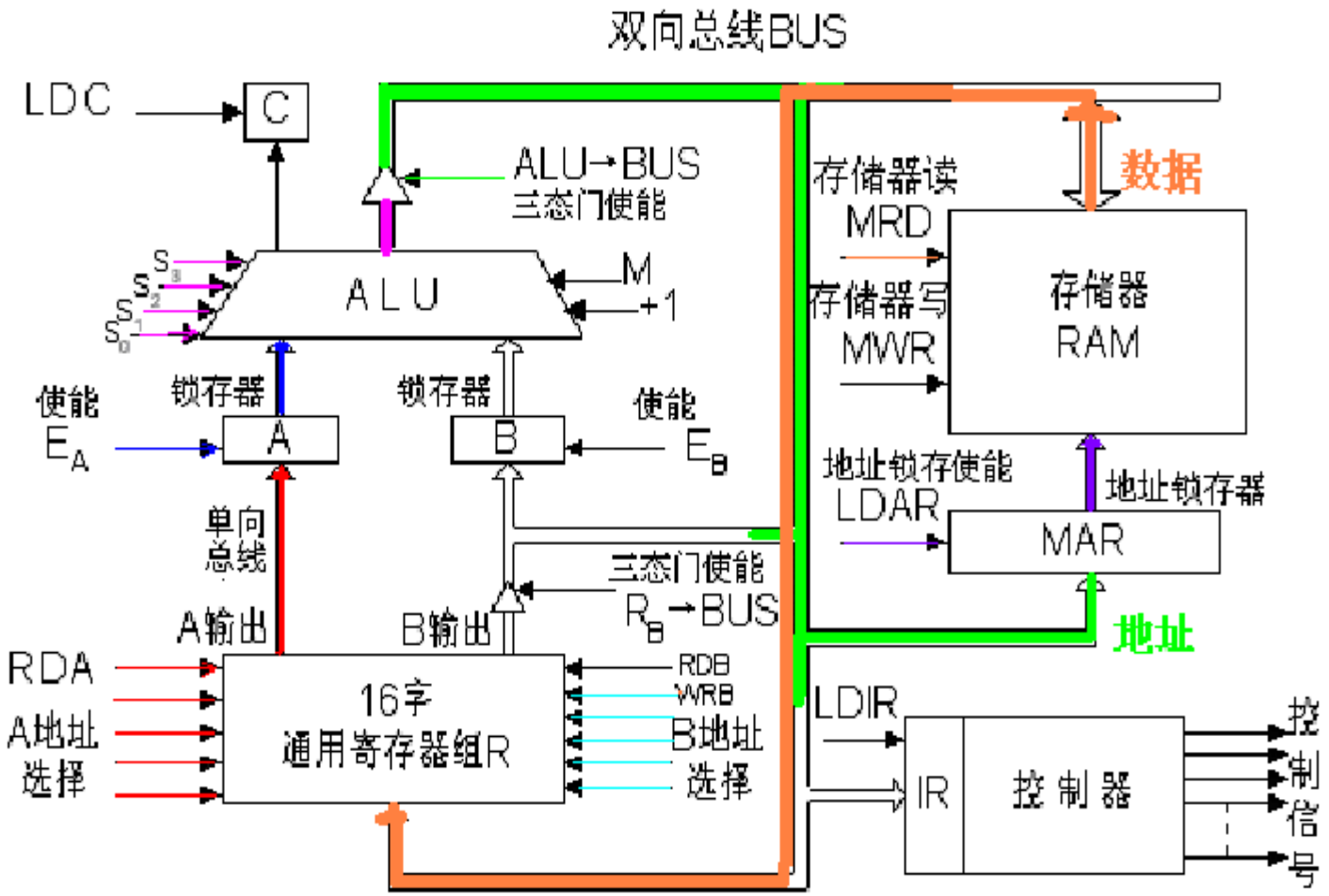
Ri=>RAM(Rj)



1 → 2 → 3 → 4

内容）选择的存储器RAM单元中。

RAM (Ri)=> Rj



1 → 2 → 3 → 4 → 5 → 6 → 7

择的

据

选择

选择

4、存

（1
单元

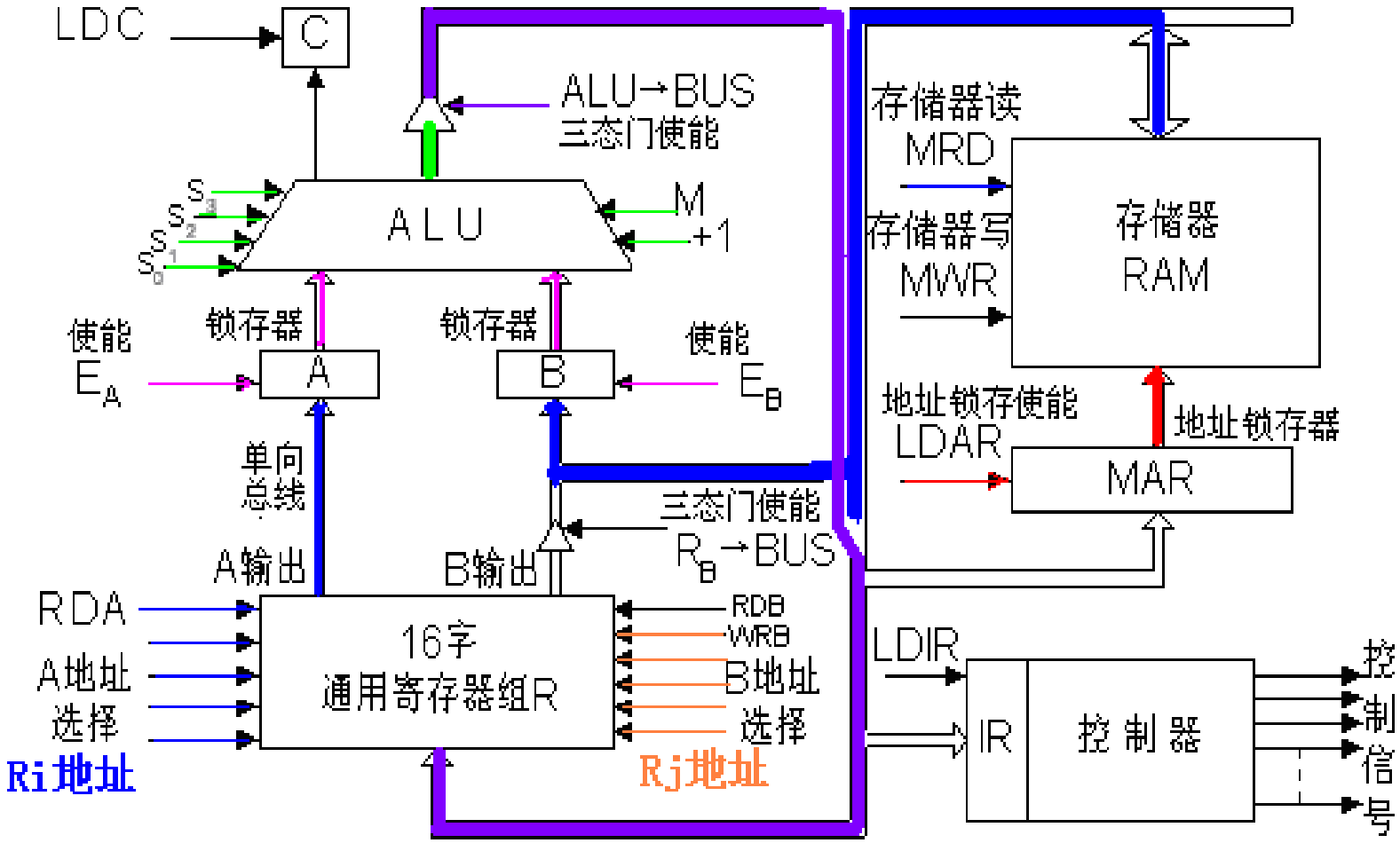
（2
据并
出R

（3
锁存

（4
果S

（5

(Ri)+(RAM)=>Rj　双向总线BUS



1 → 2 → 3 → 4 → 5 → 6

（6）B地址选择Rj，WRB有效，双向总线上的数据S被写入Rj。

6.4由顶向下的设计方法

由顶向下的设计方法：分析系统设计要求，划分任务模块，设计实现各项任务的子系统电路。

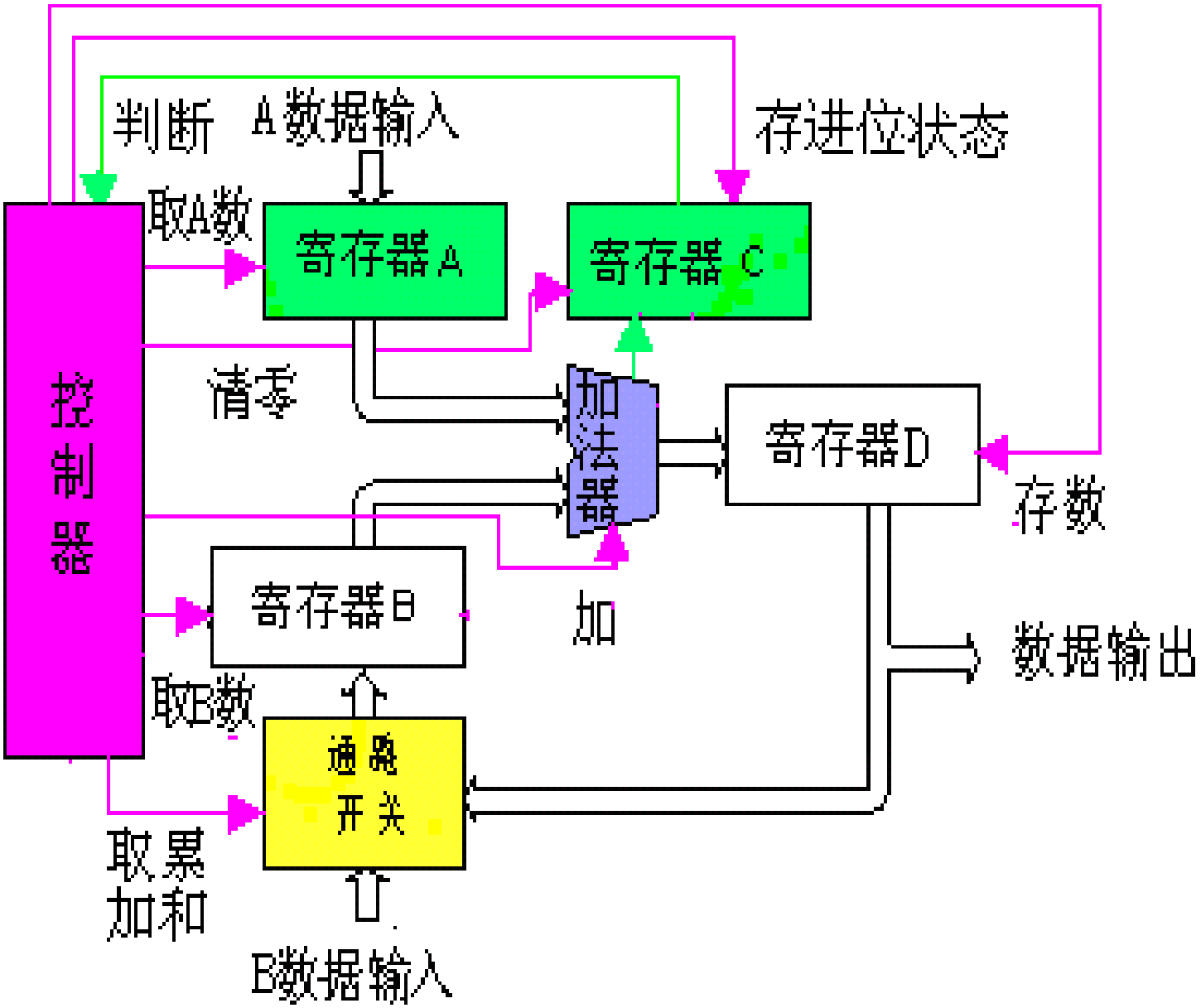
6.4.1数字系统的设计任务

1、根据设计要求划分子系统，确定各子系统需要完成的任务以及数据传输方式，确定各子系统及数据传输需要的控制命令及控制时序。

2、根据各任务设计子系统的逻辑结构。

3、根据数据传输方式设计数据传输通路。

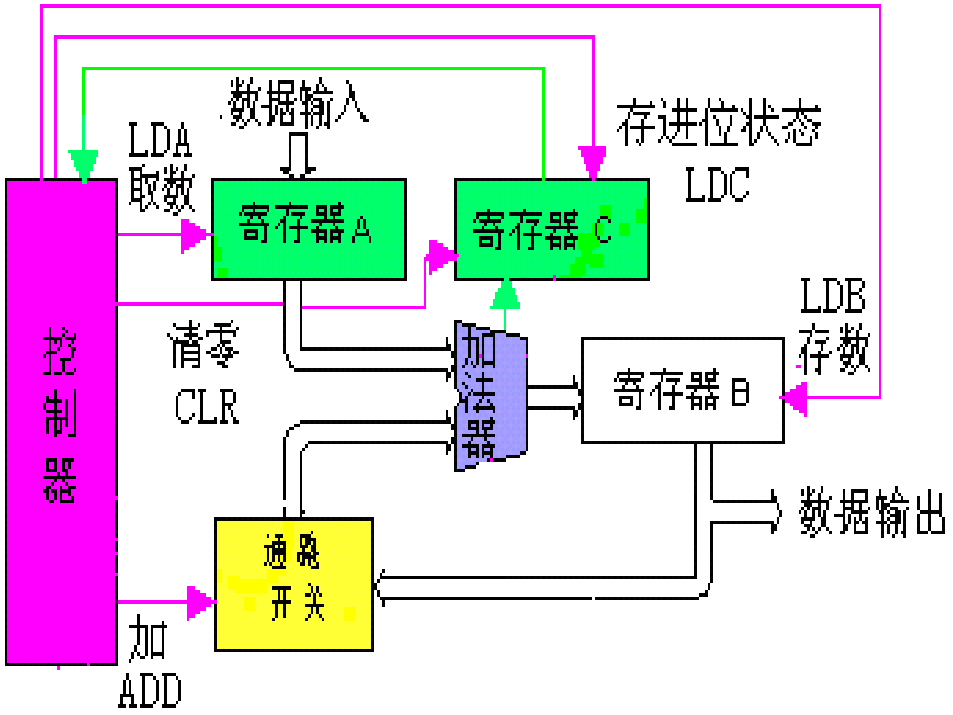
4、根据控制要求和控制时序设计控制器。


判断
A数据输入
存进位状态
取A数
寄存器A
寄存器 C
控制器
清零
加法器
寄存器D
存数
寄存器B
加
数据输出
取B数
通路开关
取累加和
B数据输入

实现方法二：

由于除第一次运算外B寄存器都是由寄存器D输入累加和，所以可以考虑将B、D寄存器合并，累加和输出到加法器输入的数据通路采用通路开关控制，由加命令控制。

第一次B寄存器的加数由A寄存器通过加法器输入。



控制命令有5条。

CLR 、LDA、LDB、LDC、ADD

第一次运算的控制顺序：

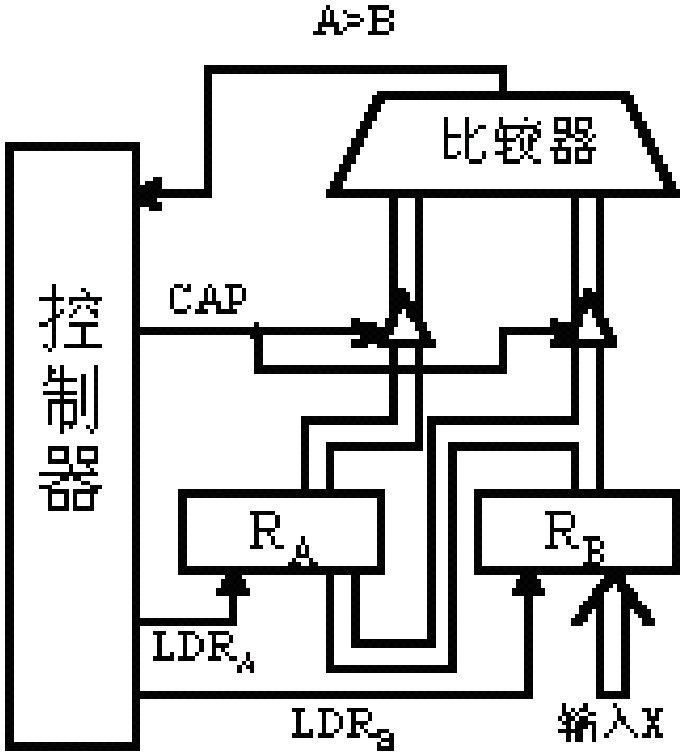
1、寄存器C清零，同时取加数B存入A寄存器。2、加数B通过加法器存入B寄存器。3、取加数A存入A寄存器输入加法器，同时切换通路开关将B数也输入到加法器。4、进位状态存C寄存器、累加和存B寄存器

要求：连续对多个二进制数比较，并根据比较结果把大数存入寄存器A保留，与下一个输入的数再次比较。

例1.2 二进制数比较系统

分析：

电路需要一个数字比较器，两个寄存器。其中寄存器A存放比较对象中大的数，寄存器B中存放新输入的数X。控制器根据比较结果状态标志 A>B决定是否将B存入A寄存器,以保留大数。



数据通路：

寄存器B的输出通过三态门切换到寄存器A的输入。

3个控制命令：

比较命令CAP

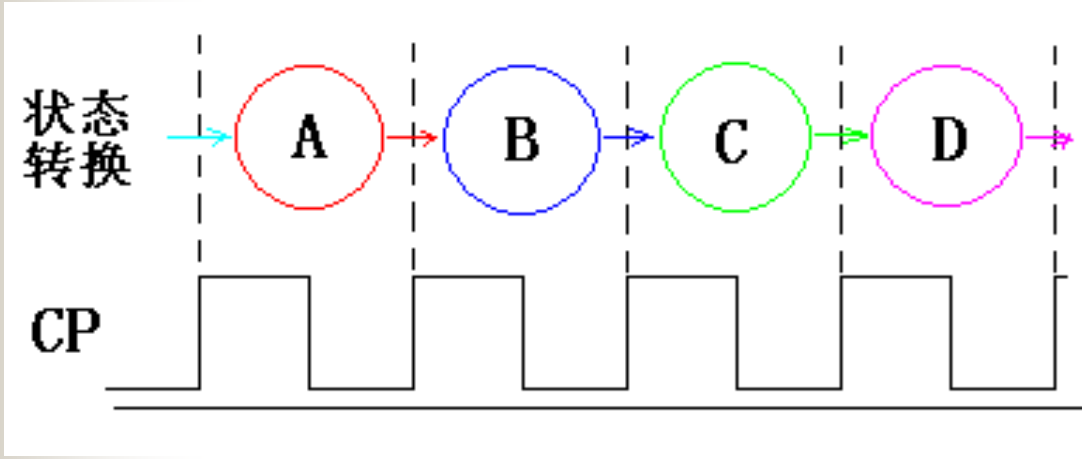
A、B寄存器存数命令LDA、LDB。

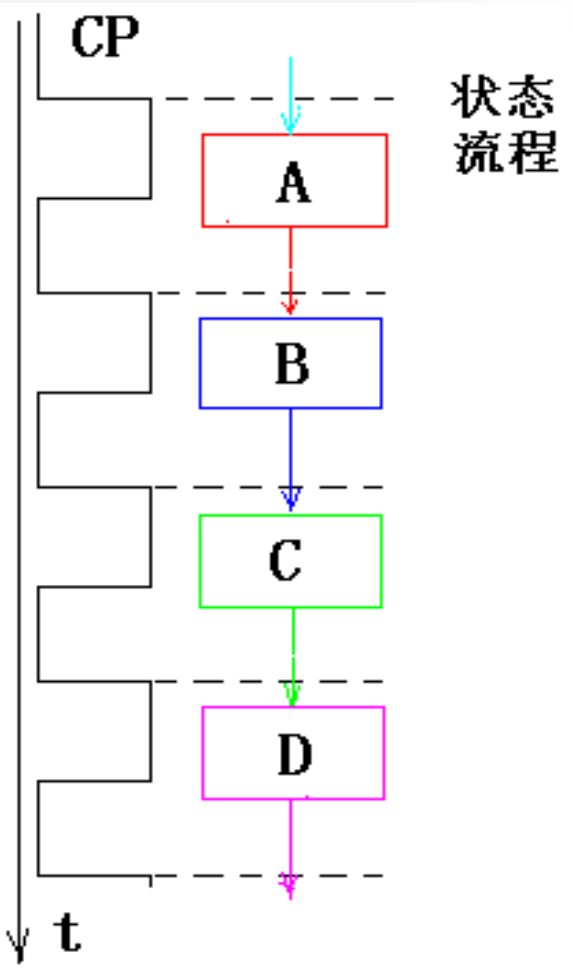
6.4.2算法状态机和算法流程图

控制算法——数字系统的控制功能描述，
求及其顺序，是设计控制器的依据。

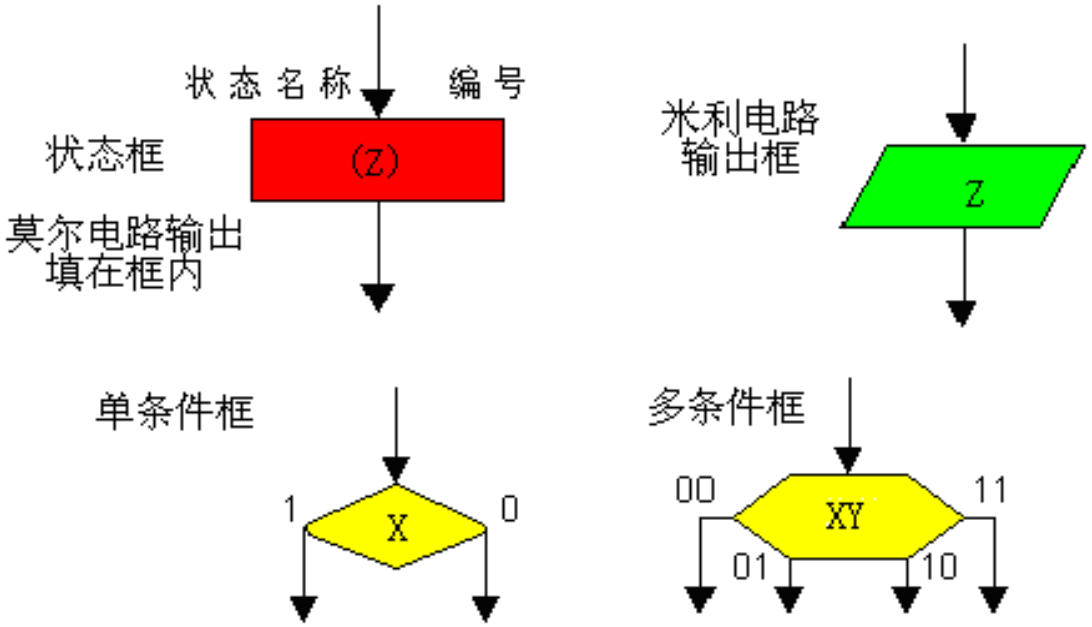
算法状态机（ASM）（ Arithmetic State
控制器的逻辑功能描述，控制状态周期性

算法流程（ASM流程）——反映状态转换
（与状态图对应），与软件流程相似。



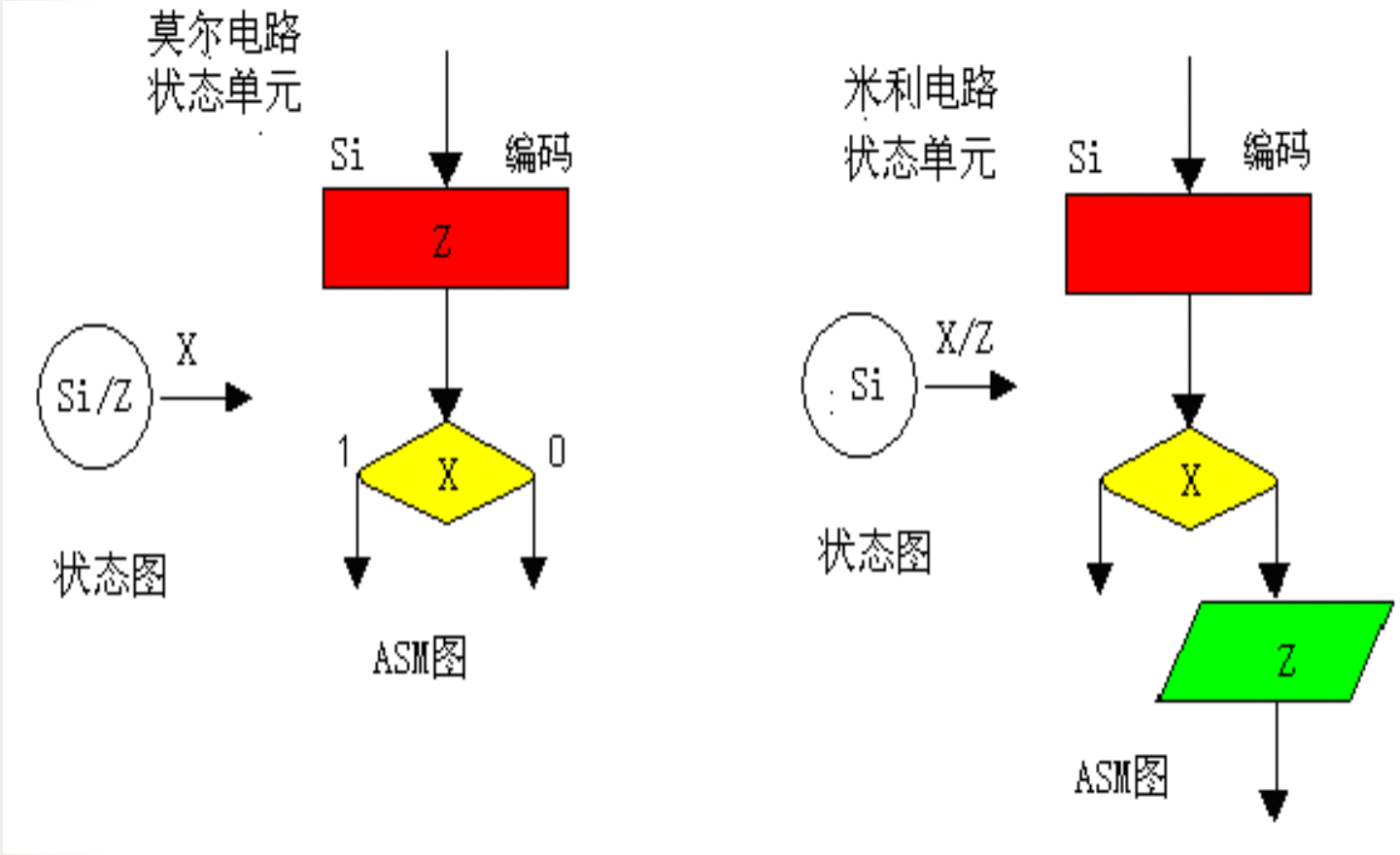


算法流程的基本图形



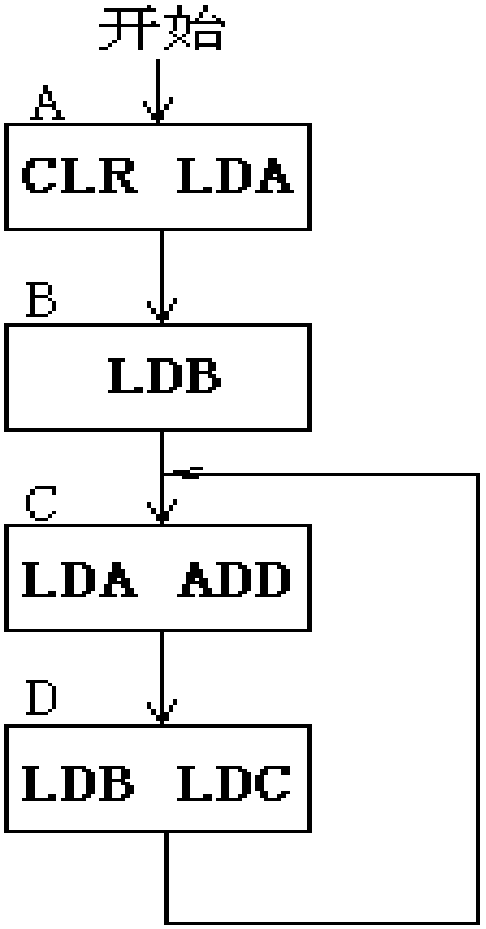
ASM状态单元（与状态图中的一个状态对应）：

状态单元的入口只有一个，总是指向状态框，出口根据控制条件确定，n个控制条件有2^n个出口，也指向状态框。



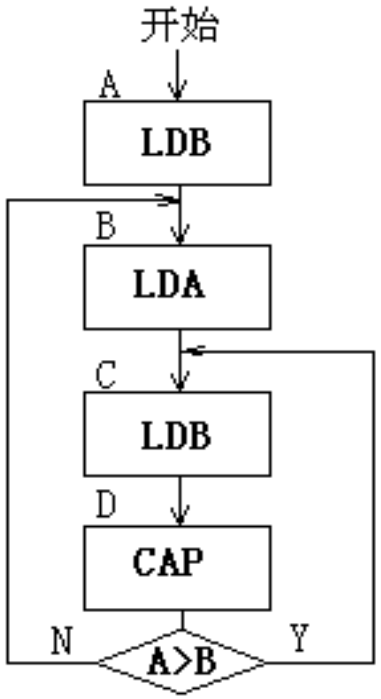
例：关于图中系统（例2）的控制流程

- 解：由系统模块划分图可知，要求控制器的指令顺序为：
- 1、CLR、LDA （状态A）
- 寄存器C清零、取加数B存入A寄存器
- 2、LDB （状态B）
- B加数存入B寄存器；
- 3、LDA、ADD（状态C）
- 取加数A、通路开关切换，
- 加数B输入加法器，A、B累加；
- 4、LDB、LDC（状态D）
- 累加和存入B寄存器，
- 进位状态存入C寄存器。
- 转回状态C，继续与新输入的A数累加
- 采用莫尔型电路实现，
- 各指令由状态直接产生。



例：数字比较系统（例10）的ASM流程

- 解：由系统模块划分图可知，要求控制器的指令顺序为：
- 1、LDB（状态A）
- 取数A存入寄存器B。
- 2、LDA （状态B）
- 寄存器B中的数存入寄存器 A。
- 3、LDB（状态C）
- 取数B存入寄存器B 。
- 4、CAP（状态D）
- 通路开关切换，数A、B比较：
- 若A>B，转回状态A，
- 将寄存器B的数存入A。
- 否则返回状态C，取新输入的数继续比较。
- 采用莫尔型电路实现，各指令由状态直接产生。



6.5.1控制器的基本概念

1、控制器的功能：根据输入信号按预定算法流程转换状态，发出指令控制各执行部件有序工作。

2、小型控制器的结构：

由时序逻辑电路中的计数器构成，状态周期性循环。

（1）输出信号的产生

莫尔型输出直接根据对应的状态码译码产生；

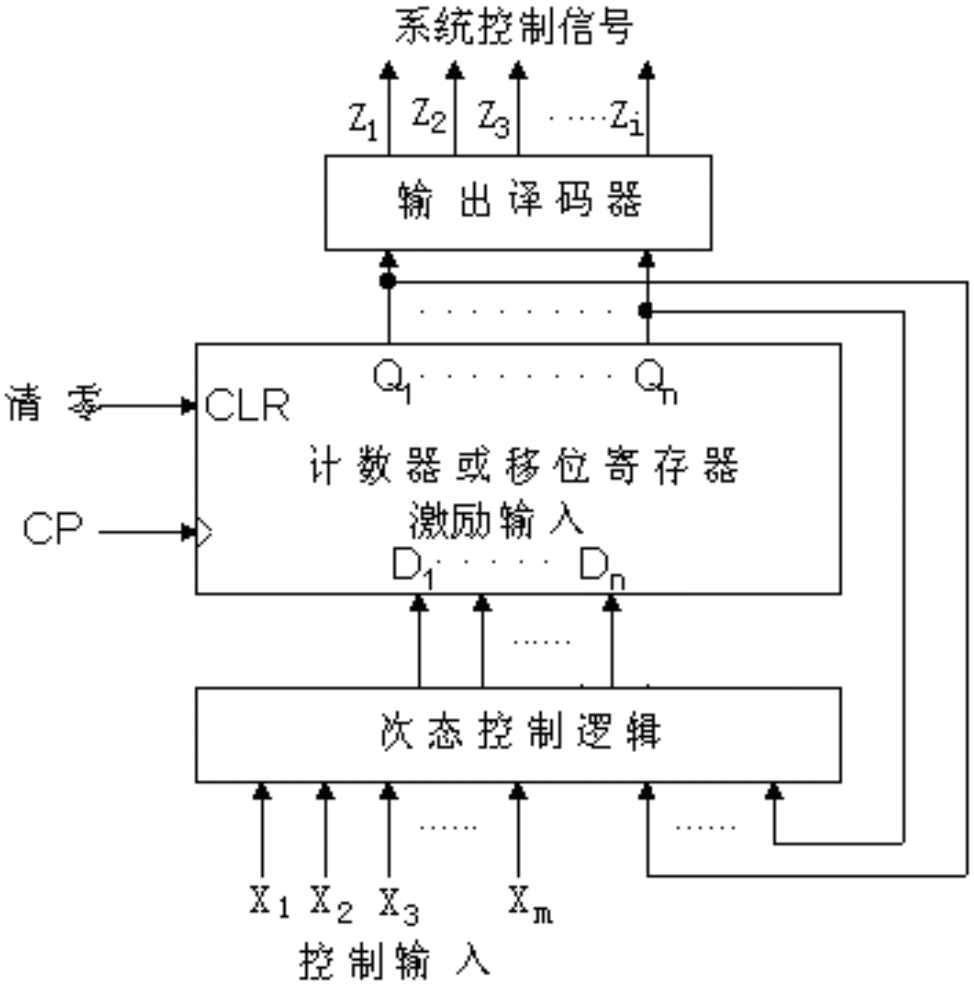
米利型输出由输入信号和状态码共同产生。

（2）状态编码形式

可以是计数码、移存码或一对一编码。

3、 小型控制器的设计方法：

与时序逻辑电路的设计步骤相同，主要求解计数器的次态控制逻辑（组合电路部分），电路实现可以是逻辑门电路或数据选择器。

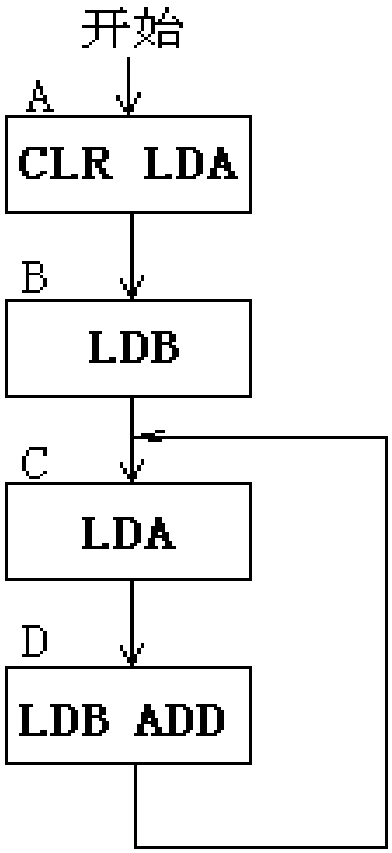


计数器型控制器结构，改动步骤烦琐。

要求例７：设计并行累加系统的计

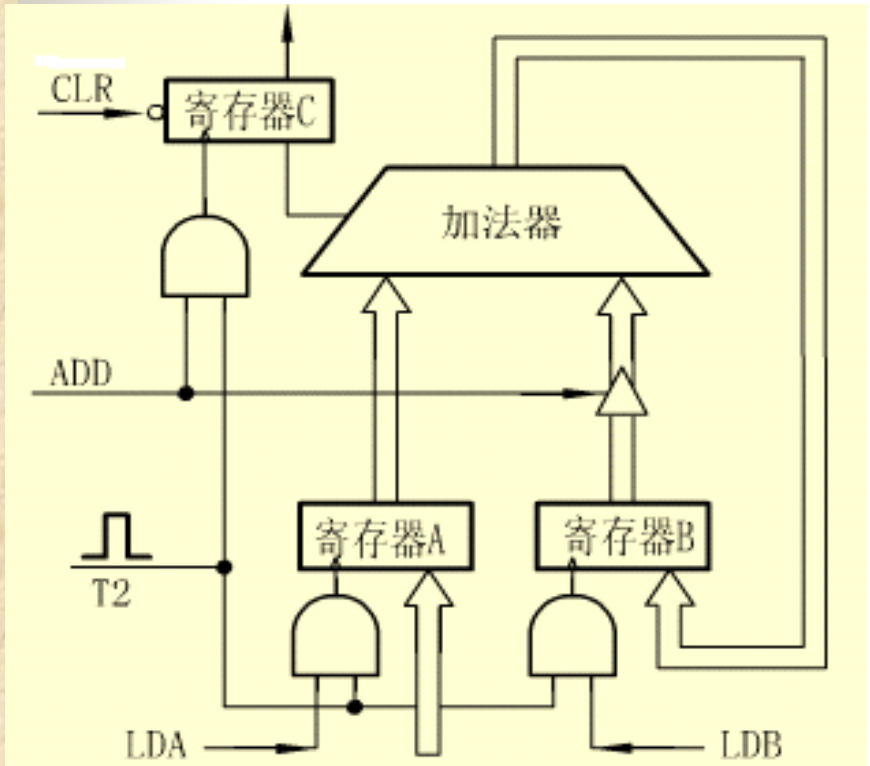
每个状态周期T分为T1、T2两拍，状态转换发生在T1时刻，电位控制信号CLR和ADD与状态同步变化，寄存器A、B的存数脉冲信号LDA、LDB在T2时刻有效。

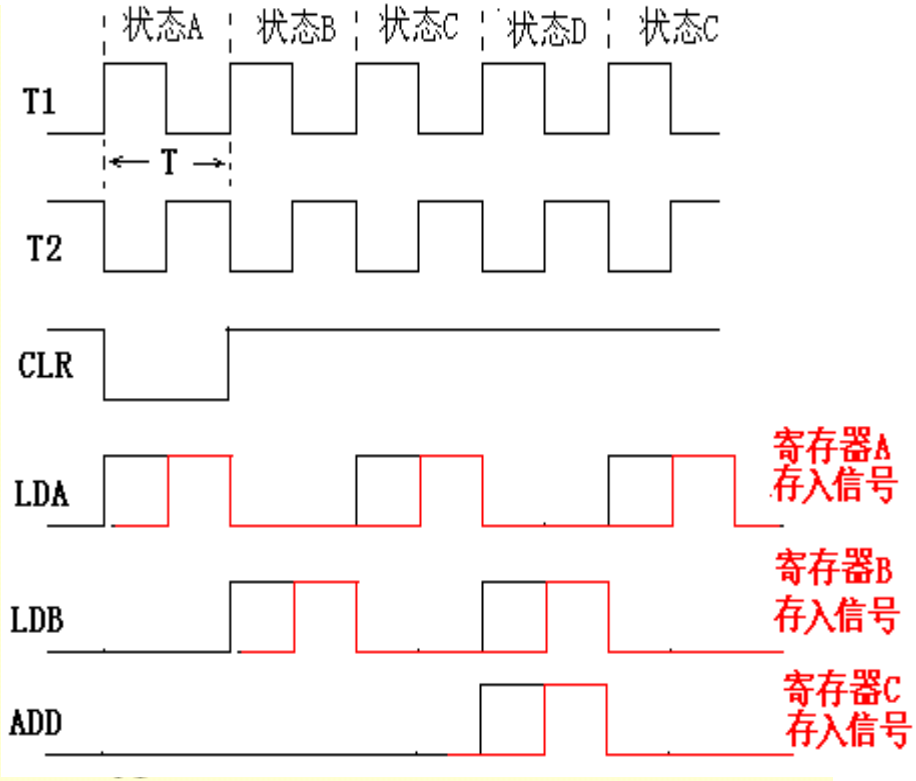
取消指令LDC，由ADD控制T2时钟代替。（ADD产生加操作，然后T2将进位存入寄存器C）。

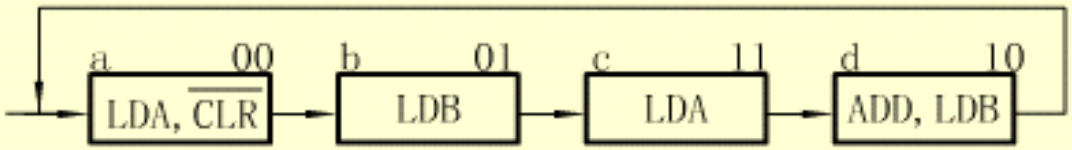


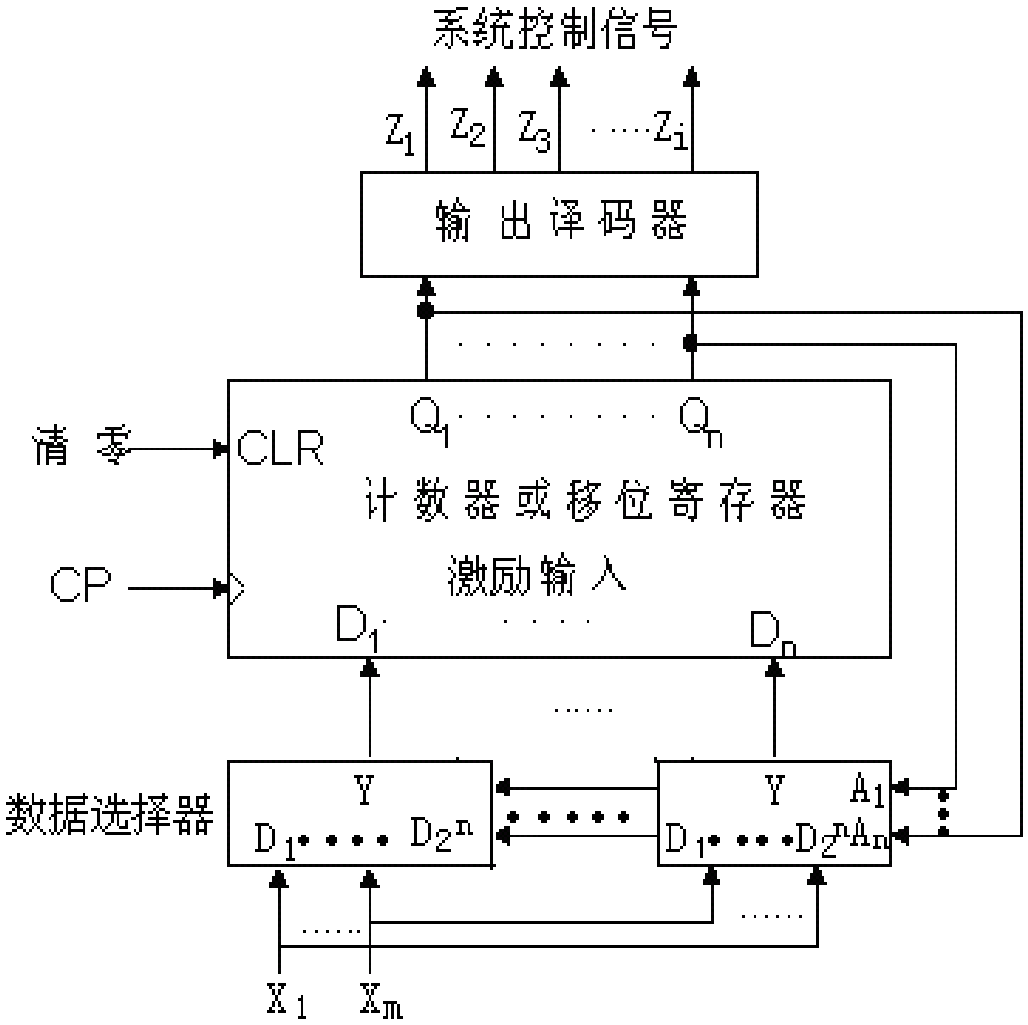
触发器的激励关系： $B(D)=B+A$ ， $A(D)=\overline{AB}$

指令输出译码逻辑：$CLR=B+A$，$LDA=A\overline{B}$，$LDB=A\oplus B$，$ADD=B\overline{A}$








系统控制信号
Z1 Z2 Z3 ……Zi
输 出 译 码 器
Q1 ……Qn
清 零
CLR
计 数 器 或 移 位 寄 存 器
激励输入
CP
D1 ……Dn
数据选择器
Y
D1 …… D2n
Y A1
D1 …… D2nAn
X1 Xm
控 制 输 入

多路器型控制器结构

6.6

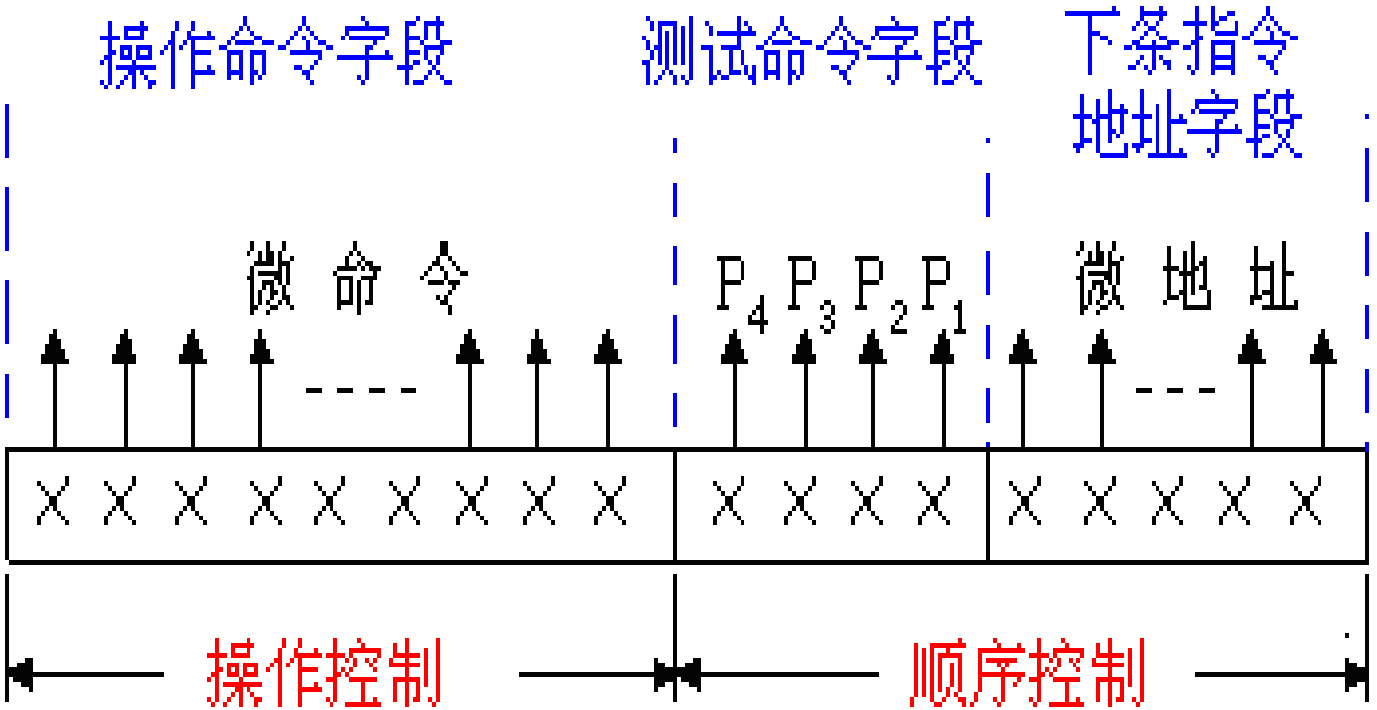
6.6.1

将控制

序

要

作



1、每

），

对

2、微

每个控制信号占据命令字段的一位

操作命令——控制执行部件工作，如取数、存数、比较等

测试命令——根据状态标志修改微地址，形成下条执行的指令地址，实现数字系统的顺序控制。

指令地址段的位数（地址码位数）决定了指令存储器的字数。

微指令的代码位数决定了存储器的字数。

解：

3个拍

1、控

2、三

编成6

3、两

4、每

T1

T2

T3

T4读

控制存储器ROM

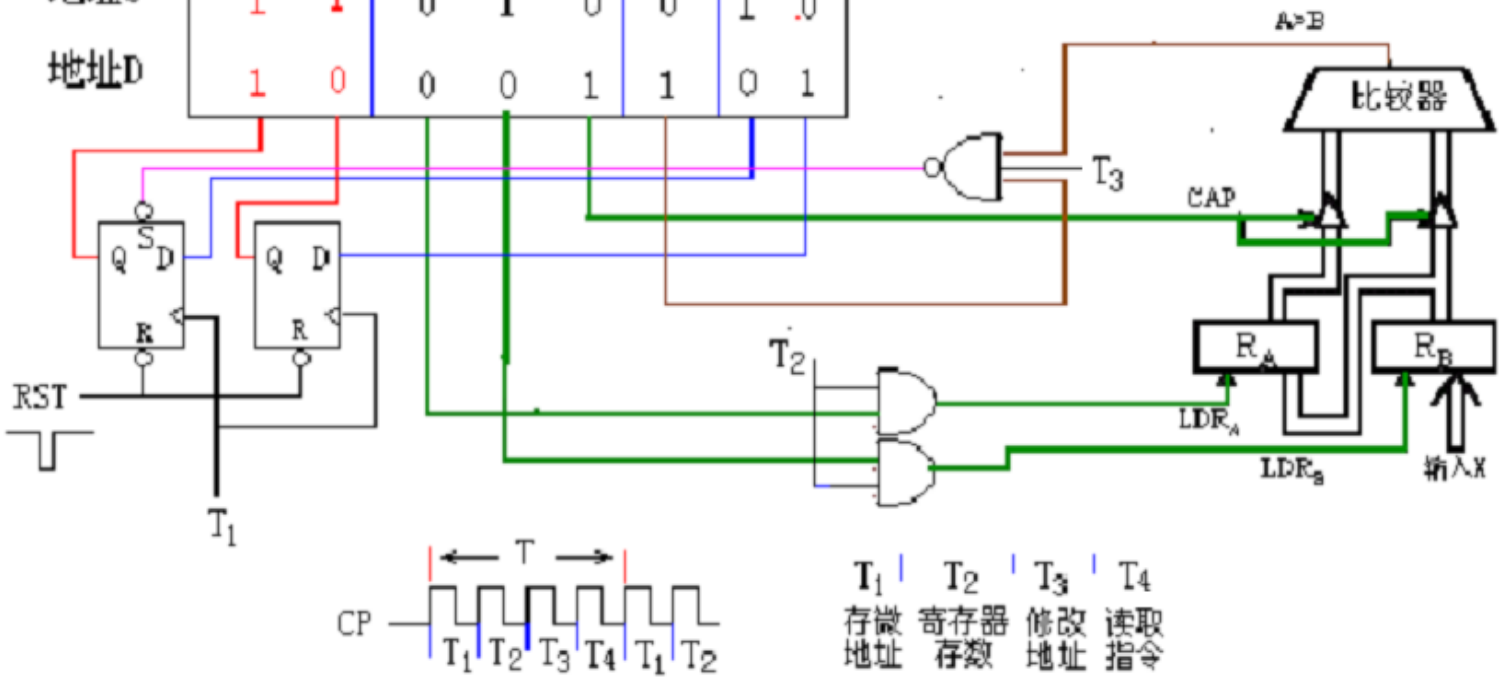
T4	RD	操作命令			测试	
	存储地址	LDA	LDB	CAP	P	微地址
	A_1 A_0	D_5	D_4	D_3	D_2	D_1D_0
初始地址	0 0	0	1	0	0	0 1
地址B	0 1	1	0	0	0	1 1
地址C	1 1	0	1	0	0	1 0
地址D	1 0	0	0	1	1	0 1

工作原理：

1、开始工作时RST=0，使ROM的地址寄存器清零，输出控制信号LDB=1，下条指令地址为“01”。

2、CP的上升沿后，地址寄存器为“01”，输出控制信号LDA=1，转“11”单元。然后再输出LDB=1，转“10”单元，输出CAP=1，P=1。

3、若A>B=1，与非门输出为0，使控制A1的D触发器置位，转向“11”单元。否则转“01”单元。



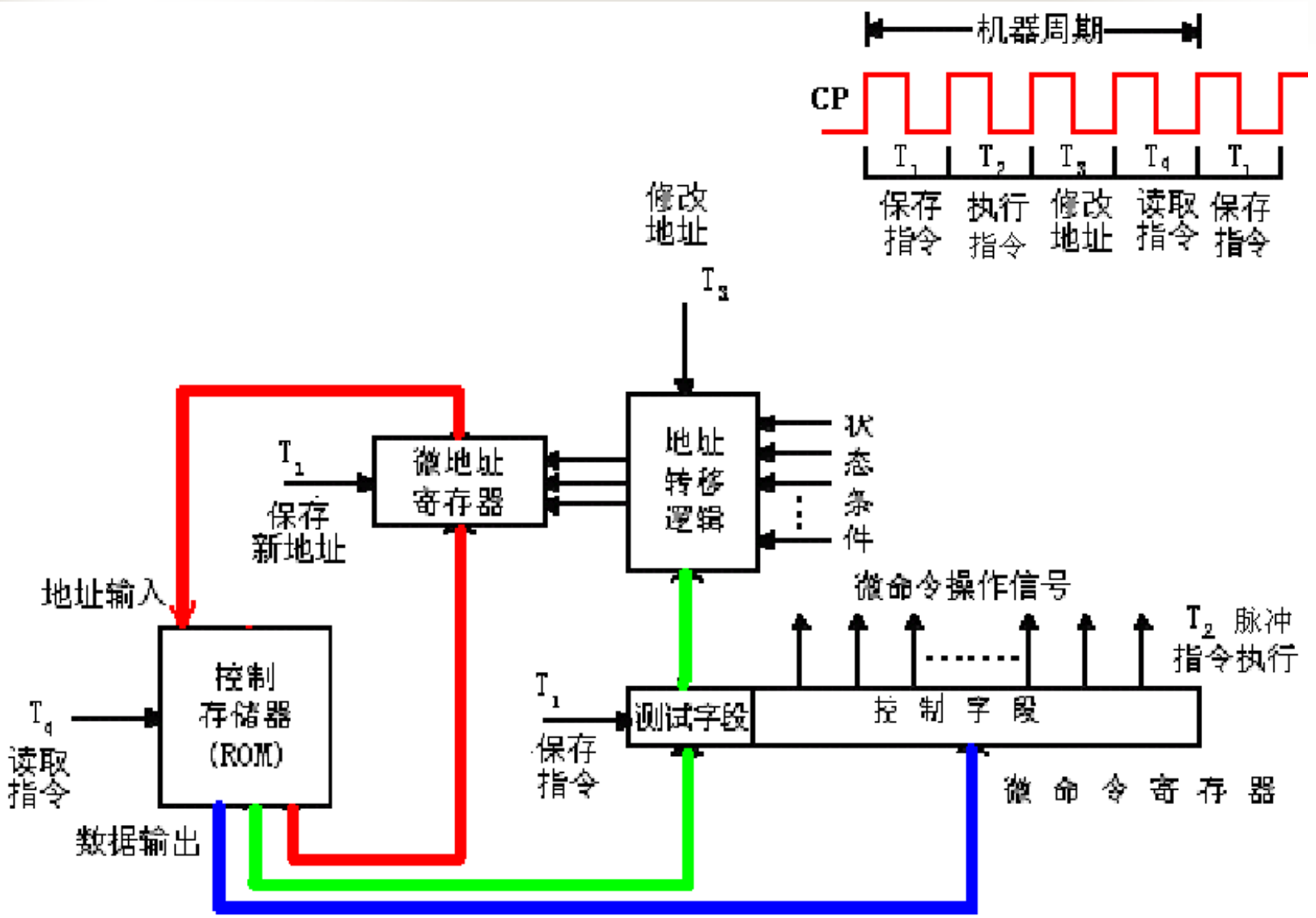
1、

2、

3、

4、

每条
可以


机器周期
CP
T1
T2
T3
T4
T1
保存指令
执行指令
修改地址
读取指令
保存指令
修改地址
T3
T1
保存新地址
微地址寄存器
地址转移逻辑
状态条件
地址输入
控制存储器(ROM)
T4
读取指令
数据输出
微命令操作信号
T2 脉冲
指令执行
T1
保存指令
测试字段
控制字段
微命令寄存器

任务：6.6.3微程序控制器的设计

1、系统分析——根据系统模块划分建立控制算法。

2、软件编制——根据控制算法确定控制器需要输入的控制信号、测试信号和需要输出的操作信号数。并根据各信号的时序关系确定系统的控制周期数（状态数）。

将系统各种具体操作所需的所有控制信号和测试信号组编成命令码，与各次操作的下条指令地址构成二进制代码形式的微程序。

3、硬件设计——根据微指令代码的位数确定控制存储器、微命令寄存器、微地址寄存器的位数；根据系统总控制周期数确定控制存储器的字数。根据各任务的执行顺序设计指令地址转移逻辑。根据指令执行中的部件操作时序确定控制周期中的时钟拍数。

例12：数据通路

10个执行部件：

ALU（74181）、RAM、寄存器堆、2个锁存器A、B
2个寄存器C、MAR、2个三态门A_S、B_S、1个任务计数器IR。

26个控制信号

BUS_A , S_3

S_2 , S_2 , S_0

M , +1 , E_A

E_B , BUS_B , RD_A

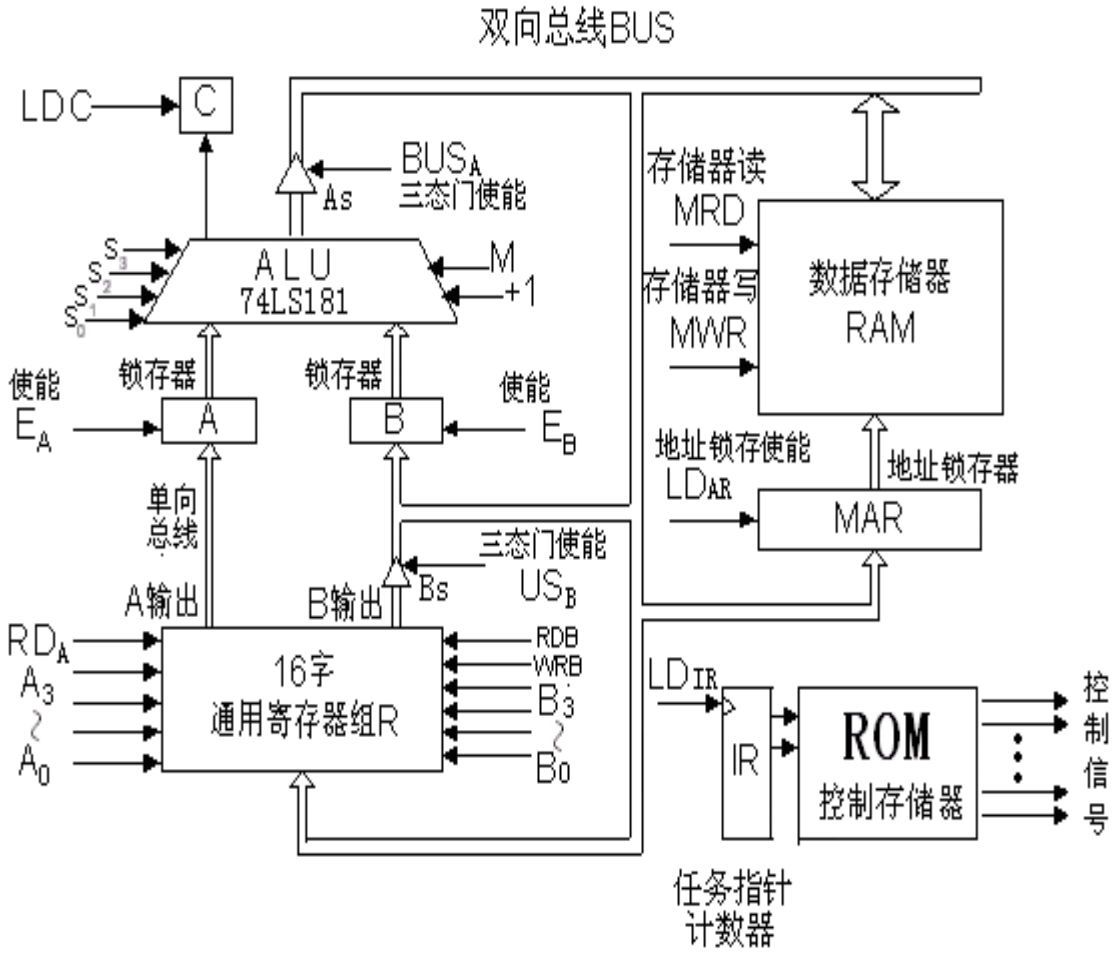
A_3 , A_2 , A_1

A_0 , RD_B , B_3

B_2 , B_1 , B_0

WR_B , MRD , MWR

LD_{AR} , LD_{IR} , LD_C



例14：

设计数据通路的微控制器。

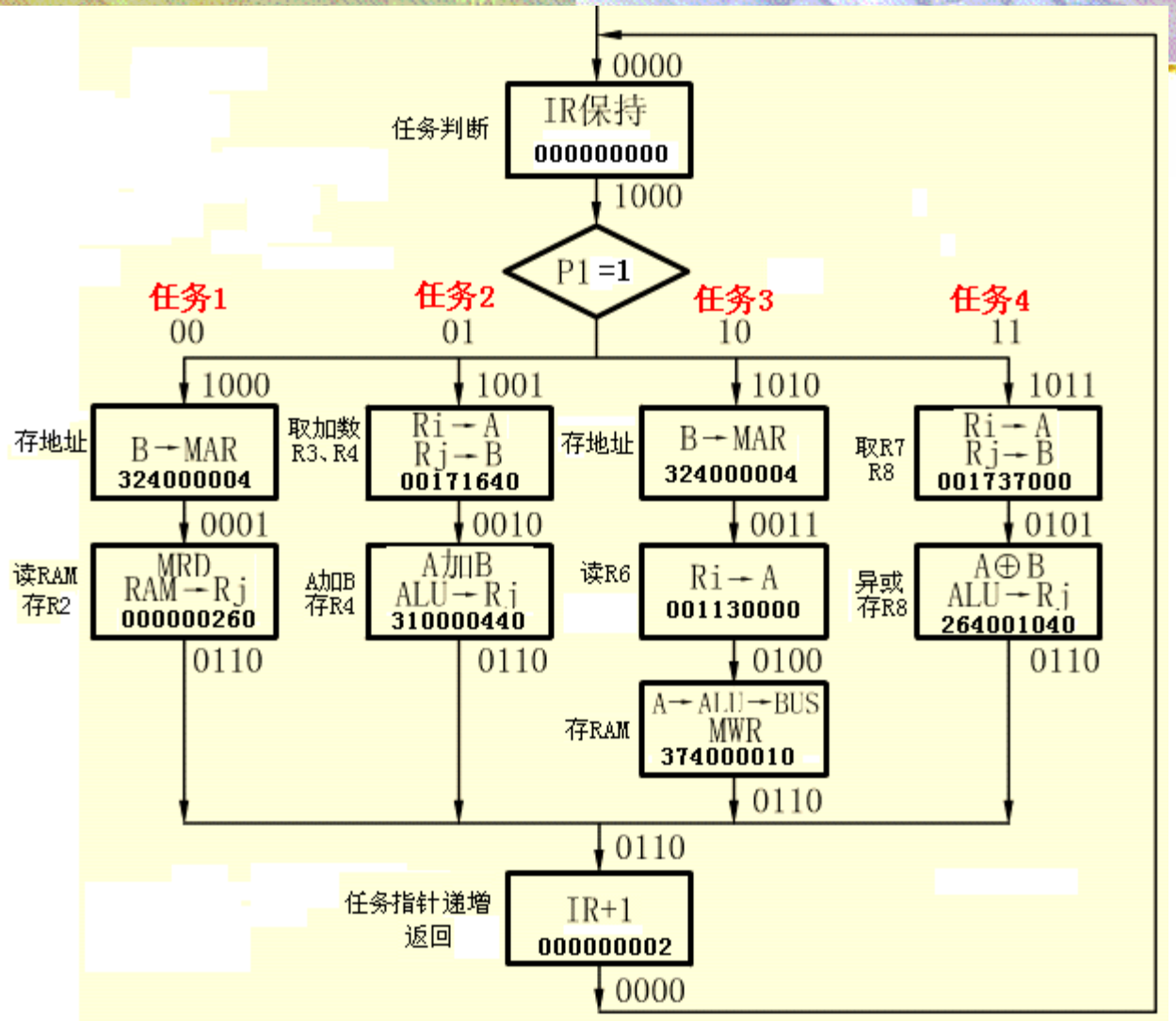
1、系统在计数器IR的控制下循环顺序执行4个任务。

2、初始状态随机，每个任务完成后输出LD_{IR}指令使计数器加1。

3、读写RAM的地址都由B寄存器给出。

控制流程如右图：

完成四个任务所需的指令分别是3、3、4、3条，其中有1条公共指令“任务指针递增后返回”。所以共需要11个存储单元存放11条指令（其中一条任务判断指令）。



控制存储器的字数取16，地址码4位；数据位为32位：其中26位控制码、2位测试码、4位微地址码。

数据通路控制存储器的ROM内容表

存储地址	存　　储　　数　　据		
	操　作　命　令	测试	微 地 址
$A_3A_2A_1A_0$	D_{31},D_{30} , D_{29}~D_{27},D_{26}~D_{24},D_{23}~D_{21} , D_{20}~D_{18},D_{17}~D_{15}, D_{14}~D_{12} , D_{11}~D_9,D_8~D_6	D_5D_4	D_3D_2 D_1D_0
	X,Bus_A,S_3, S_2, S_1, S_0, M,+1,E_A, E_B, Bus_B,RD_A, A_3, A_2, A_1, A_0, RD_B, B_3, B_2, B_1, B_0, WR_B, MRD, MWR, LD_{AR}, LD_{IR}, LD_C	P_2P_1	$A_3A_2A_1A_0)_{nex}$
0 0 0 0	八进制码　0 0 0 0 0 0 0 0 0 0	0 1	1 0 0 0+(IR)
0 0 0 1	八进制码　0 0 0 0 0 0 2 6 0	0 0	0 1 1 0
0 0 1 0	八进制码　3 1 0 0 0 0 4 4 0	0 0	0 1 1 0
0 0 1 1	八进制码　0 0 1 1 3 0 0 0 0	0 0	0 1 0 0
0 1 0 0	八进制码　3 7 4 0 0 0 0 1 0	0 0	0 1 1 0
0 1 0 1	八进制码　2 6 4 0 0 1 0 4 0	0 0	0 1 1 0
0 1 1 0	八进制码　0 0 0 0 0 0 0 0 0 2	0 0	0 0 0 0
0 1 1 1			
1 0 0 0	任务指针: IR=00　八进制码　3 2 4 0 0 0 0 0 4	0 0	0 0 0 1
1 0 0 1	IR=01　八进制码　3 2 4 0 0 0 0 0 4	0 0	0 0 1 0
1 0 1 0	IR=10　八进制码　3 2 4 0 0 0 0 0 4	0 0	0 0 1 1
1 0 1 1	IR=11　八进制码　0 0 1 7 3 7 0 0 0	0 0	0 1 0 1

任务1(IR指针00)：　(RAM)→R2

以B寄存器内容为地址的RAM单元内容存入寄存器堆的R2（3条指令）

X,Bus_A,S_3	S_2, S_1, S_0	$M,+1,E_A$	E_B, Bus_B, RD_A	A_3, A_2, A_1	A_0, RD_B, B_3	B_2, B_1, B_0	WR_B, MRD, MWR	LD_{AR}, LD_{IR}, LD_C
0, 1, 1	0, 1, 0	1, 0, 0	000	000	000	000	000	1, 0, 0
3	2	4	0	0	0	0	0	4
000	000	000	000	000	0, 0, 0	0, 1, 0	1, 1, 0	000
0	0	0	0	0	0	2	6	0
000	000	000	000	000	000	000	000	0, 1, 0
0	0	0	0	0	0	0	0	2

操作说明:

1、ALU控制码M=1，执行逻辑功能，S_3~S_0=1010，输出F=B； ALU->BUS、LDAR有效，把B寄存器内容送上总线并存入RAM的地址寄存器。

2、MRD有效，读RAM内容；B地址选择寄存器R2，WRB有效，将RAM读出的内容存入R2。

3、LDIR有效，读IR的内容准备执行下一任务。回到初始状态。

任务2(IR指针01)：(R3)+(R4)→R4

两个寄存器（R3和R4）的内容相加，并把和存入寄存器堆的R4（3条指令）

X,Bus_A,S_3	S_2, S_1, S_0	M,+1,E_A	E_B, Bus_B,RD_A	A_3, A_2, A_1	A_0, RD_B, B_3	B_2, B_1, B_0	WR_B, MRD, MWR	LD_{AR}, LD_{IR}, LD_C
000	000	0, 0, 1	1, 1, 1	0, 0, 1	1, 1, 0	1, 0 , 0	000	000
0	0	1	7	1	6	4	0	0
0, 1, 1	0, 0, 1	000	000	000	0, 0, 0	1, 0, 0	1, 0, 0	000
3	1	0	0	0	0	4	4	0
000	000	000	000	000	000	000	000	0, 1, 0
0	0	0	0	0	0	0	0	2

操作说明:

1、A地址选择R3，B地址选择R4；RDA、RDB有效，读出R3、R4内容；EA、EB、R_B->Bus有效，R3、R4的内容送ALU的输入A和B。

2、ALU的控制码M=0 ,执行算术操作, S3~S0=1001，执行A加B；

ALU->BUS有效，把加运算的和送上总线；B地址仍选择R4，WRB有效，把和存入R4。

3、LDIR有效，读IR的内容准备执行下一任务。回到初始状态。

任务3(IR指针10)：(R6)→RAM

把寄存器R6的内容存入以B寄存器 内容为地址的RAM单元内（4条指令）

X,Bus_A,S_3	S_2, S_1, S_0	$M,+1,E_A$	E_B, Bus_B,RD_A	A_3, A_2, A_1	A_0, RD_B, B_3	B_2, B_1, B_0	WR_B, MRD, MWR	LD_{AR}, LD_{IR}, LD_C
0, 1, 1	0, 1, 0	1, 0, 0	000	000	000	000	000	1, 0, 0
3	2	4	0	0	0	0	0	4
000	000	0, 0, 1	0, 0, 1	0, 1, 1	0, 0, 0	000	000	000
0	0	1	1	3	0	0	0	0
0, 1, 1	1, 1, 1	1, 0, 0	000	000	0, 0, 0	000	0, 0, 1	000
3	7	4	0	0	0	0	1	0
000	000	000	000	000	000	000	000	0, 1, 0
0	0	0	0	0	0	0	0	2

操作说明:

1 、ALU控制码M=1，执行逻辑功能，S3~S0=1010，输出F=B；
ALU->BUS、LDAR有效，把B寄存器内容送上总线并存入RAM的地址寄存器。
2、寄存器堆的A地址选择R6，RDA、EA有效，R6的内容送ALU；
3、ALU控制码M=1，执行逻辑功能，S3~S0=1111，输出F=A；
ALU->BUS、MWR有效，把R6的内容送上总线并存入RAM。
4、LDIR有效，读IR的内容准备执行下一任务。回到初始状态。

任务4(IR指针11)：(R7) ⊕ (R8)→R8

两个寄存器（R7和R8）的内容异或，并把结果存入寄存器堆的R8（3条指令）

X,Bus_A,S_3	S_2, S_1, S_0	M,+1,E_A	E_B, Bus_B,RD_A	A_3, A_2, A_1	A_0, RD_B, B_3	B_2, B_1, B_0	WR_B, MRD, MWR	LD_{AR}, LD_{IR}, LD_C
000	000	0, 0, 1	1, 1, 1	0, 1, 1	1, 1, 1	0, 0, 0	000	000
0	0	1	7	3	7	0	0	0
0, 1, 0	1, 1, 0	1, 0, 0	000	000	0, 0, 1	0, 0, 0	1, 0, 0	000
2	6	4	0	0	1	0	4	0
000	000	000	000	000	000	000	000	0, 1, 0
0	0	0	0	0	0	0	0	2

操作说明：

1、A地址选择R7，B地址选择R8；RDA、RDB有效，读出R1、R4内容；EA、EB、R_B->Bus有效，R7、R8的内容送ALU的输入A和B。

2、ALU的控制码M=1 ,执行算术操作, S3~S0=0110，执行A异或B；ALU->BUS有效，把ALU的运算结果送上总线；B地址仍选择R8，WRB有效，把运算结果存入R8。

3、LDIR有效，读IR的内容准备执行下一任务。回到初始状态。

ALU算术逻辑运算单元真值表

74181 (Alu/Function Generator)

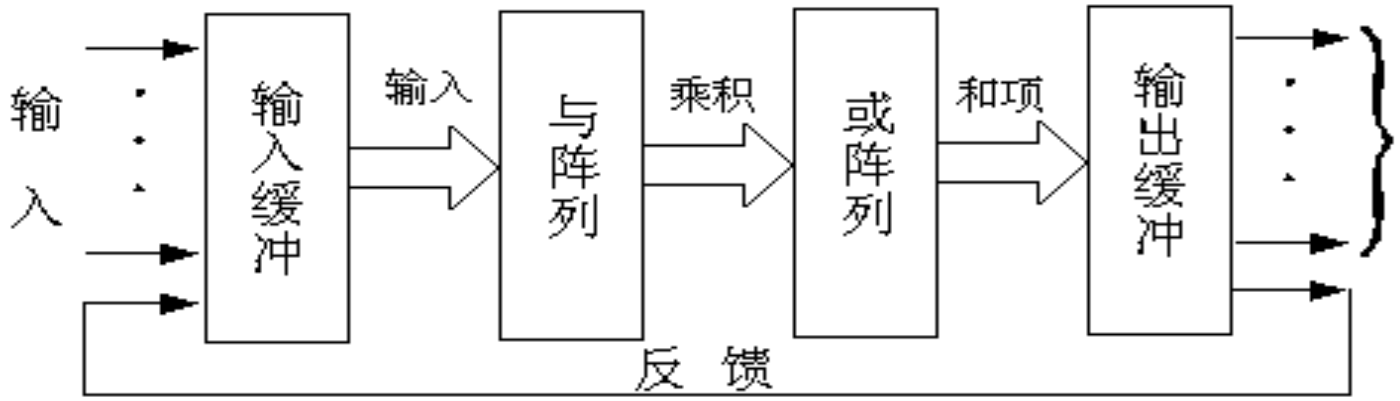
Selection S3	S2	S1	S0	M=H LOGIC FUNCTIONS	M=L; Arithmetic Operations Cn=L (no carry)	Cn=H (with carry)
0	0	0	0	F= $\overline{A}$	F=A MINUS 1	F=A
0	0	0	1	F= $\overline{AB}$	F= AB MINUS 1	F=AB
0	0	1	0	F= $\overline{A}$+B	F= $A\overline{B}$ MINUS 1	F=$A\overline{B}$
0	0	1	1	F= 1	F=MINUS 1(2's comp)	F=Zero
0	1	0	0	F= A+$\overline{B}$	F=A PLUS (A+$\overline{B}$)	F=A PLUS(A+$\overline{B}$) Plus 1
0	1	0	1	F= $\overline{B}$	F=AB PLUS(A+$\overline{B}$)	F=AB PLUS(A+$\overline{B}$)Plus 1
0	1	1	0	F= A⊕B	F=A MINUS B MINUS 1	F= A MINUS B
0	1	1	1	F= A+$\overline{B}$	F= A+$\overline{B}$	F= (A+$\overline{B}$) PLUS 1
1	0	0	0	F= $\overline{A}$B	F= A PLUS (A+B)	F=A PLUS(A+B)PLUS 1
1	0	0	1	F= A⊙B	F= A PLUS B	F=A PLUS B PLUS 1
1	0	1	0	F = B	F= $A\overline{B}$ PLUS (A+B)	F=$A\overline{B}$ PLUS (A+B)PLUS 1
1	0	1	1	F = A + B	F = (A + B)	F=(A+B)PLUS 1
1	1	0	0	F = 0	F = A PLUS A	F=A PLUS A PLUS 1
1	1	0	1	F = A$\overline{B}$	F= A$\overline{B}$ PLUS A	F=A$\overline{B}$ PLUS A PLUS 1
1	1	1	0	F=A$\overline{B}$	F=A$\overline{B}$ PLUS A	F=A$\overline{B}$ PLUS A PLUS 1
1	1	1	1	F = A	F = A	F= A PLUS 1

第七章　可编程逻辑器件 PLD

Programmable Logic Device

集成器件的逻辑功能可以由用户自行设计，修改。

7.1引言



PLD的功能框图

户、器、的用

户自行设计、可在系统现场修改的大规模、超大规模集成电路。

二、可编程器件的基本结构

由与、或阵列、（存储单元）、输入输出缓冲电路构成，可实现任何形式的组合逻辑电路（或时序逻辑电路）。

三、可编程器件的主要类型

- 1、可编程只读存储器PROM（ Programmable Read Only Memory ）-------与阵列固定、或阵列可编程
- 2、现场可编程逻辑阵列FPLA （Field Programmable Logic Array）------- 与阵列可编程、或阵列可编程
- 3、通用阵列逻辑 GAL（Generic Array Logic ） --------
- 与阵列可编程、或阵列固定、输出可组构OLMC
- 4、现场可编程门阵列 FPGA （ Field Programmable Gate Array）--------由可编程的逻辑单元组成的阵列，单元间的互连通道和输入、输出端口均可组构，现场在系统可编程。

7.2 随机读写存储器

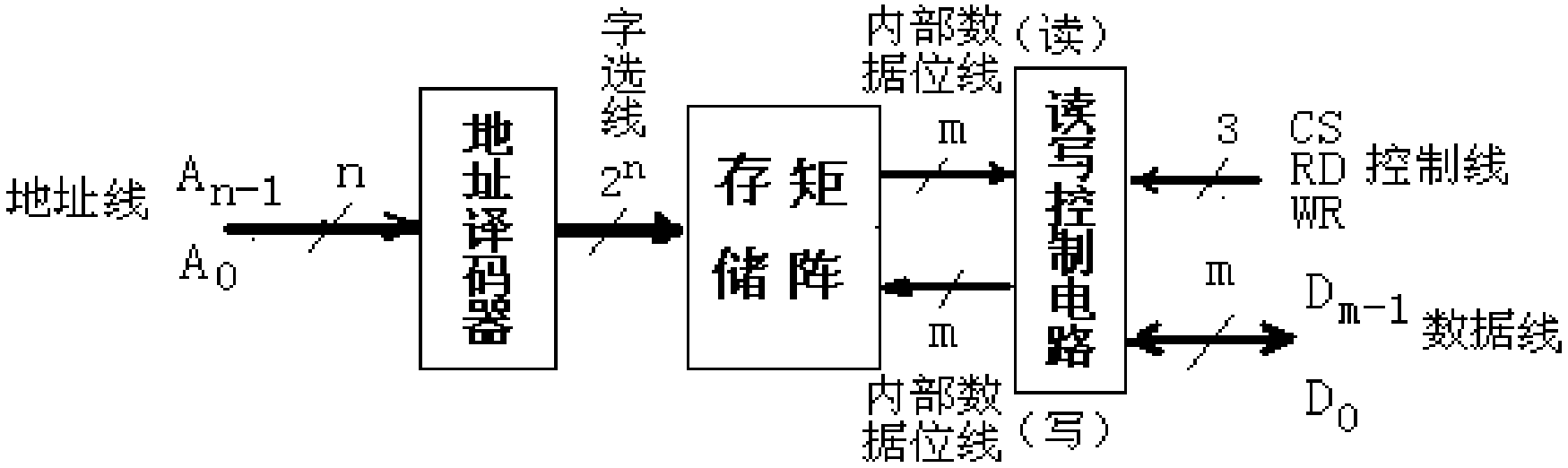
存储器根据访问的方式被分为只读存储器（Read Only Memory 简称ROM）和随机访问存储器（Random Access Memory 简称RAM）。

ROM内存储的数据是预先存入的，在系统运行中只能被读出而不能被修改，系统失电时数据保留，在计算机系统作为存放程序或数据库的部件。

ROM为组合逻辑电路，由与-或逻辑阵列构成。

RAM 内的数据存入和读出（访问）是任意的，时间和单元都不受限制，所以系统运行时可以修改数据（存入），但电路失电时数据丢失，在计算机系统中起暂存处理中间信息的内存功能。

RAM为时序逻辑电路，根据存储元的构成（触发器或电容）又分为静态SRAM和动态DRAM。



数据线（m条）——输入、输出存储数据：D_0~D_{m-1}

控制线（2~3条）——

片选线**CS**，有效时存储器工作，允 许数据存入或取出。

读控制**RD**，有效时存储器内的被地址线选中单元的数据通过数据线输出。

写控制**WR**，有效时数据总线上的数据存入被地址线选中的存储单元中。

读、写分时 操作，不能同时有效。

二、存储矩阵的结构：

存储矩阵由$2^n \times m$个存储元（触发器或电容）构成。每m个单元被分成一组，称为“存储单元”或“字”，每个字可以被唯一的一组地址码选中访问，“字”中的m位信息是被同时读出或写入。

每个存储单元中的存储元数m称为字长或“位”数，各个字中位序相同存储元被挂在同一条数据线上，通过同一个数据端口输入或输出数据。

所以，存储器的数据线数m等于其位数，存储器的地址码数n决定了其存储单元的字数N：

$$N= 2^n \text{ 。}$$

存储器的存储容量M （存储元数）等于其字数N乘以其位数m：

$$M= 2^n \times m$$

7.2.2地址译码方式

一、单译码方式

采用一

存

n位地

二、

采用

处

n位地

大

7.2.3

SRA

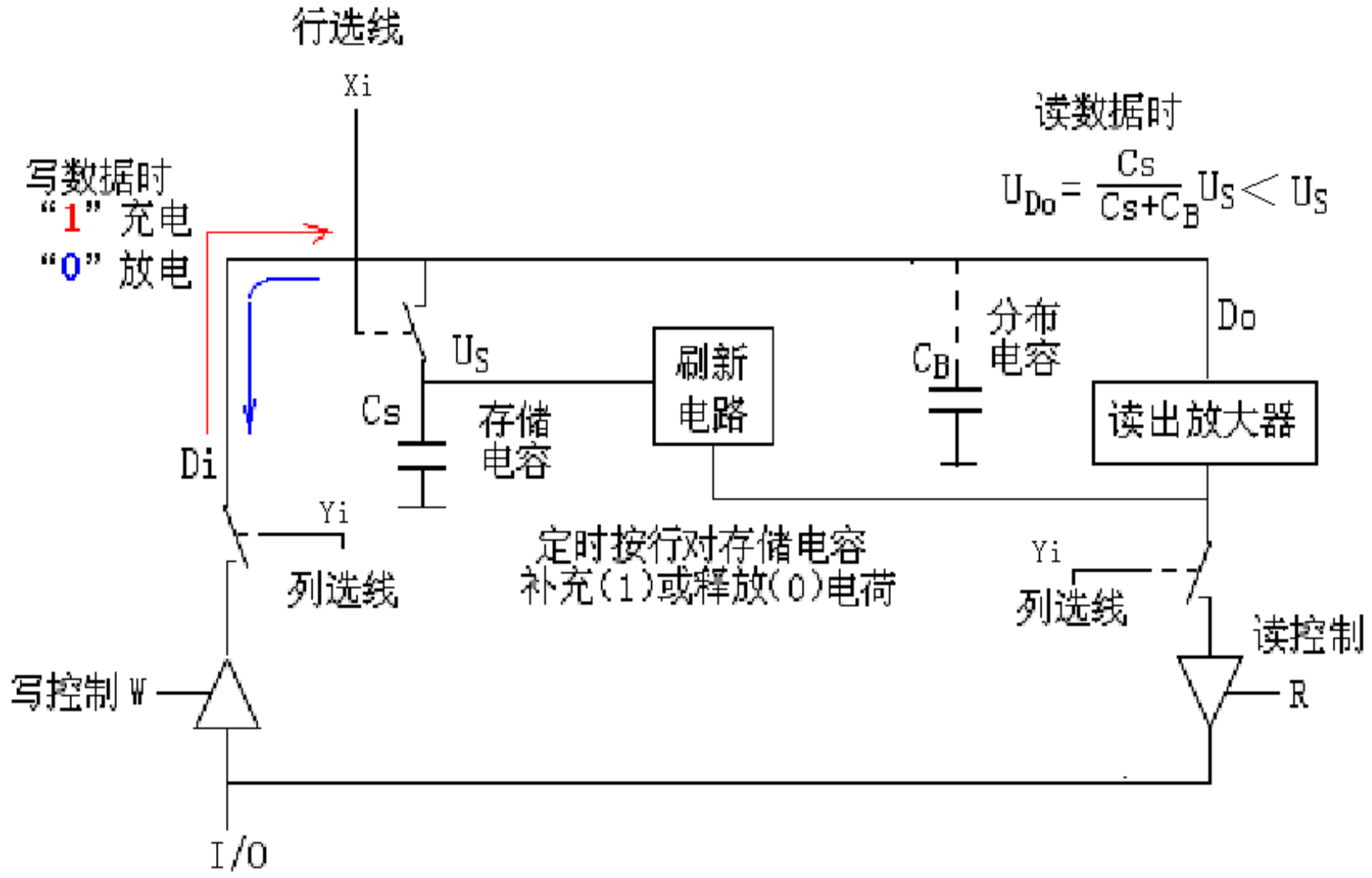
转

DRA

读

由于

读



存储元的读、写原理：

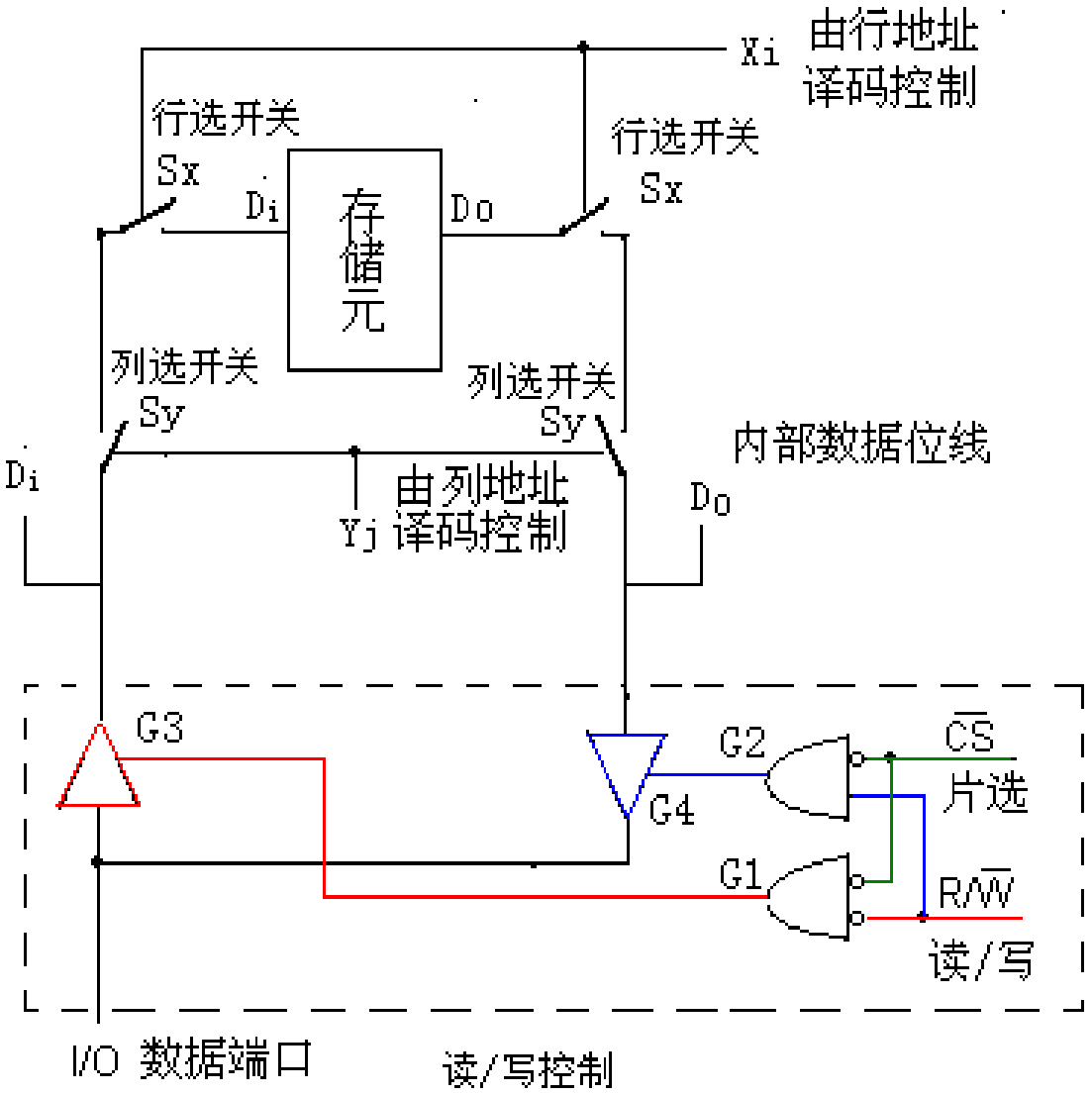
当片选信号无效时：

存储器的读/写控制电路被封锁，存储器的数据总线与外部隔离（G1、G2输出0、G3、G4输出高阻）。

当片选信号CS有效时：

若读信号有效，各位信息通过读/写控制电路被输出到数据端口（G1输出0、G3输出高阻、G2输出1、G4选通）；

若写信号有效，、数据端口的信息通过读/写电路存入各存储元（G1输出1、G3选通、G2输出0、G4输出高阻）。



7.2.4存储器容量的扩展

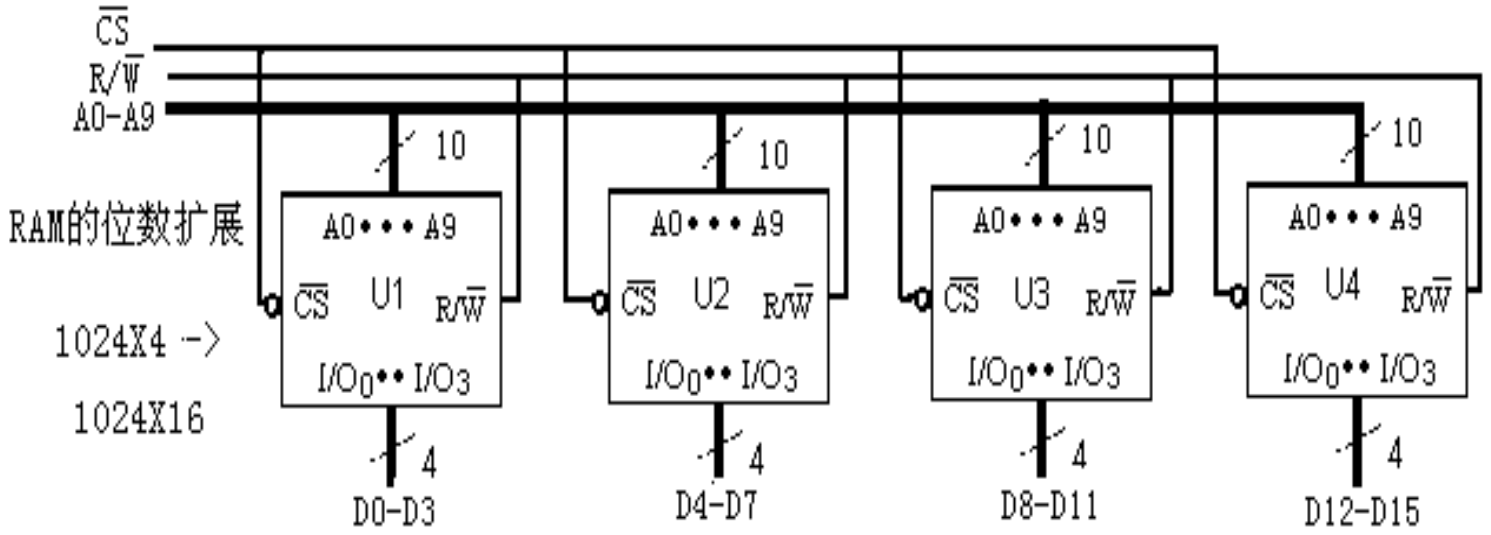
一、位扩展（存储字数不变、数据位数增加）

地址码位数不变，需要的存储器数量：

P=扩展后的位数÷存储器原位数

各存储器连接方式：地址线、控制线一一对应连接，数据线（存储数据位数）分别引出，位数增加。

例：用四片1024×4的存储器扩展成1024×16的存储系统。



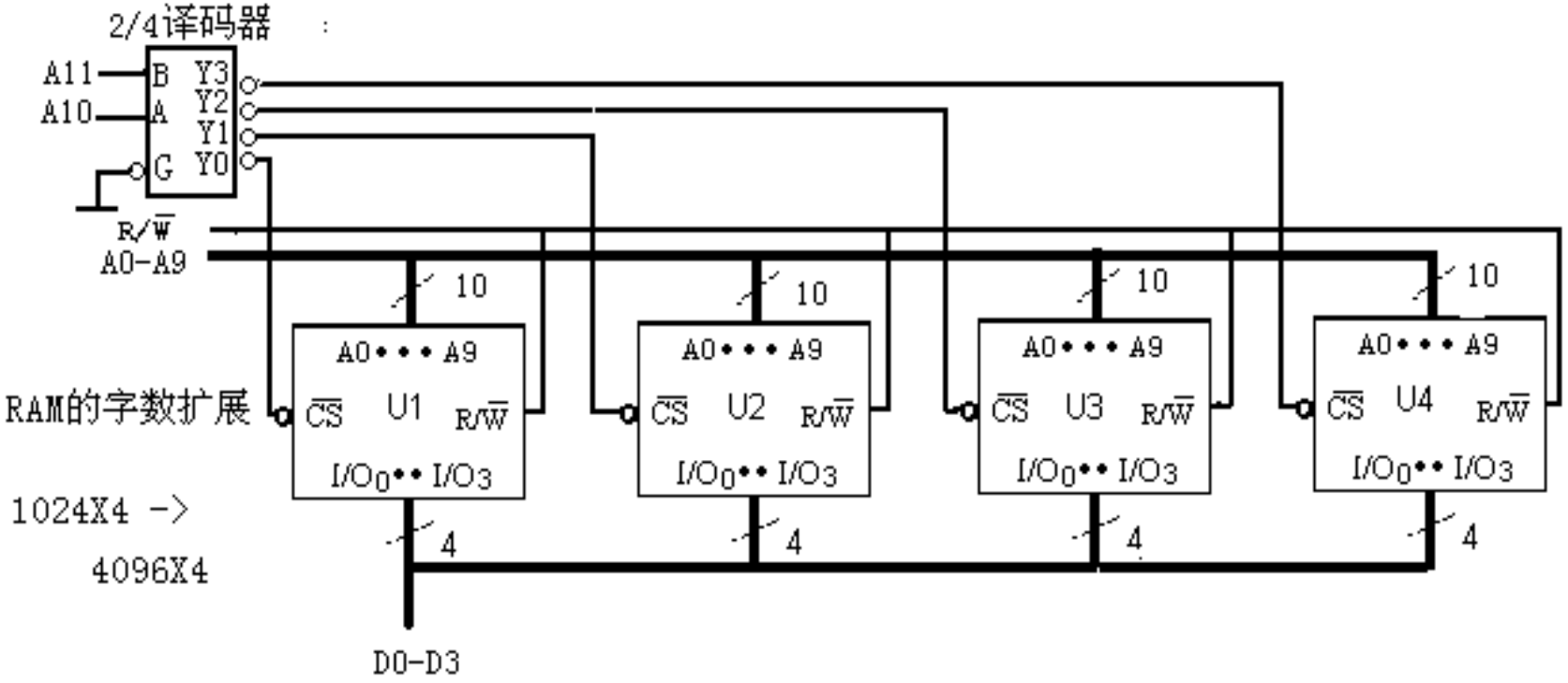
二、字扩展（数据位数不变、存储字数增加）

数据线数不变，需要的存储器总数：

P=扩展后的字数÷存储器原字数

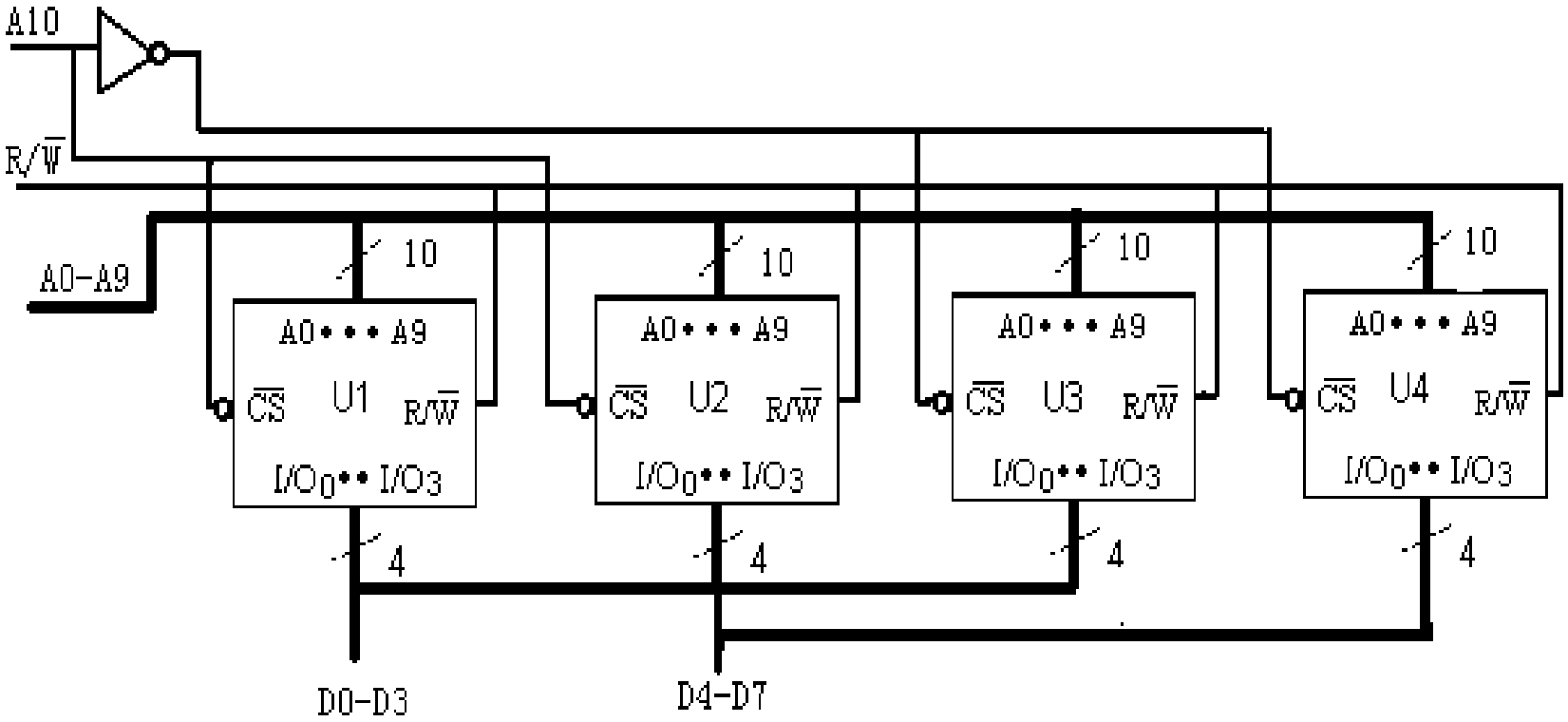
各存储器连接方式：数据线、存储器原地址线、读写控制线一一对应连接。扩展后增加的高位地址码译码后控制各存储器的片选端CS。

例：用四片1024×4的存储器扩展成4096×4的存储系统。增加的高位地址码A10、A11译码后控制四片存储器的片选CS。



三、位数、字数同时扩展。先按字数扩展方法连接，然后增加相同的电路进行位数扩展。

例：用四片1024×4的存储器扩展成2048×8的存储系统。增加的一位地址码A10以不同的状态分别控制两组存储器，每组存储器的位数构成扩展成8位。



RAM的字长、字数同时扩展

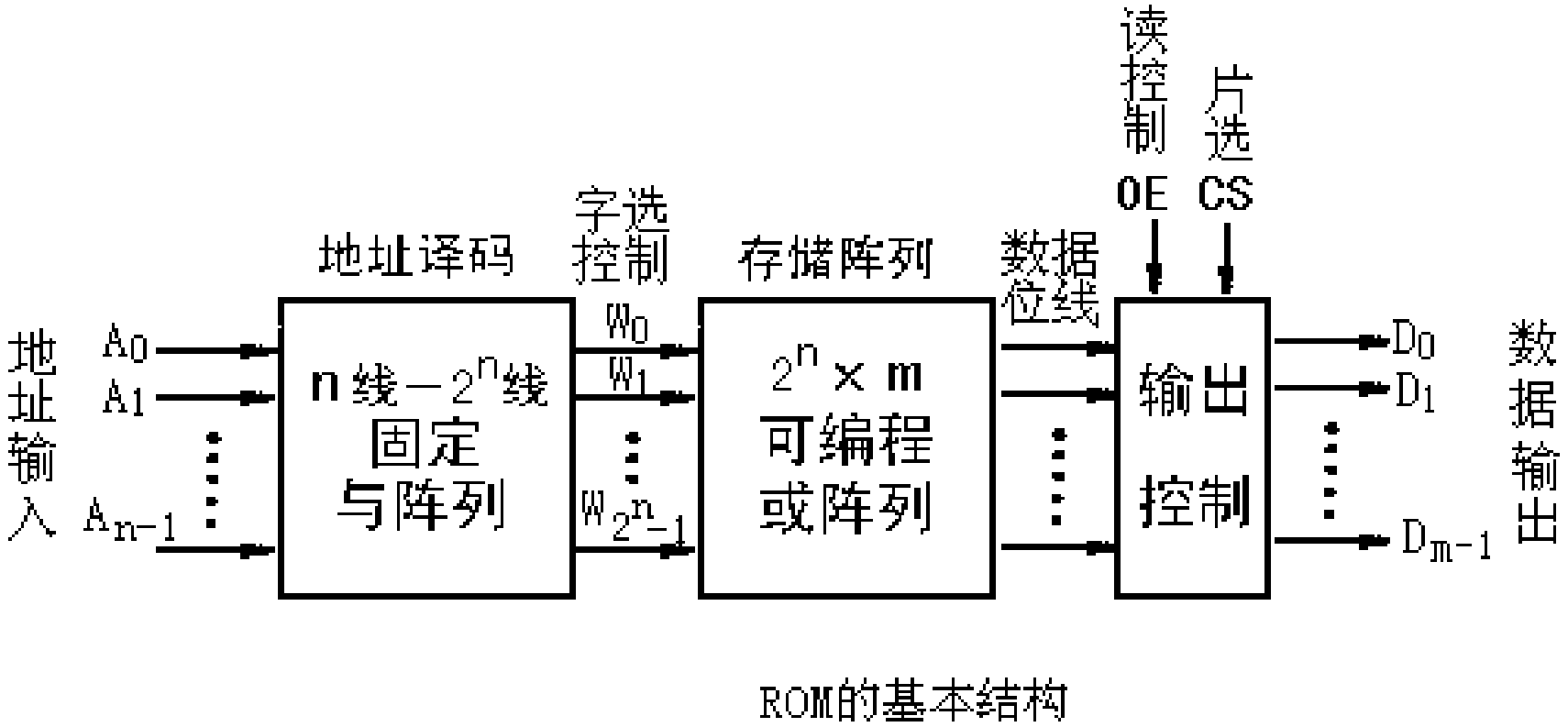
4.3只读存储器（Read Only Memory）

一、功能：

在系统运行时，存储器内部的数据只能读出，不能被修改。

二、结构：

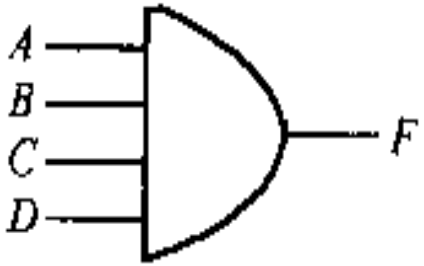
由地址译码器、存储矩阵和三态输出缓冲级组成。



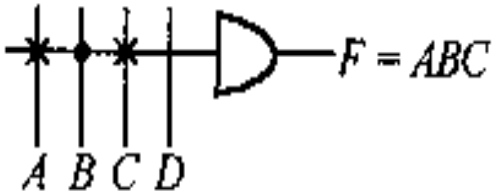
ROM的基本结构

三、PLD电路的表示方法

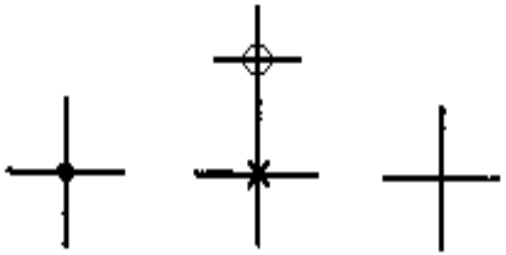
（a）输入缓冲器符号



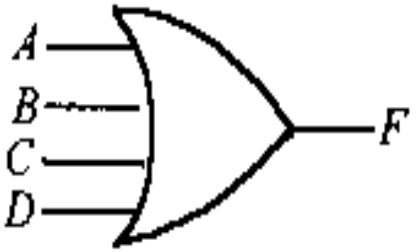
（b）标准与门逻辑符号



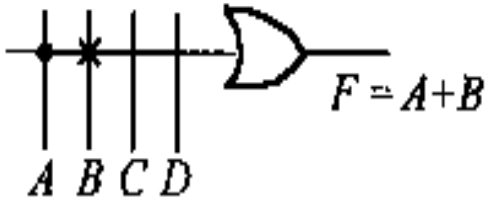
（c）可编程与门逻辑符号



（d）固定连接　可编程连接　不连接



（e）标准或门逻辑符号

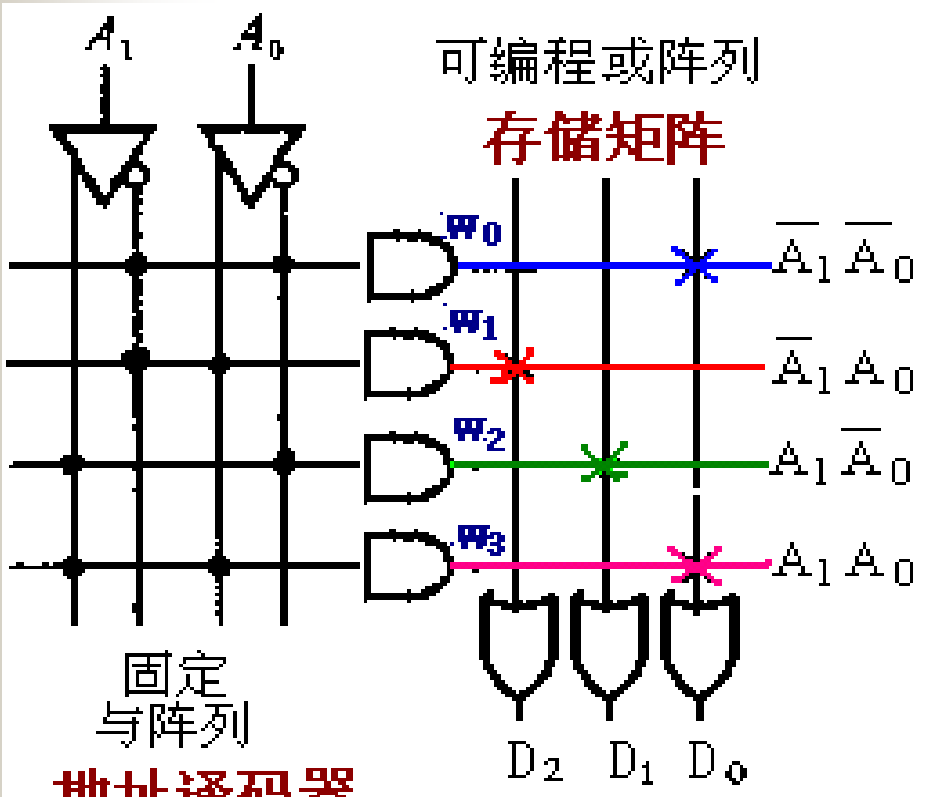


（f）可编程或门逻辑符号

四、ROM电路的与-或阵列结构

ROM的地址译码器可以用n输入、 2^n输出的固定与阵列表示；存储矩阵可以用 2^n输入、m输出的可编程或阵列表示。

与阵列的每个输出字线与或阵列输入位线的交点是一个可编程的存储元，当存储数据为“1”时，交点连接；当存储数据为“0”时，交点断开， 或门的输入与该与门输出没有关系。



地址 A_1 A_0	存储内容 D_2	D_1	D_0
0 0	0	0	1
0 1	1	0	0
1 0	0	1	0
1 1	0	0	1

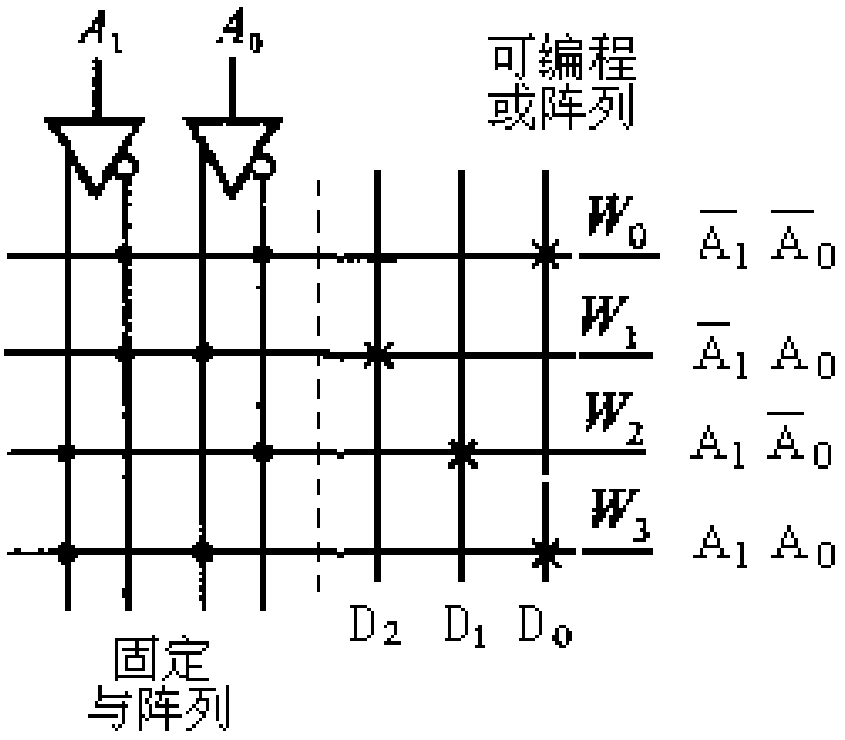
PROM的电路描述

ROM的与-或阵列可以再简化为只用字线和位线示意。

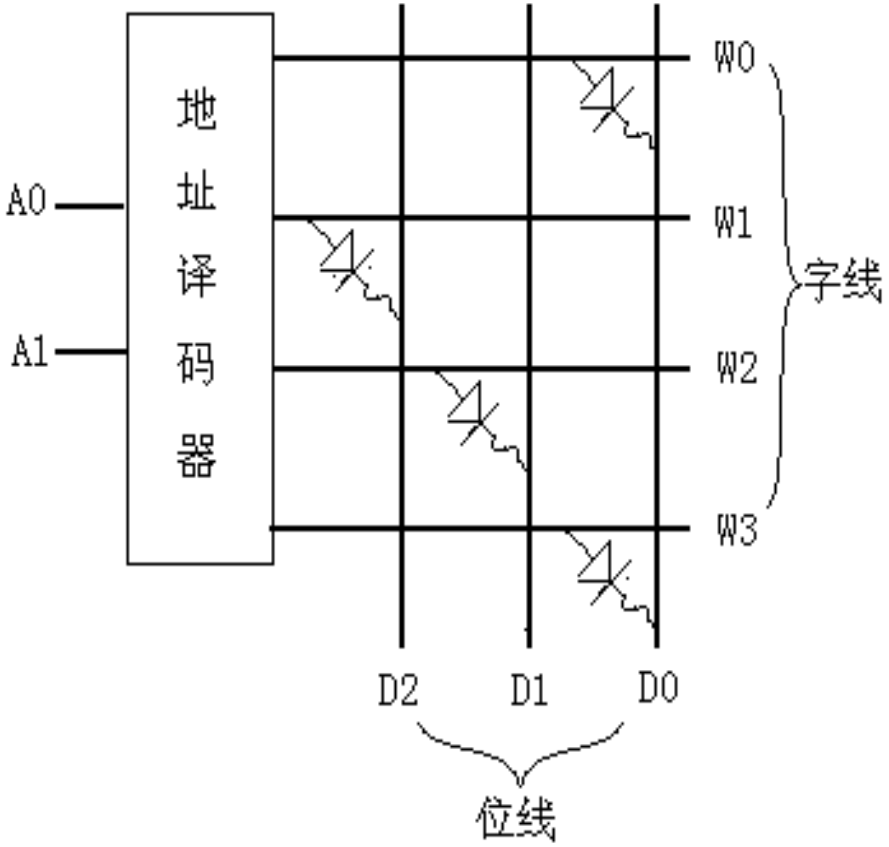
图左的列线是经缓冲后的互补输入， 2 个输入 4 （2对）个变量；

图中的行线是各与门的逻辑关系， 2 个输入有4个（2^2）与项（最小项），行、列交点表示了各变量和与门输入间的联系。

图右的列线是各或门的逻辑关系，4位数据输出有4个或门，行、列交点表示了各与门输出和或门输入间的联系。



PROM 的简化描述



7.3.1 ROM的分类

1、ROM——用户不可编程，掩膜式编程工艺。用于大批量定型产品。

2、PROM——一次性编程，熔丝或PN结击穿编程工艺。用于小批量定型产品。（Programmable ROM）

3、 EPROM——可擦除编程 (编程器编程，紫外光擦除)。用于产品的研发过程。（Erasable PROM）

4、 E^2PROM（EEPROM ）——电可擦除、现场编程，但速度较低（毫秒级/字节），用于存储系统断电后需要保存的数据。（ Electrically Erasable PROM）

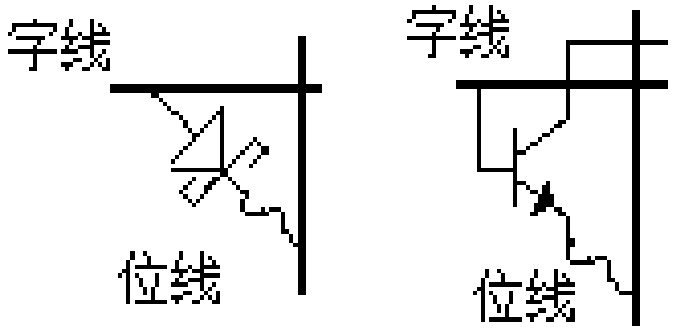
5、 FLASH——闪速存储器，结构类似E^2PROM、擦除、编程速度略低于RAM，是目前广泛应用的只读存储器。

7.3.2 ROM结构与工作原理

一、PROM的编程原理

1、PROM一次性编程原理

可编程点是由带熔丝的半导体开关管 或反熔丝的介质构成的。熔丝型开关出厂时所有编程点都连通，如图所示。如果用户希望编程点断开，可以通以较大的电流使熔丝烧断。反熔丝开关的核心是介质，出厂时开关呈现很高的阻抗(>100MΩ)，相当于编程点断开。编程时利用高电压将介质击穿，开关接通。



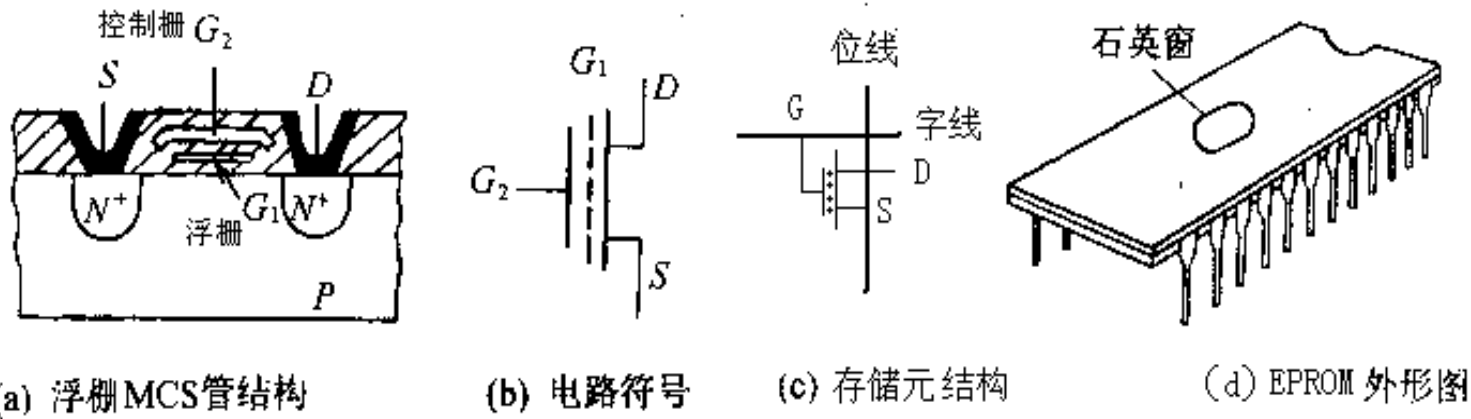
熔丝和反熔丝器件编程后不可恢复，所以是一次性编程的，但抗干扰性能好，适用于可靠性要求高的定型产品。

2、EPROM可擦除、可编程原理

利用悬浮栅MOS管构成可编程的电子开关。编程时利用编程器产生高压脉冲对悬浮栅注入电子，使电子开关断开。擦除时通过紫外线照射释放悬浮栅上的电子，使电子开关接通。器件失电后编程信息（电子）保持，并可重复多次编程，但密度不高。

3、EEPROM电可擦除、可编程原理

芯片内部集成了编程/擦除控制电路和高压脉冲产生电路，所以器件可在用户系统上由正脉冲编程、负脉冲擦除（在系统可编程技术）



(a) 浮栅MCS管结构　(b) 电路符号　(c) 存储元结构　(d) EPROM 外形图

二、ROM的逻辑关系

本例ROM的存储内容实现了一个1位二进制数比较器：对地址口输入的两个一位的二进制数A1、A0进行比较。数据口输出3个高电平有效的开关量：D2表示A1小于A0；D1表示A1大于A0；D0表示A1等于A0。

所以，ROM可以实现n输入、 m输出的组合逻辑函数，函数变量从ROM的地址口输入、函数值从数据口输出、存储表内容为函数的真值表。

例：4字3位的ROM结构。

由阵列逻辑可得各输出表达式Di。

D_2（A_1、A_0)

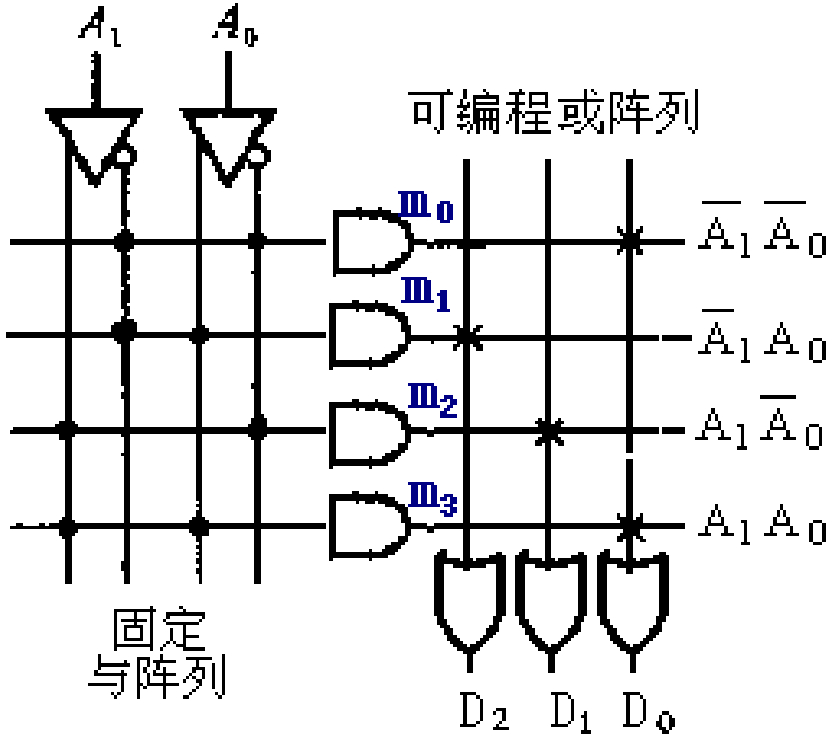
$=m_0\ 0+m_1\ 1+m_2\ 0+m_3\ 0=m_1$；

D_1（A_1、A_0)

$=m_0\ 0+m_1\ 0+m_2\ 1+m_3\ 0=m_2$；

D_0（A_1、A_0)

$=m_0\ 1+m_1\ 0+m_2\ 0+m_3\ 1=m_0+m_3$。



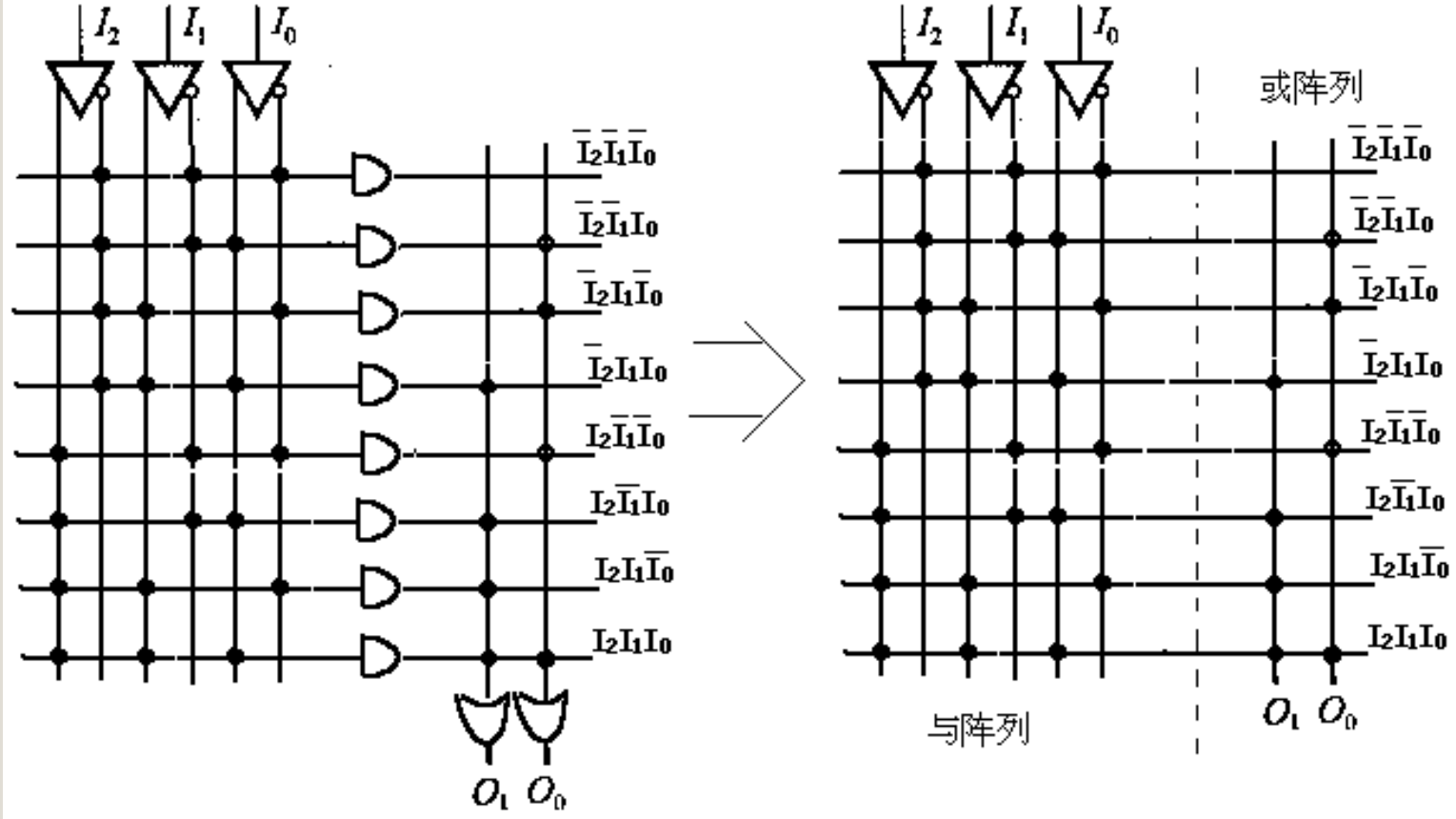
3位地址码（8字）、2位数据的ROM结构：

由存储矩阵编程关系可知各输出表达式Oi。

O_1（ I_2、 I_1、 I_0) =m_3+m_5+m_6+m_7;

O_0（ I_2、 I_1、 I_0) =m_1+m_2+m_4+m_7。

若输入的I_2~ I_0是3个一位的二进制数，ROM实现了全加器的功能， O_1是进位输出、 O_0是相加和输出。



7.3.3 ROM的应用

一、产生组合逻辑函数

n位地址、m位数据的ROM可以产生n输入、m输出的组合逻辑函数。

函数的变量从地址码输入端输入，函数值从数据端输出，一个数据口输出一个n变量的逻辑函数。

实现方法：

（1）将要产生的函数表达式整理成最小项表达式。

（2）按最小项中函数变量的位序输入ROM的地址口

（3）选定函数输出的数据端。

（4）根据各数据位代表的函数最小项表达式，将表达式中最小项（使函数值为“1”）对应地址的存储元填“1”（或阵列的编程点保留），否则填“0”（编程点断开）。

二、应用举例

1、代码转换

要转换的代码作为ROM的地址输入，所对应的转换输出码。例5：

平方函数存储信息表

X	地址 $A_3A_2A_1A_0$	存储数据 $D_7D_6D_5D_4D_3D_2D_1D_0$	$Y=X^2$
0	0000	00000000	0
1	0001	00000001	1
2	0010	00000100	4
3	0011	00001001	9
4	0100	00010000	16
5	0101	00011001	25
6	0110	00100100	36
7	0111	00110001	47
8	1000	01000000	64
9	1001	01010001	81
10	1010	01100100	100
11	1011	01111001	121
12	1100	10010000	144
13	1101	10101001	169
14	1110	11000100	196
15	1111	11100001	225

2、字符发生器——产生控制点阵

将字符的点阵数据预先存储在ROM储矩阵中逐行读出点阵显示数据送入

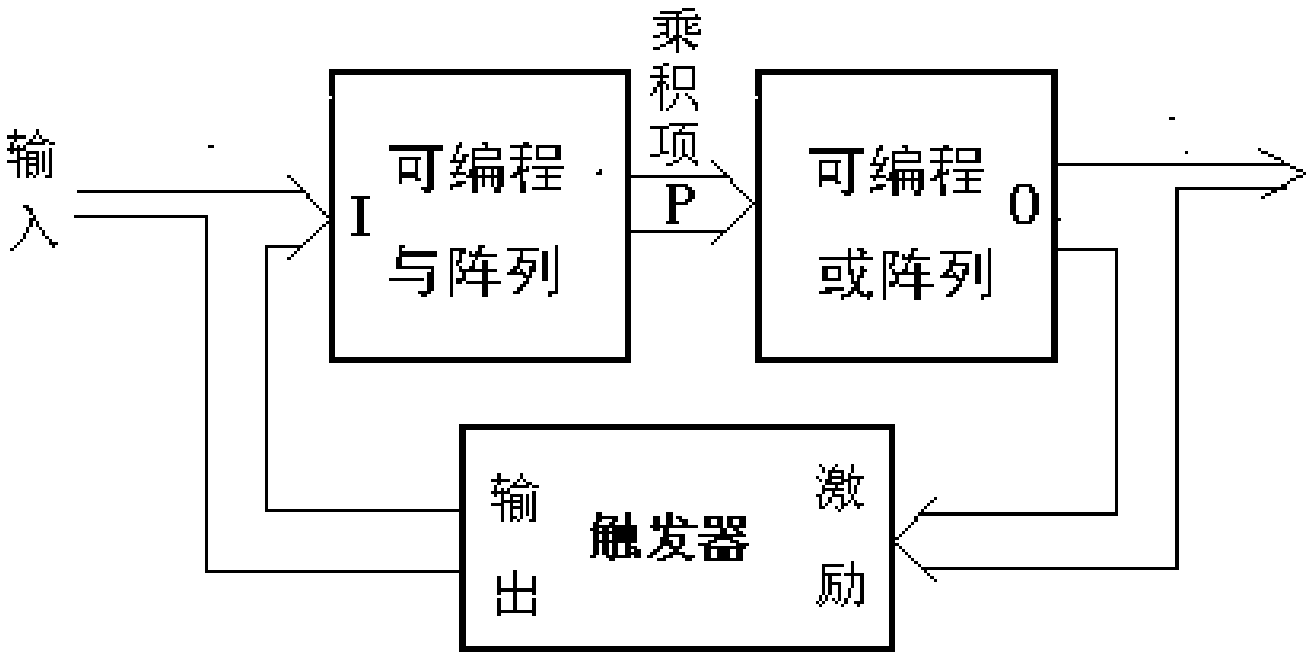
•点阵显示的每个点是一个发光二极管极管的阴极相连，各行发光二极管的平，则各行信号控制该列的发光二极

3、数学函数表

把因变量和自变量的函数关系存在ROM中，函数的自变量作为地址码输入，因变量作为数据输出，函数关系用二进制数表示填在存储单元中。比如平方表，8位的ROM可存0~15的平方值（16个字）。

7.4现场可编程逻辑阵列（Field PLA）

7.4.1 FPLA的结构和特点

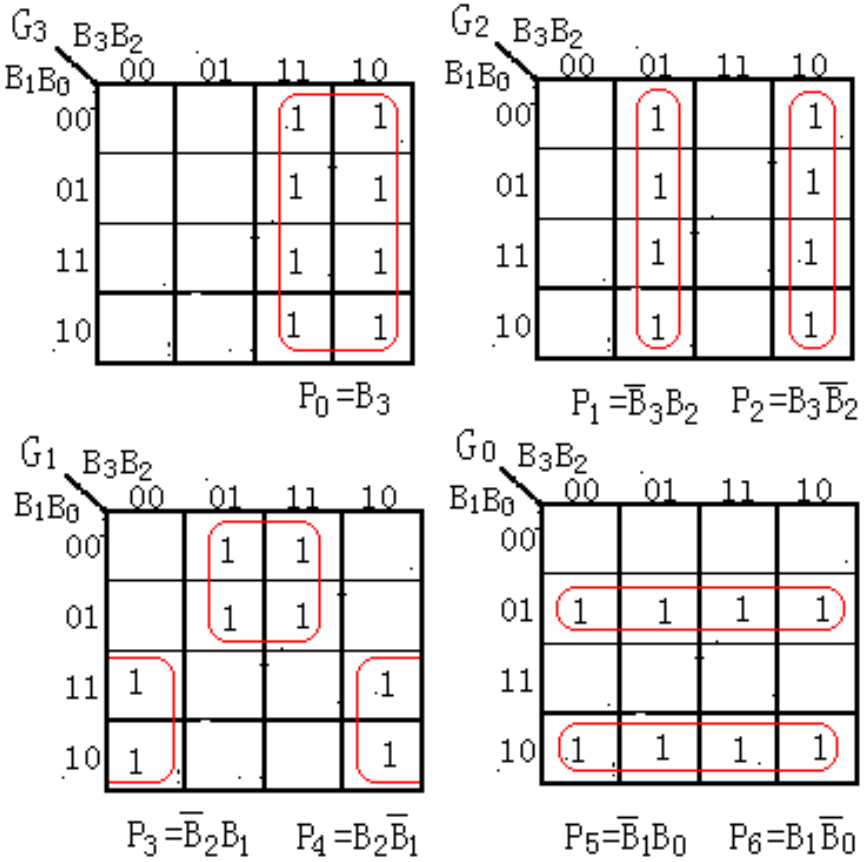


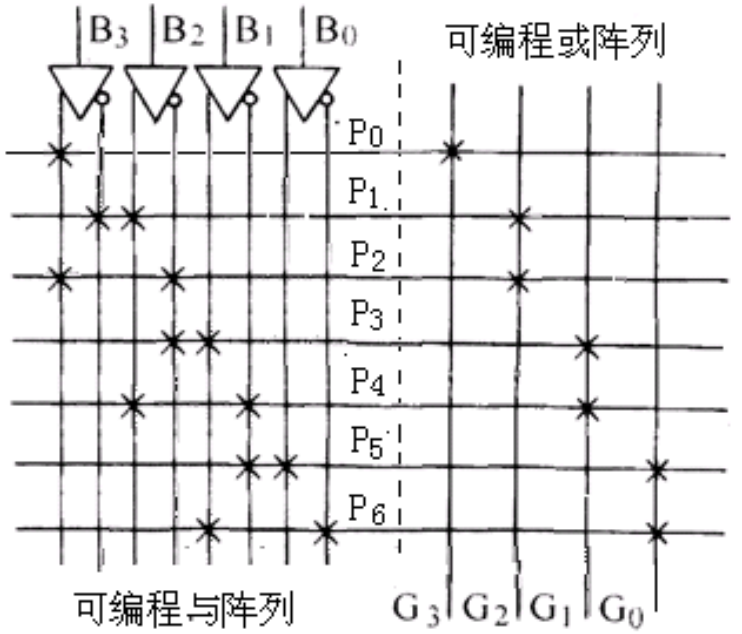
时序FPLA的基本结构

组合FPLA的基本结构

2、时序FPLA内含触发器，可以实现时序逻辑函数。

7.4.2FPLA实现组合逻辑函数





B－G码变换器的FPLA阵列图

$G_2(B_3、B_2、B_1、B_0)=m_4+m_5+m_6+m_7+m_8+m_9+m_{10}+m_{11}$

$=B_3 \oplus B_2=\overline{B}_3B_2+B_3\overline{B}_2=P_1+P_2$

$G_1(B_3、B_2、B_1、B_0)=m_2+m_3+m_4+m_5+m_{10}+m_{11}+m_{12}+m_{13}$

$=B_2 \oplus B_1=\overline{B}_2B_1+B_2\overline{B}_1=P_3+P_4$

$G_0(B_3、B_2、B_1、B_0)=m_1+m_2+m_5+m_6+m_9+m_{10}+m_{13}+m_{14}$

$=B_1 \oplus B_0=\overline{B}_1B_0+B_1\overline{B}_0=P_5+P_6$

7.4.3FPLA实现时序逻辑函数

例：分析用FPLA实现时序逻辑电路的功能。

解：触发器的输出反馈回与阵列的输入，与-或阵列的输出控制触发器的激励。

1、由与-或阵列得各触发器激励方程

$J_0=\overline{Q}_2+\overline{Q}_1$，$K_0=\overline{Q}_2+Q_2=1$；$J_1=Q_0$，$K_1=Q_2+Q_0$

$J_2=Q_0Q_1$，$K_2=Q_1$

2、激励方程代入触发器特性方程得电路状态方程

$$Q_0^{n+1}=\overline{Q_2^nQ_1^n+Q_0^n}\;;\quad Q_1^{n+1}=\overline{Q}_1^nQ_0^n+\overline{Q}_2^nQ_1^n\overline{Q}_0^n$$

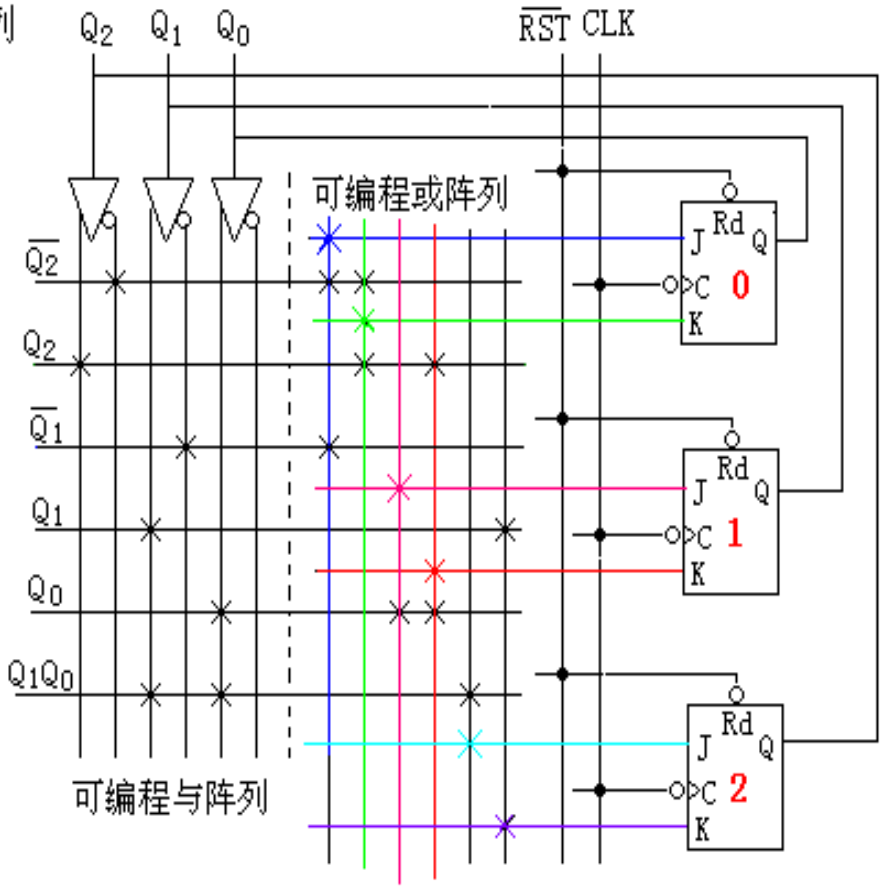
$$Q_2^{n+1}=\overline{Q}_2^nQ_1^nQ_0^n+Q_2^n\overline{Q}_1^n$$

3、状态转换关系：

111->000->001->010->011->100->101->110

4、计数脉冲由FPLA的公共时钟端输入，3个触发器的清零可由FPLA的公共复位端控制。

5、功能：同步七进制加计数器，有自启动能力。



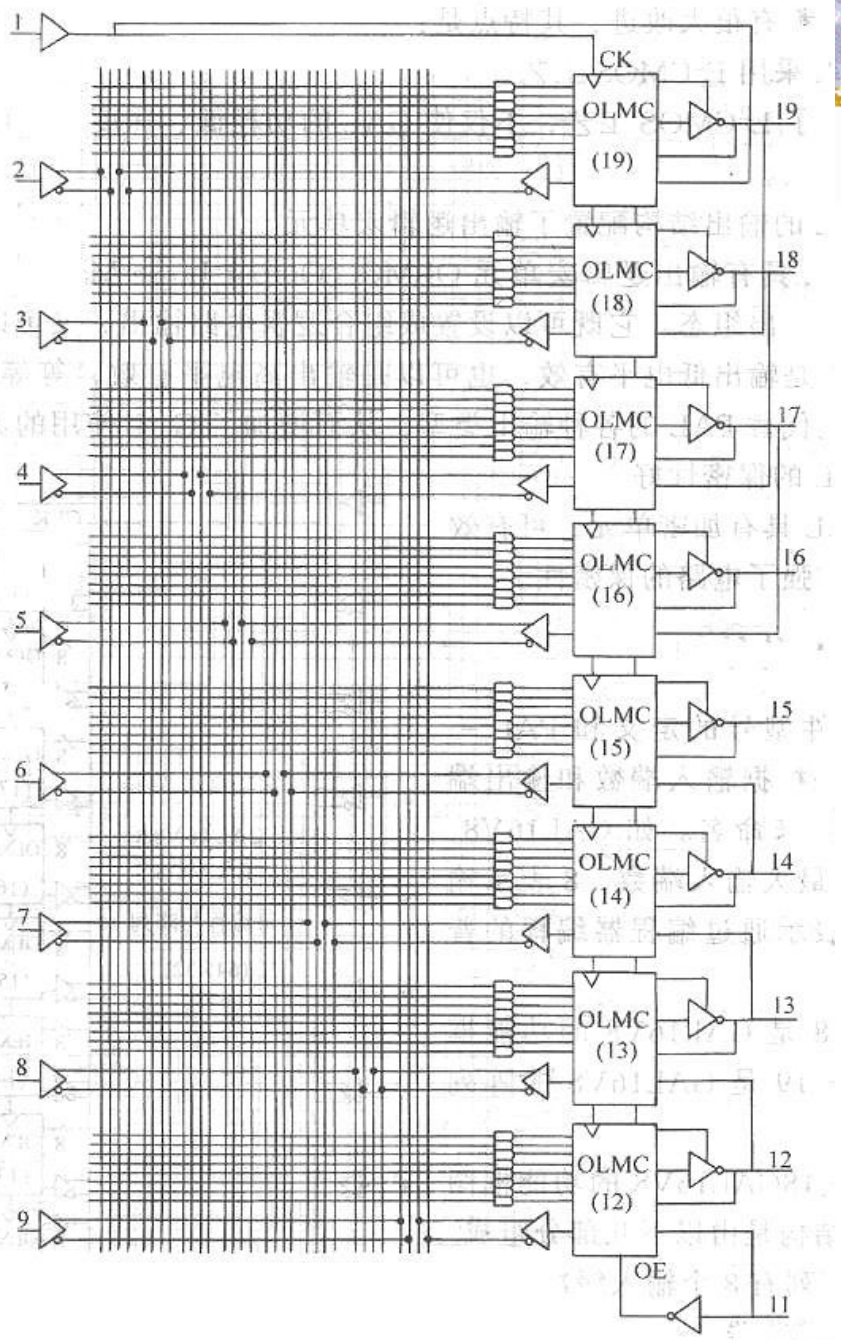
FPLA实现的同步七进制加法计数器

7.5通用逻辑阵列GAL

基本结构——

由输入、输出缓冲和可编程的与阵列以及输出逻辑宏单元OLMC组成。GAL的型号表示了其输入、输出的规模。GAL16V8表示其输入信号最多可达16个，输出端数可达8个。V表示输出方式可编程，所以共有8个可编程的输出逻辑宏单元OLMC。还有32×64的可编程与阵列、8个输入缓冲器（引脚2-9）、8个三态反相输出缓冲器（引脚12-19）和8个反馈输入缓冲器（引脚1、11-14、17-19）。16个输入通过缓冲器构成同相和反相共32个变量，64个乘积项均分成8组通过OLMC组态输出。

有触发器的公共时钟CK和三态输出的公共使能OE。



GAL16V8 的原理图

OLMC——含不可编程的或门、触发器、数据选择器等。

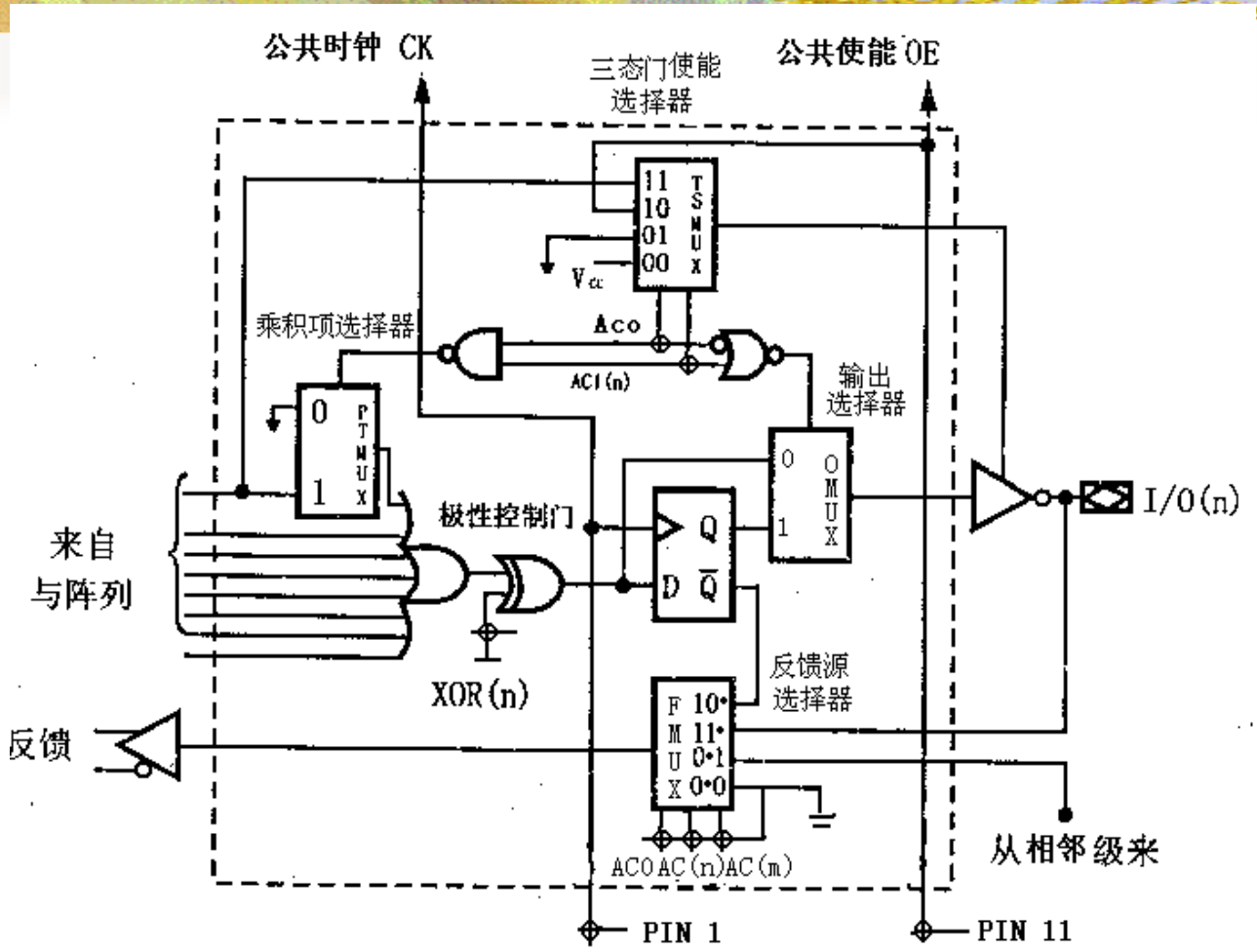
数据选择器的功能：

乘积项选择PTMUX：选择第一与门是否输入或阵列。

三态门控制选择TSMUX：选择输出三态门的控制源是第一与门输出、Vcc、GND或公共使能OE。

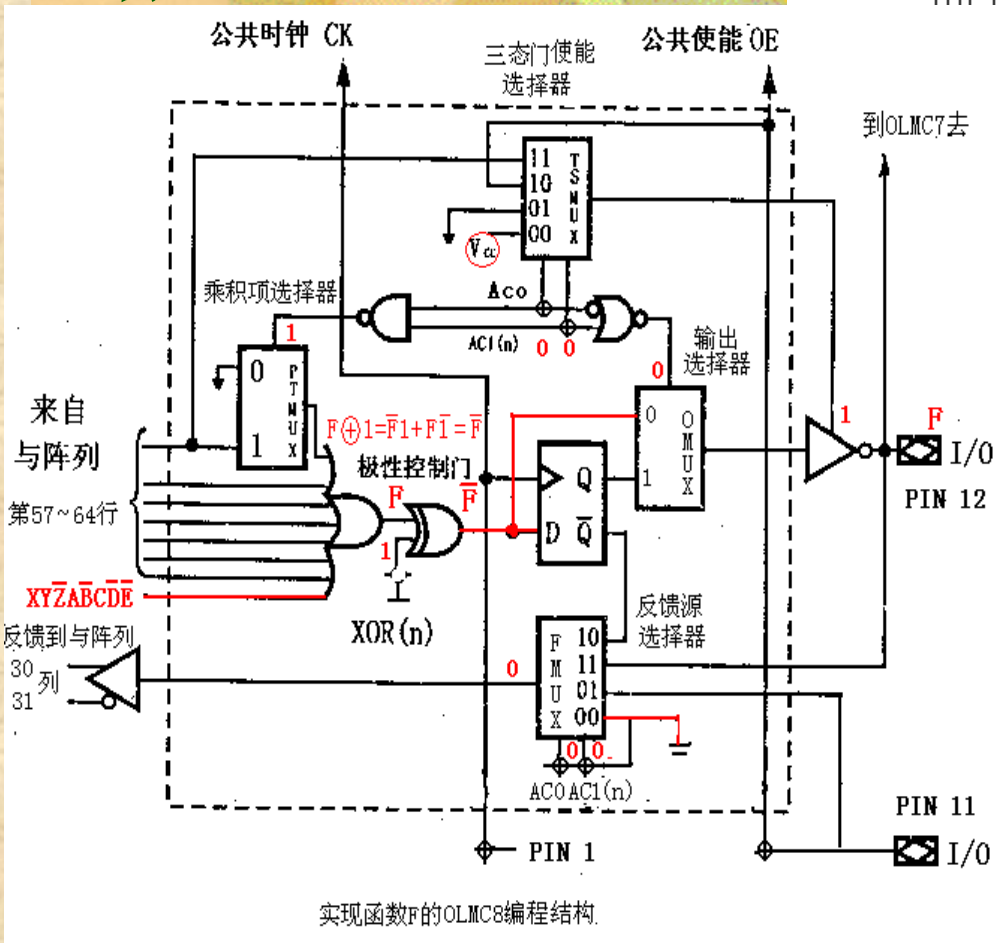
反馈控制选择FMUX选择反馈回与阵列的信号源是寄存器输出、端口输入、邻级的输出或接地（无反馈）

输出控制选择OMUX选择组合输出或寄存器输出,使GAL可以实现不同的输出方式.



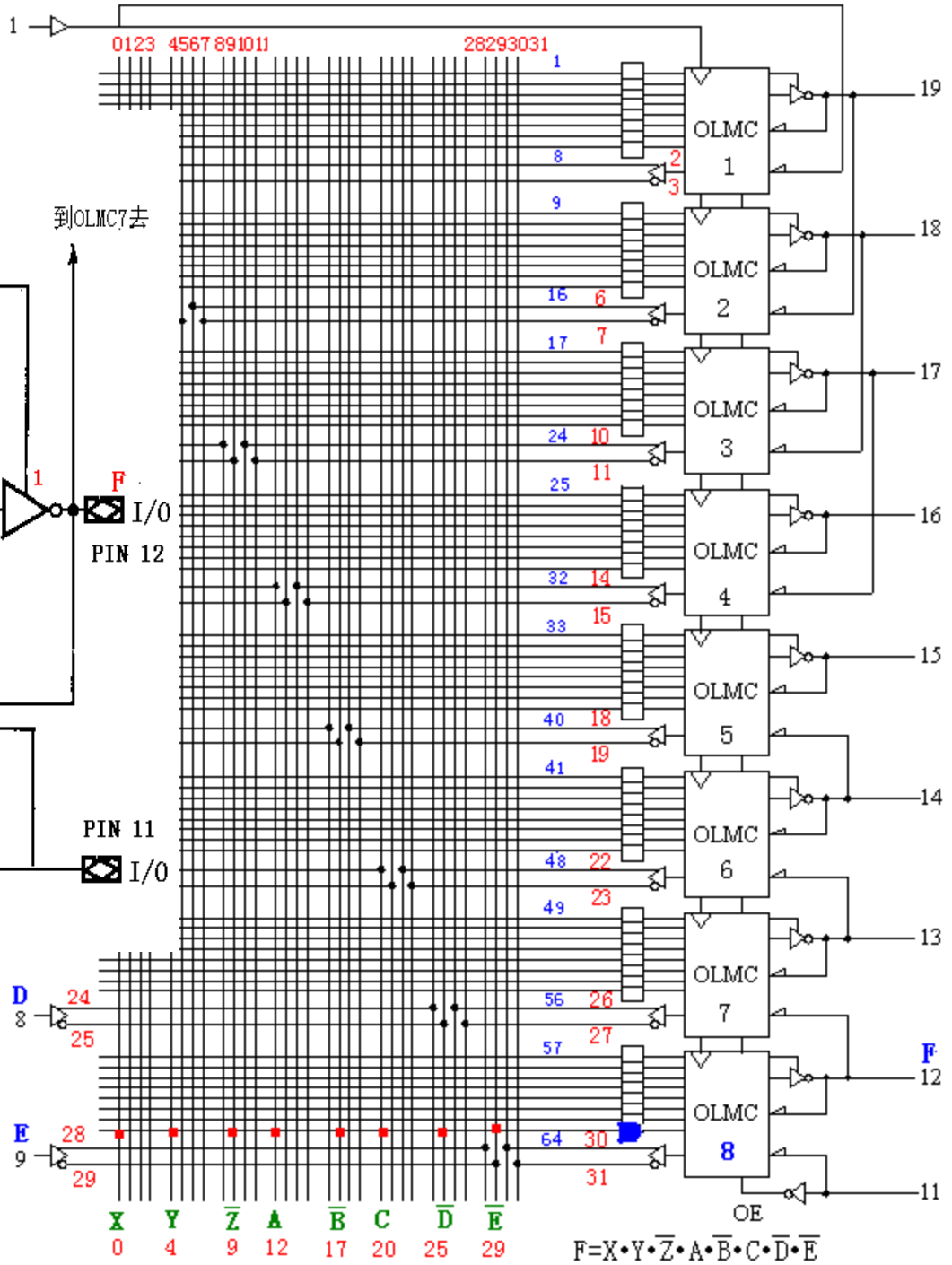
OLMC的结构原理

例：



实现函数F的OLMC8编程结构

25、29编桯连接（红点表示），将各列变量相乘后通过OLMC组态从引脚12输出，实现F函数的功能。 由于只有一个与项，所以OLMC的组态为：



7.6现场可编程门阵列FPGA

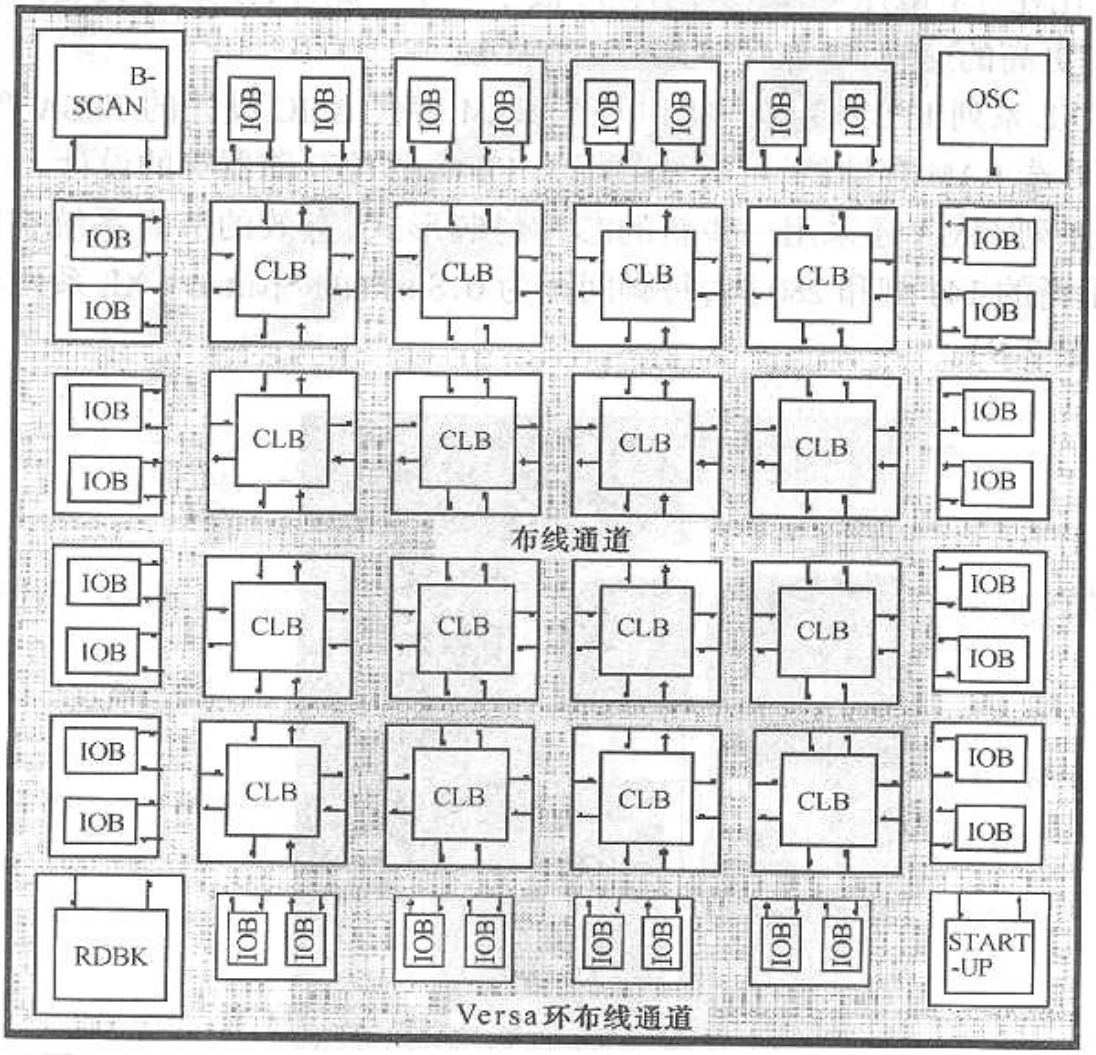
FPGA的基本结构以独立可编程的逻辑单元LCB （单元型PLD），组成的阵列LCA使器件的集成度更高，资源更丰富，适用实现逻辑运算能力较强的数据密集型数字系统。

目前国内教学中应用较多的PLD:

公司	产品型号举例	支持 辅助设计软件
Lattice	isp1000系列	isp EXPERT
Altera	FLEX10K系列	Quartus II4.2
Xilinx	Virtex系列	ISE 6.3

7.6.1FPGA的基本结构

右图是XILINX推出的FPGA-CX3020的结构示意图，中间是可配置的逻辑元CLB（Configurable Logic Block）构成的逻辑元阵列LCA（Logic Cell Array），邻近I/O端口的周围模块是输入、输出单元IOB，阵列中分布有可编程的开关矩阵PSM（Programmable Switching Matrix）和互连线资源，（布线通道）连接所有的CLB和IOB。



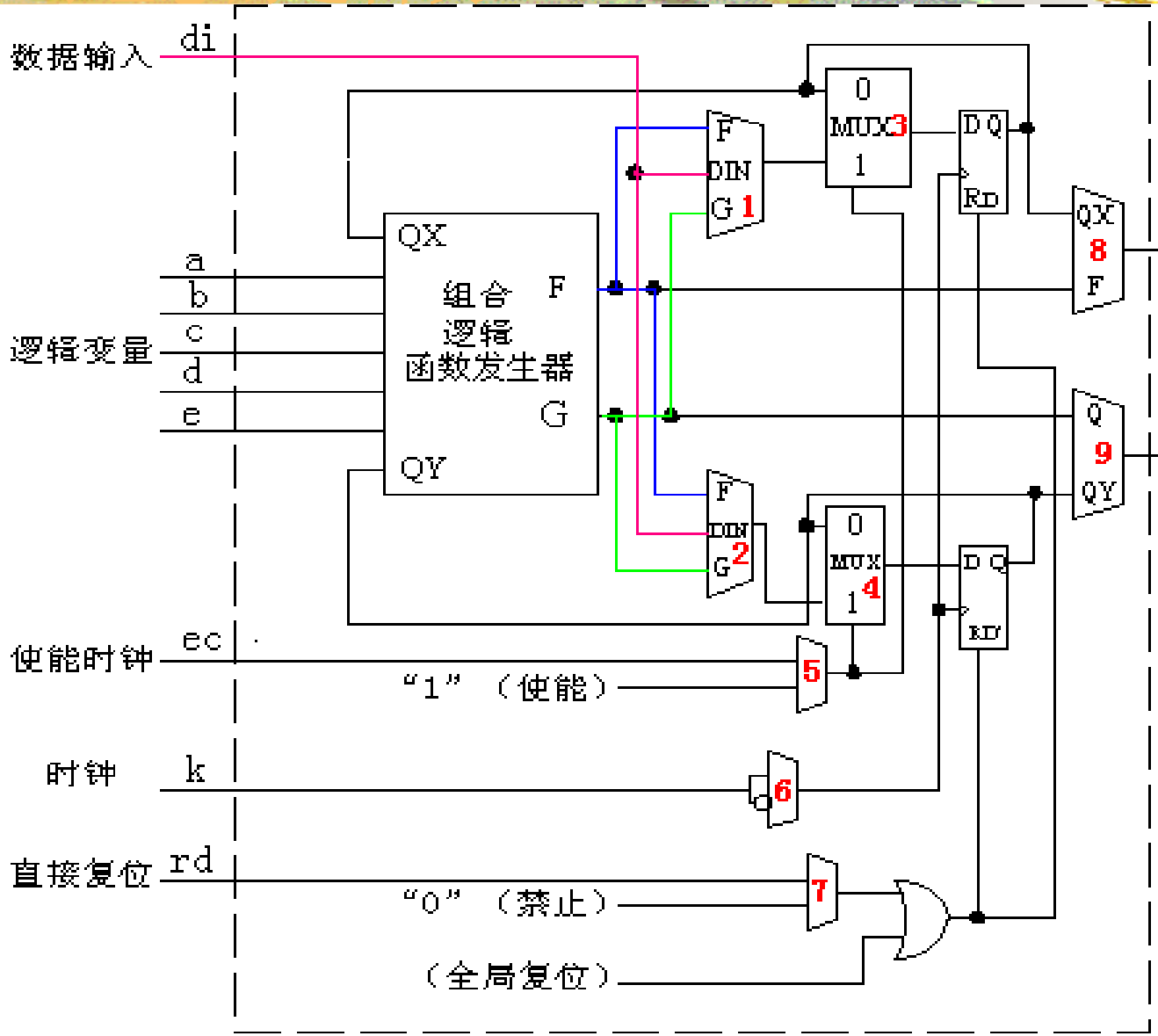
某款FPGA的结构示意图

一、可组态的逻辑块CLB

- 可
  择
1、组
  方
  反
  的
  或

2、9个
  9控

3、两
  Qx
  沿
  1）
  或



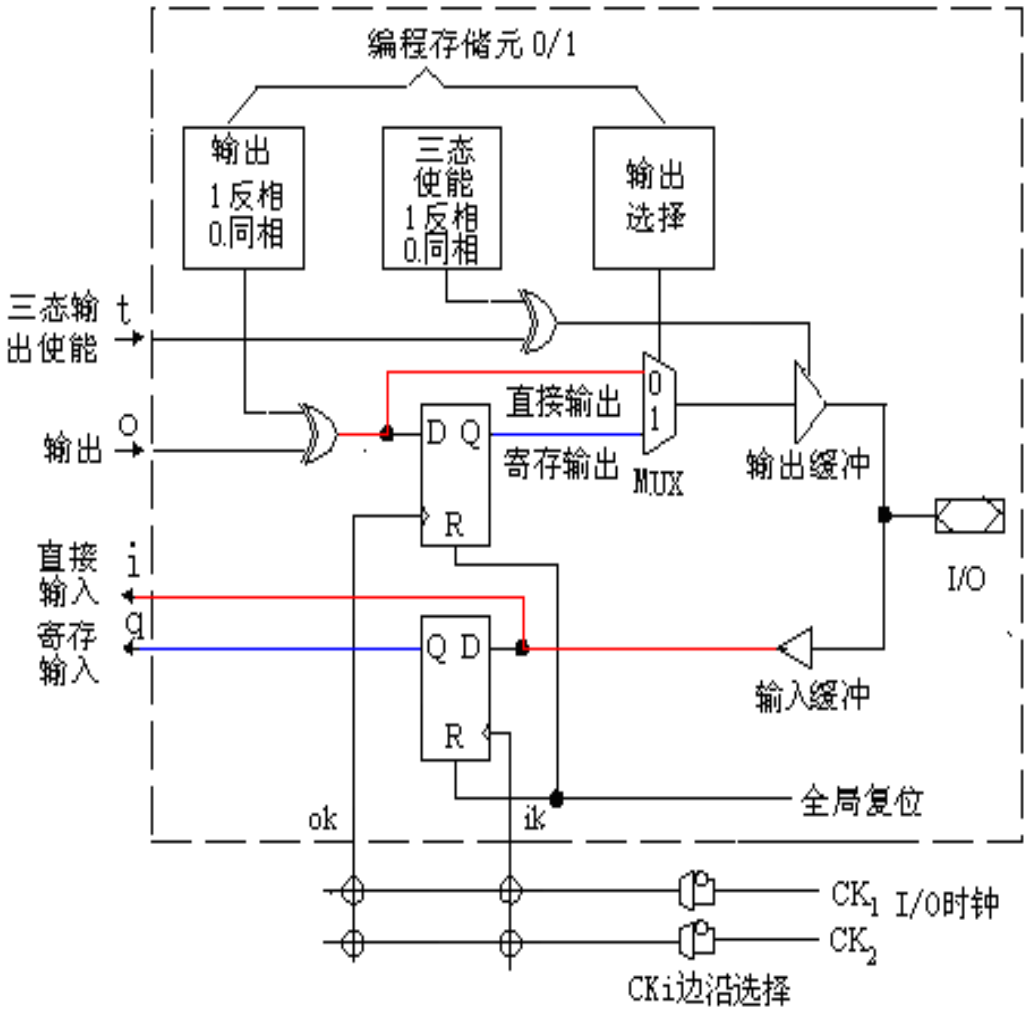
CLB 结构

二、输入、输出块IOB

IOB的作用是对FPGA的端口进行组态。

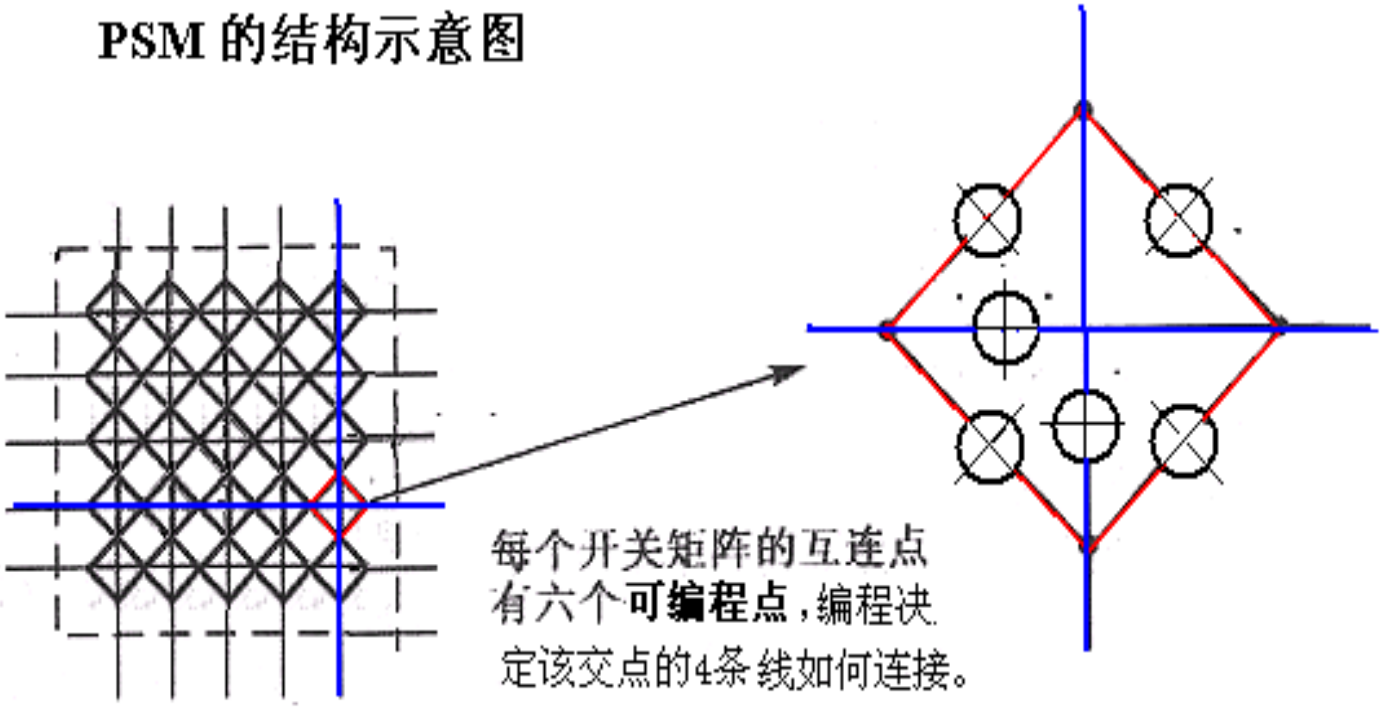
通过编程可以控制输出三态缓冲器，使每个I/O端口直接被组态成输入（三态缓冲器禁止）、输出（三态缓冲器选通）或双向端口（三态缓冲器由使能控制）。

通过编程选择I/O端口的缓冲输入是直接还是经寄存后向LCA传送。



三 可编程的互连线资源

1、



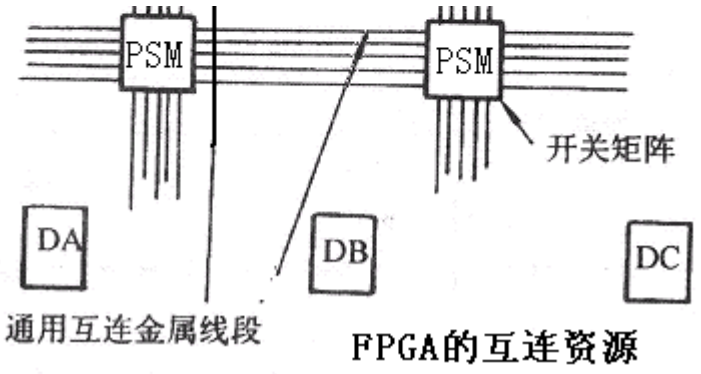
PSM
功能所

BC

CC

2、每个通用行、列线段的汇集
实现多信号转接的开关接线
过对其编程，可以实现该
实现需要的信号传输路径。
列线，每条线段可以实现2

接，



FPGA的互连资源

7.6.2 FPGA的配置和开发

一、 FPGA的配置

FPGA的编程信息存储采用了类似SRAM（静态随机存储器）的工艺，器件失电后编程信息丢失。所以，系统每次上电必须重新加载信息以重构电路 。电路重构的方式有两种：

被动重构——FPGA每次上电后由计算机重新下载编程信息文件。

主动重构——在目标系统电路板上配备只读存储器（PROM、EPROM、EEPROM），上电时由FPGA本身自动从只读存储器中获取编程信。

二、应用FPGA实现数字逻辑的设计方法：

(1)在计算机开发环境中输入满足设计功能要求的逻辑图（可直接调用开发环境提供的元器件库和宏单元库中的器件）或文本文件(硬件描述语言)。

(2)在开发环境中对设计项目进行功能仿真或速度测试，如不符合设计要求，重新修改设计文件。

(3)通过仿真测试后，在开发环境中对设计项目编译（翻译成逻辑表达式）并进行器件适配（确定FPGA的各CLB、IOB和互连资源的编程数据以实现所有的逻辑表达式满足设计要求的逻辑功能），生成熔丝图文件*.JED。

(4)将熔丝图文件通过计算机的并行口下载到安装在电路中的可编程器件（FPGA）中

(5)系统调试、观察设计实际效果。

(6)若需修改设计，重新编译、适配、下载。

三、PLD的性能特点

1、功能集成度高

2、设计灵活，电路可反复修改重构。

3、工作速度高，可达数百兆。

4、设计、修改方便，借助计算机辅助，实现电路设计、模拟仿真等烦琐的工作。产品设计周期短。

5、保密性强。