第2章 计算机的逻辑部件

2.1 计算机中常用的组合逻辑电路

2.2 时序逻辑电路

2.3 阵列逻辑电路

主要知识点:

1、组合逻辑电路
2、时序逻辑电路
3、阵列逻辑电路

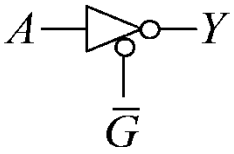
2.1 计算机中常用的组合逻辑电路

组合逻辑电路：逻辑电路的输出状态仅和当时的输入状态有关，而与过去的输入状态无关。

2.1.1 三态电路

功能表

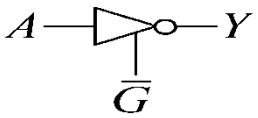
$\overline{G}$	$\overline{Y}$
0	A
1	Z



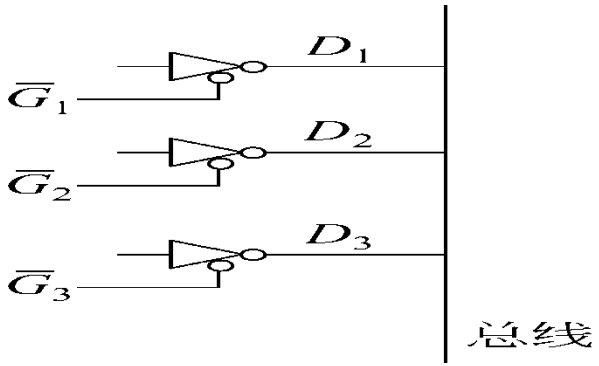
三态反相门

功能表

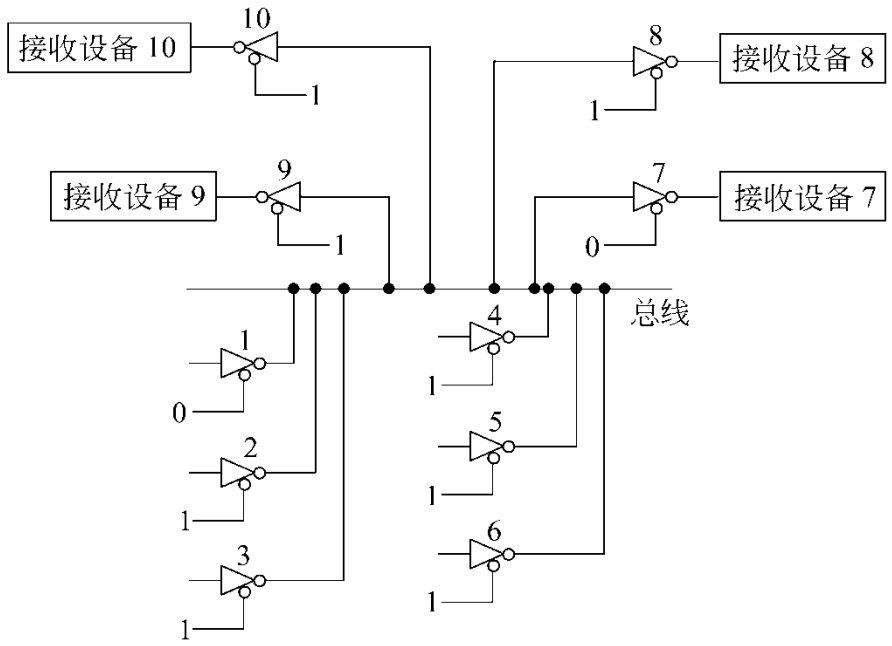
$\overline{G}$	$\overline{Y}$
0	Z
1	$\overline{A}$



三态反相门



三态门应用实例



三态电路应用实例

2.1.2 异或门及其应用

异或门的功能表和逻辑图

功能表

A_i	B_i	Y_i
0	0	0
1	0	1
0	1	1
1	1	0

异或门运算式:

1) $A\oplus A=1$

2) $A\oplus A=0$

3) $A\oplus 0=A$

4) $A\oplus 1=\overline{A}$

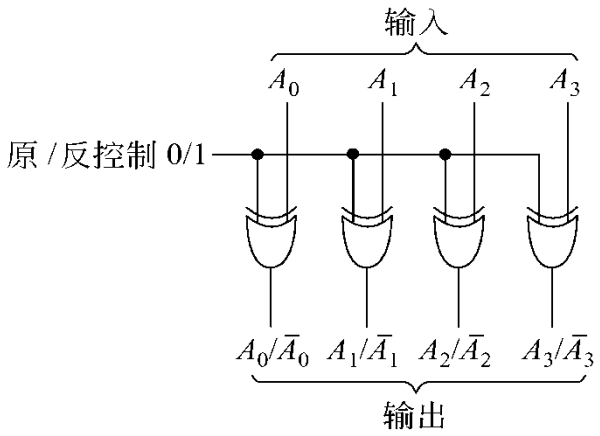
5) $A\oplus B=B\oplus A$

6) $A\oplus(B\oplus C)=(A\oplus B)\oplus C$

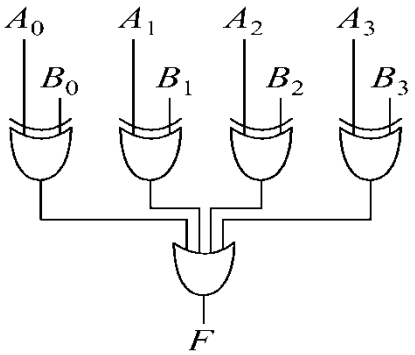
7) $A\cdot(B\oplus C)=(A\cdot B)\oplus(A\cdot C)$

异或门常用的作用:

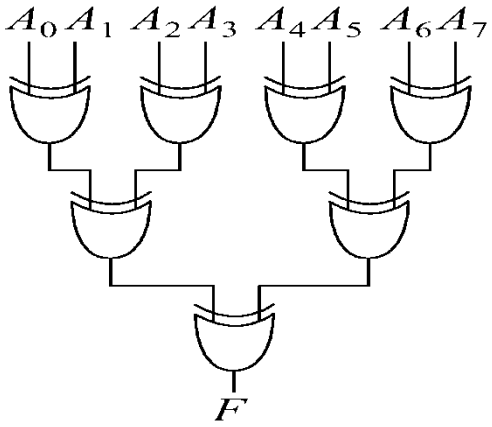
1、可控原码反码输出电路



2、数码比较器



3、奇偶检测电路



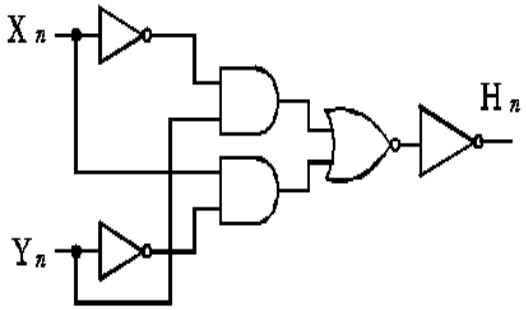
4、半加器

$$H_n=X_n\cdot\overline{Y_n}+\overline{X_n}\cdot Y_n=X_n\oplus Y_n$$

功能表

X_n	X_n	X_n
0	0	0
1	0	1
0	1	1
1	1	0

(a)



(b)

(c)

2.1.3全加器： X_n， Y_n及进位输入C_{n-1}相加称全加

$$F_n=\overline{X_n}\,\overline{Y_n}\,C_{n-1}+\overline{X_n}\,Y_n\,\overline{C_{n-1}}+X_n\,\overline{Y_n}\,\overline{C_{n-1}}+X_n\,Y_n\,C_{n-1}$$

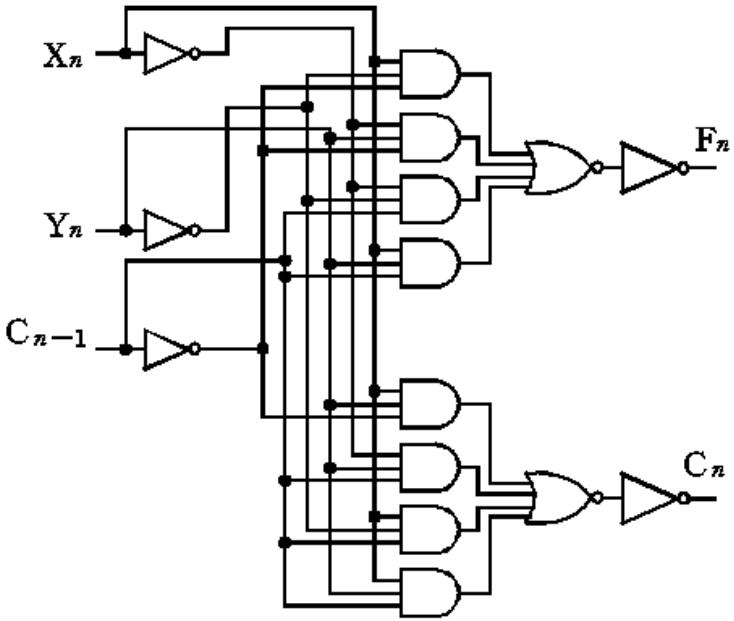
$$C_n=X_n\,Y_n\,\overline{C_{n-1}}+\overline{X_n}\,Y_n\,C_{n-1}+X_n\,\overline{Y_n}\,C_{n-1}+X_nY_nC_{n-1}$$

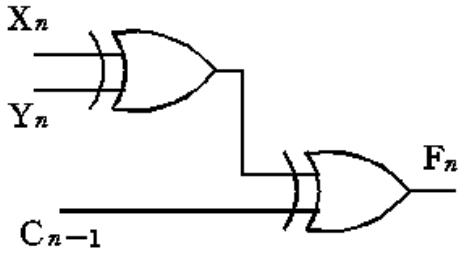
全加器：两个半加器来形成。F_n是X_n、Y_n相加再和C_{n-1}相加的结果：

$$F_n=X_n\oplus Y_n\oplus C_{n-1}$$

功能表

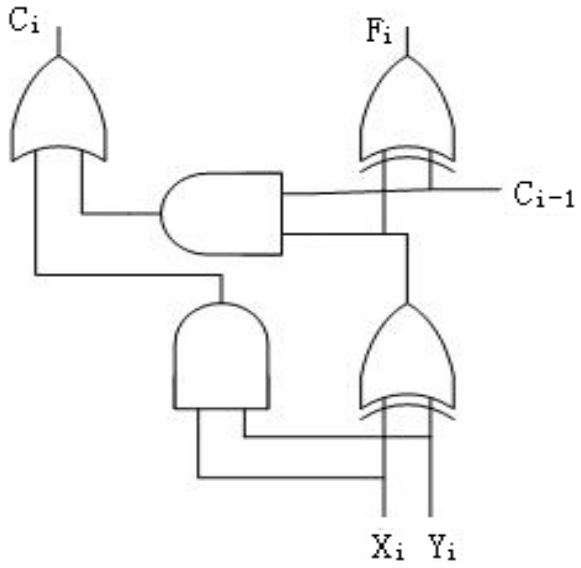
X_n	Y_n	C_{n-1}	F_n	C_n
0	0	0	0	0
0	0	1	1	0
1	0	0	1	0
1	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	1	0	0	1
1	1	1	1	1





（a）功能表　　（b）逻辑图　　（c）逻辑图

图2.6 全加器的功能表及逻辑图


C_i
F_i
C_{i-1}
X_i
Y_i

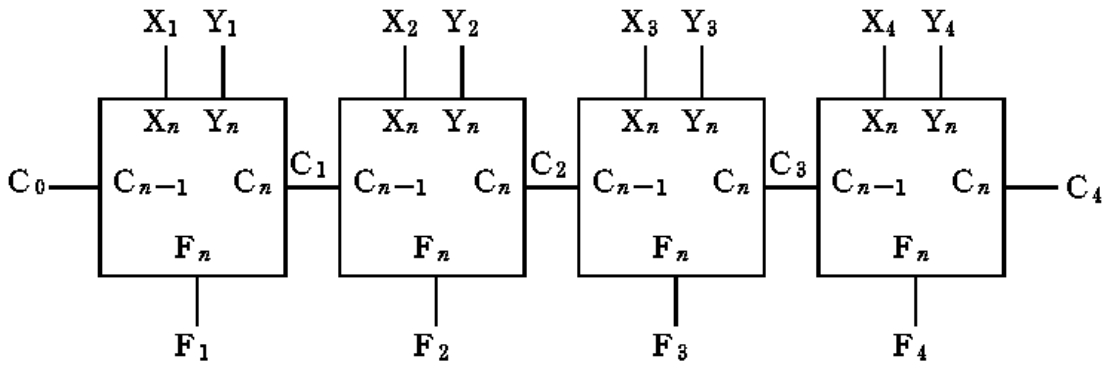
1、 当全加器的输出为Ci＝1且S＝1时。其输入为（ ）。（题中A，B为数据输入， Ci-1为进位输入，Ci为进位输出，S为和）

A：A=B=1, Ci-1=0　　B： A=B= Ci-1=1

C：A= Ci-1=1,B=0　　D： B= Ci-1=1,A=0

2、假设异或门延迟时间T=60ns，与门、或门延迟时间T=20ns，1位的全加器进位、和运算时间

四位串行加法器



$$C_1=X_1Y_1+(X_1+Y_1)C_0$$

$$C_2=X_2Y_2+(X_2+Y_2)X_1Y_1+(X_2+Y_2)(X_1+Y_1)C_0$$

$$C_3=X_3Y_3+(X_3+Y_3)X_2Y_2+(X_3+Y_3)(X_2+Y_2)X_1Y_1+(X_3+Y_3)(X_2+Y_2)(X_1+Y_1)C_0$$

$$C_4=X_4Y_4+(X_4+Y_4)X_3Y_3+(X_4+Y_4)(X_3+Y_3)X_2Y_2+(X_4+Y_4)(X_3+Y_3)(X_2+Y_2)X_1Y_1+(X_4+Y_4)(X_3+Y_3)(X_2+Y_2)(X_1+Y_1)C_0$$

进位传递函数P_i

$$P_i=X_i+Y_i$$

进位产生函数G_i

$$G_i=X_i\cdot Y_i$$

- P_1的意义是：当X_1，Y_1中有一个为“1”时，若有进位输入，则本位向高位传送进位，这个进位可看成是低位进位越过本位直接向高位传递的。
  - G_1的意义是：当X_1，Y_1均为“1”时，不管有无进位输入，定会产生向高位的进位。

串行进位方式

$$C_1 = G_1 + P_1C_0$$

$$C_2 = G_2 + P_2C_1$$

$$C_3 = G_3 + P_3C_2$$

$$C_4 = G_4 + P_4C_3$$

进位链：进位信号的产生与传递的逻辑结构

将n个全加器相连可得n位加法器，但其加法时间较长。这是因为其位间进位是串行传送的，本位全加和F_i必须等低位进位C_{i-1}来到后才能进行，加法时间与位数有关。只有改变进位逐位传送的路径，才能提高加法器工作速度。

超前进位（先行进位）基本思想：让高位的进位与低位进位无关，仅与两个参加的操作数有关

超前进位加法器（并行）：采用“超前进位产生电路”来同时形成各位进位，从而实现快速加法

4位并行加法器

组内并行的进位方式：

将P_1、G_1代入$C_1 \sim C_4$式，便可得：

$C_1=G_1+P_1C_0$

$C_2=G_2+P_2G_1+P_2P_1C_0$

$C_3=G_3+P_3G_2+P_3P_2G_1+P_3P_2P_1C_0$

$C_4=G_4+P_4G_3+P_4P_3G_2+P_4P_3P_2G_1+P_4P_3P_2P_1C_0$

$$C_1=P_1+G_1C_0$$

$$C_2=P_2+G_2P_1+G_2G_1C_0$$

$$C_3=P_3+G_3P_2+G_3G_2P_1+G_3G_2G_1C_0$$

$$C_4=P_4+G_4P_3+G_4G_3P_2+G_4G_3G_2P_1+G_4G_3G_2G_1C_0$$

由P_i、G_i定义，也可把半加和改写成以下形式：

$$H_i=P_i\oplus G_i$$

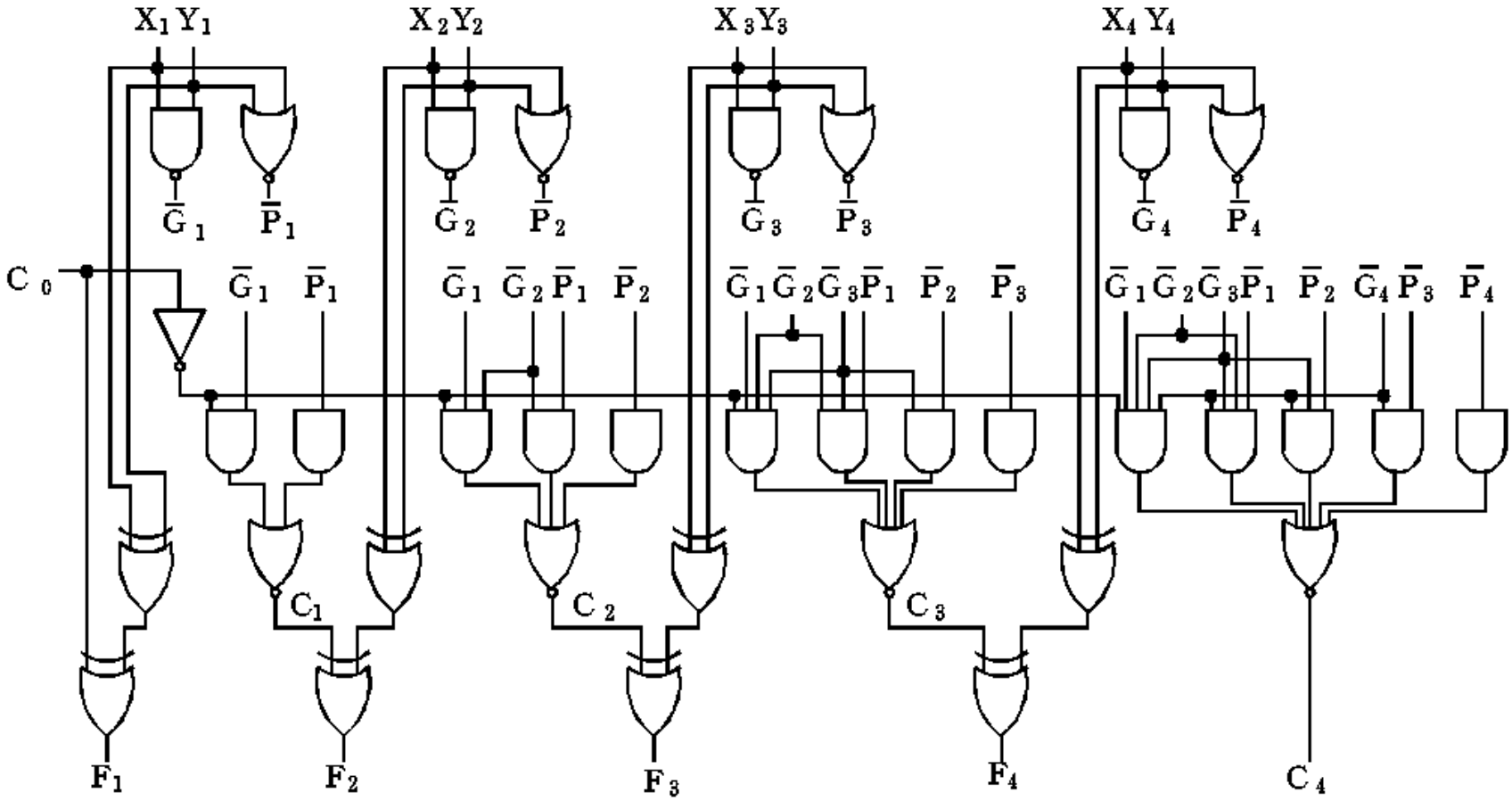


图2.8 四位超前进位加法器

16位加法器

并行加法器及其进位链

（1）组内并行，组间串行的进位链

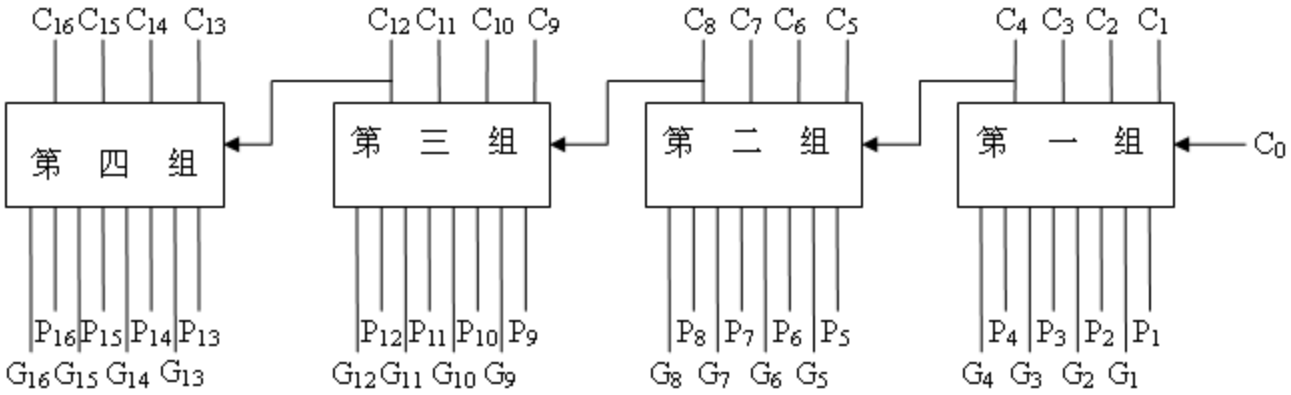
这种进位链每小组4位，组内采用并行进位结构，组间采用串行进位传递结构。进位表达式为：

$$C_1 = G_1 + P_1C_0$$

$$C_2 = G_2 + P_2G_1 + P_2P_1C_0$$

$$C_3 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1C_0$$

$$C_4 = G_4 + P_4G_3 + P_4P_3G_2 + P_4P_3P_2G_1 + P_4P_3P_2P_1C_0$$


C16 C15 C14 C13
C12 C11 C10 C9
C8 C7 C6 C5
C4 C3 C2 C1
第 四 组
第 三 组
第 二 组
第 一 组
C0
P16 P15 P14 P13
P12 P11 P10 P9
P8 P7 P6 P5
P4 P3 P2 P1
G16 G15 G14 G13
G12 G11 G10 G9
G8 G7 G6 G5
G4 G3 G2 G1

（2）组内并行、组间并行的进位链

$G_1^* = G_4 + P_4G_3 + P_4P_3G_2 + P_4P_3P_2G_1$　（小组的进位产生函数）

$P_1^* = P_4P_3P_2P_1$　（小组的进位传递函数）

因此

$C_4 = G_1^* + P_1^*C_0$

同理

$C_8 = G_2^* + P_2^*C_4$

$C_{12} = G_3^* + P_3^*C_8$

$C_{16} = G_4^* + P_4^*C_{12}$

将其展开为：

$C_4 = G_1^* + P_1^*C_0$

$C_8 = G_2^* + P_2^* G_1^* + P_2^*P_1^*C_0$

$C_{12} = G_3^* + P_3^* G_2^* + P_3^* P_2^* G_1^* + P_3^*P_2^*P_1^*C_0$

$C_{16} = G_4^* + P_4^*G_3^* + P_4^* P_3^* G_2^* + P_4^*P_3^* P_2^* G_1^* + P_4^*P_3^*P_2^*P_1^*C_0$

其中

$G_1^* = G_4 + P_4G_3 + P_4P_3G_2 + P_4P_3P_2G_1$

$G_2^* = G_8 + P_8G_7 + P_8P_7G_6 + P_8P_7P_6G_5$

$G_3^* = G_{12} + P_{12}G_{11} + P_{12}P_{11}G_{10} + P_{12}P_{11}P_{10}G_9$

$G_4^* = G_{16} + P_{16}G_{15} + P_{16}P_{15}G_{14} + P_{16}P_{15}P_{14}G_{13}$

$P_1^* = P_4P_3P_2P_1$

$P_2^* = P_8P_7P_6P_5$

$P_3^* = P_{12}P_{11}P_{10}P_9$

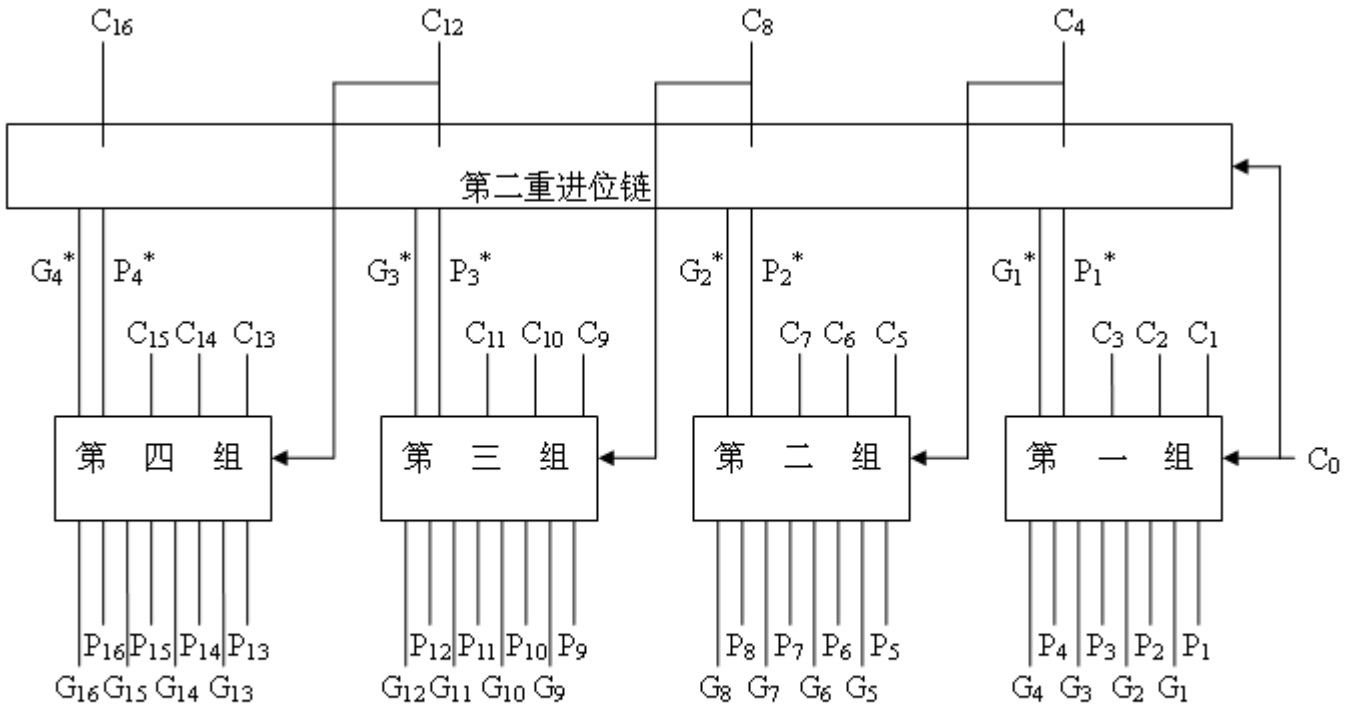
$P_4^* = P_{16}P_{15}P_{14}P_{13}$

进位产生次序：

① 产生第一组的C_1、C_2、C_3及所有的G_i^*、P_i^*；

② 产生组间的进位信号C_4、C_8、C_{12}、C_{16}；

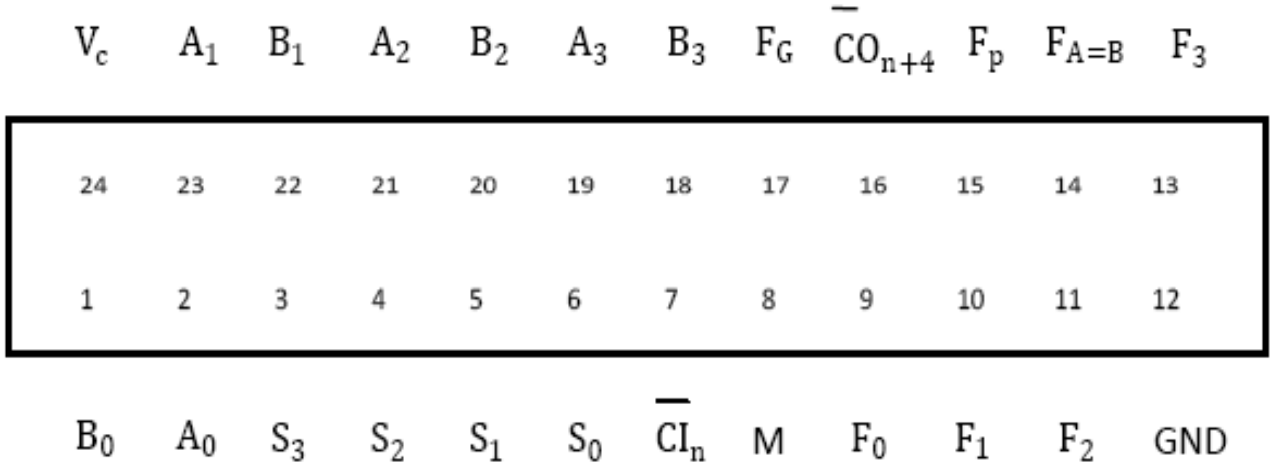
③ 产生第二、三、四小组的C_5、C_6、C_7，C_9、C_{10}、C_{11}，C_{13}、C_{14}、C_{15}；


C16
C12
C8
C4
第二重进位链
G4*
P4*
G3*
P3*
G2*
P2*
G1*
P1*
C15 C14 C13
C11 C10 C9
C7 C6 C5
C3 C2 C1
第 四 组
第 三 组
第 二 组
第 一 组
C0
P16 P15 P14 P13
G16 G15 G14 G13
P12 P11 P10 P9
G12 G11 G10 G9
P8 P7 P6 P5
G8 G7 G6 G5
P4 P3 P2 P1
G4 G3 G2 G1

2.4.2 算术逻辑单元

算术逻辑单元简称ALU,是一种功能较强的组合逻辑电路。

1、四位算术逻辑单元SN74181，执行***16种算术运算和16种逻辑运算***



A_0~A_3	运算器输入端
B_0~B_3	运算器输入端
$\overline{CI}_n$	进位输入端（低电平有效）
$\overline{CO}_{n+4}$	进位输出端（低电平有效）
F_0~F_3	运算输出端
$F_{A=B}$	比较输出端
F_G	进位产生输出端
F_p	进位传输输出端
M	工作方式控制
S_0~S_3	功能选择

功能表

高电平作用数据

输入				逻辑功能	算术运算 M = L	
S_3	S_2	S_1	S_0	M = H	$\overline{CI}_n$ = H （无进位）	$\overline{CI}_n$ = L （有进位）
L	L	L	L	$F = \overline{A}$	F = A	F = A 加 1
L	L	L	H	$F = \overline{A+B}$	F = A + B	F = （A + B）加 1
L	L	H	L	$F = \overline{A}B$	$F = A + \overline{B}$	F = （A + $\overline{B}$）加 1
L	L	H	H	F = 0	F = 减 1 （2 的补数）	F = 0
L	H	L	L	$F = \overline{AB}$	F = A 加 $A\overline{B}$	F = A 加 $A\overline{B}$ 加 1
L	H	L	H	$F = \overline{B}$	F = （A + B）加 $A\overline{B}$	F = （A + B）加 $A\overline{B}$ 加 1
L	H	H	L	$F = A \oplus B$	F = A 减 B 减 1	F = A 减 B
L	H	H	H	$F = A\overline{B}$	F = $A\overline{B}$ 减 1	$F = A\overline{B}$
H	L	L	L	$F = \overline{A} + B$	F = A 加 AB	F = A 加 AB 加 1
H	L	L	H	$F = \overline{A \oplus B}$	F = A 加 B	F = A 加 B 加 1
H	L	H	L	F = B	F = （A + $\overline{B}$）加 AB	F = （A + $\overline{B}$）加 AB 加 1
H	L	H	H	F = AB	F = AB 减 1	F = AB
H	H	L	L	F = 1	F = A 加 A*	F = A 加 A 加 1
H	H	L	H	$F = A + \overline{B}$	F = （A + B）加 A	F = （A + B）加 A 加 1
H	H	H	L	F = A + B	F = （A + $\overline{B}$）加 A	F = （A + $\overline{B}$）加 A 加 1
H	H	H	H	F = A	F = A 减 1	F = A

用4片74181电路可组成16位ALU。图中片内进位是快速的，但片间进位是逐片传递的，因此形成$F_0 \sim F_{15}$的时间还是比较长。

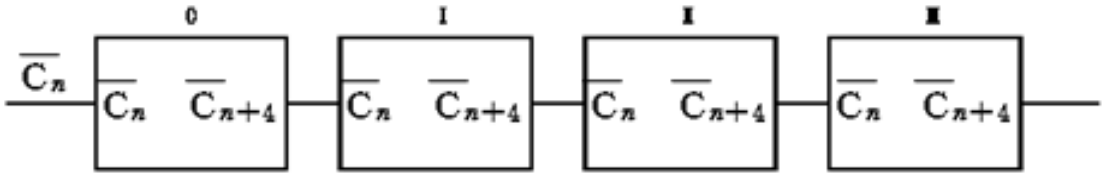
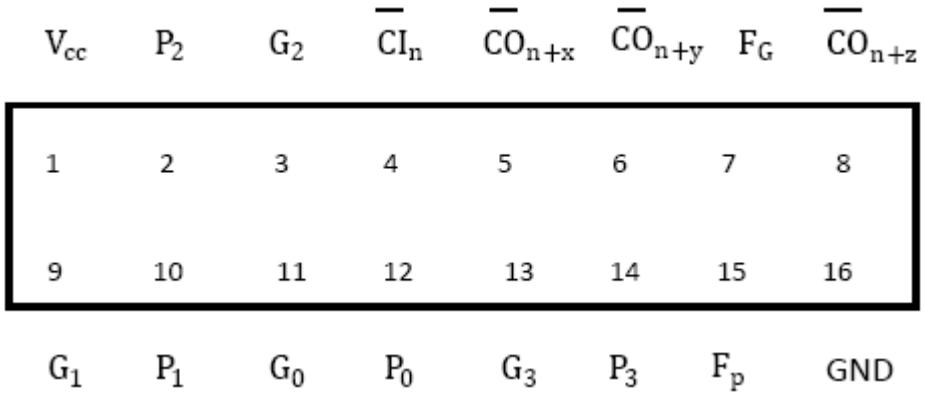


图2.10 用4片ALU构成的16位ALU

2、超前进位产生器74182



$\overline{CI}_n$ 进位输入端（低电平有效）

$\overline{CO}_{n+x}$ $\overline{CO}_{n+y}$ $\overline{CO}_{n+z}$ 进位输出端（低电平有效）

F_G 进位产生输出端

F_p 进位传输输出端

G_0~G_3 进位产生输入端

P_0~P_3 进位传输输入端

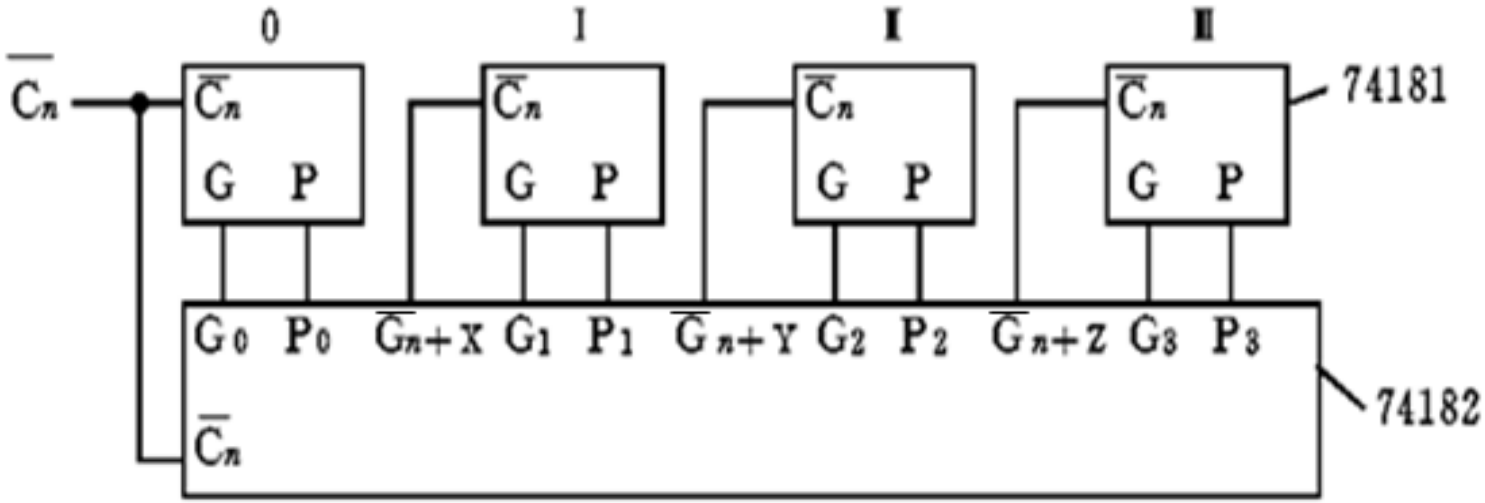
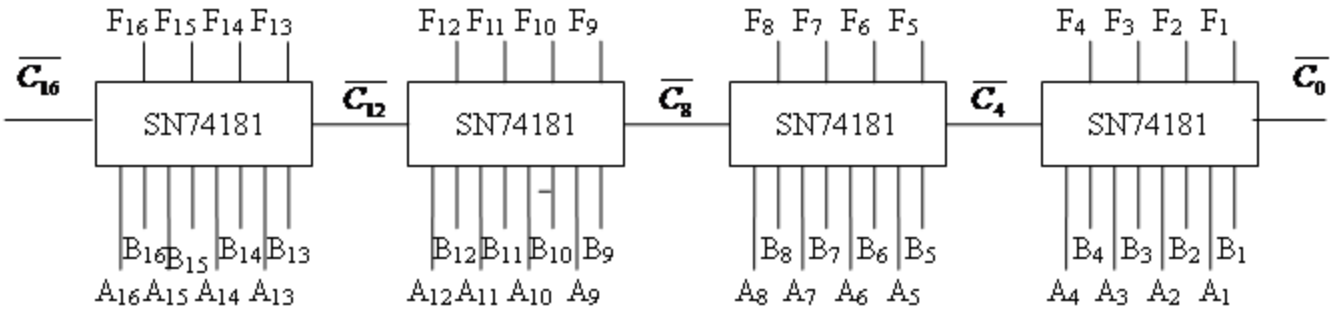


图2.12 16位快速ALU

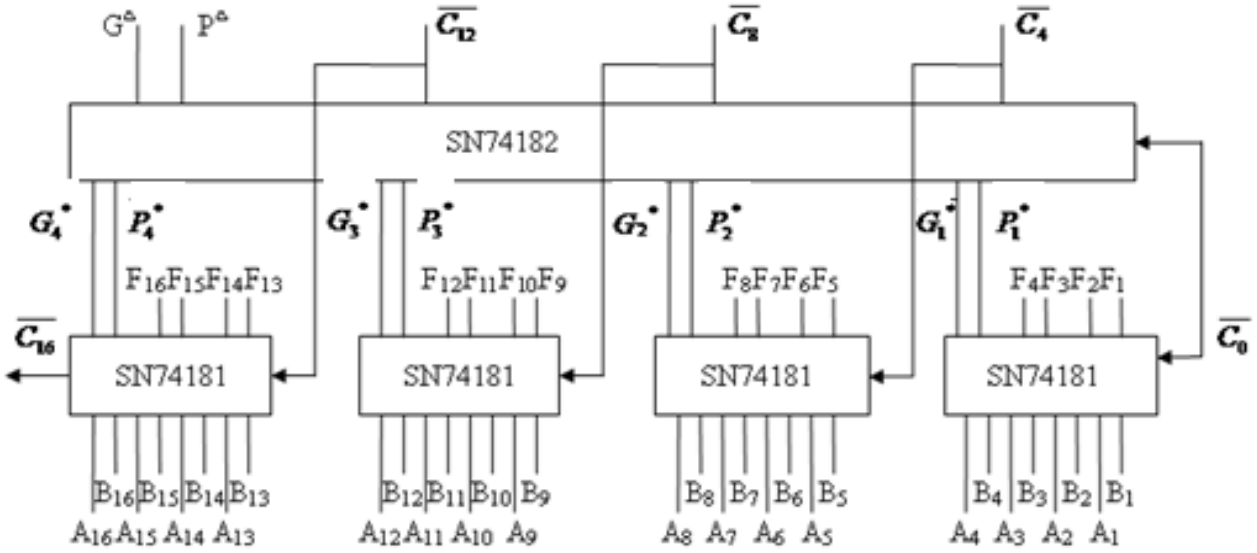
SN74181是一种具有并行进位的多功能ALU芯片，每片4位，构成一组，组内是并行进位，利用SN74181芯片可构成16位ALU。

① 组间串行进位的16位ALU的构成



② 组间并行进位的16位ALU的构成

组间采用并行进位时，则只需增加一片SN74182芯片。SN74182是与SN74181配套的产品，是一个产生并行进位信号的部件。组间进行进位的16位ALU。



作业:

1、若计算机系统字长为16位，每4位构成一个小组。（1）写出实现小组内并行，组间串行的进位链；（2）写出实现小组内并行，组间并行的进位链

2、用74181和74182构成16的ALU写出逻辑电路图

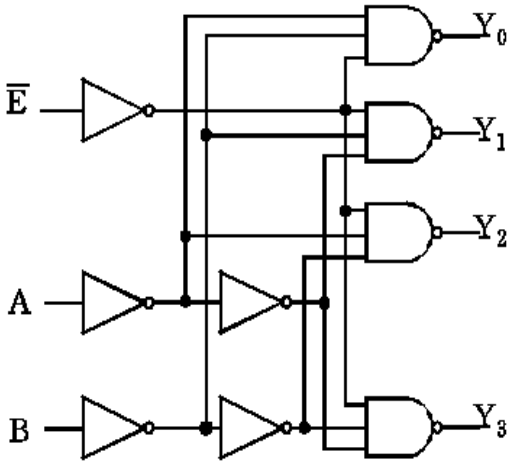
（1）实现组内并行，组间串行进位方式

（2）实现组内并行，组间并行进位方式

2.4.3 译码器

- **译码器**有**n个**输入变量，**2^n个**(或少于2^n个)输出，每个输出对应于n个输入变量的一个最小项。当输入为某一组合时，对应的仅有一个输出为“0”(或为“1”)，其余输出均为“1”(或为“0”)。

- **译码器的用途**是把输入代码译成相应的控制电位，以实现代码所要求的操作。



功能表

$\overline{E}$	A	B	Y_0	Y_1	Y_2	Y_3
0	0	0	0	1	1	1
0	1	0	1	0	1	1
0	0	1	1	1	0	1
0	1	1	1	1	1	0
1	×	×	1	1	1	1

图2.13 二输入四输出译码器

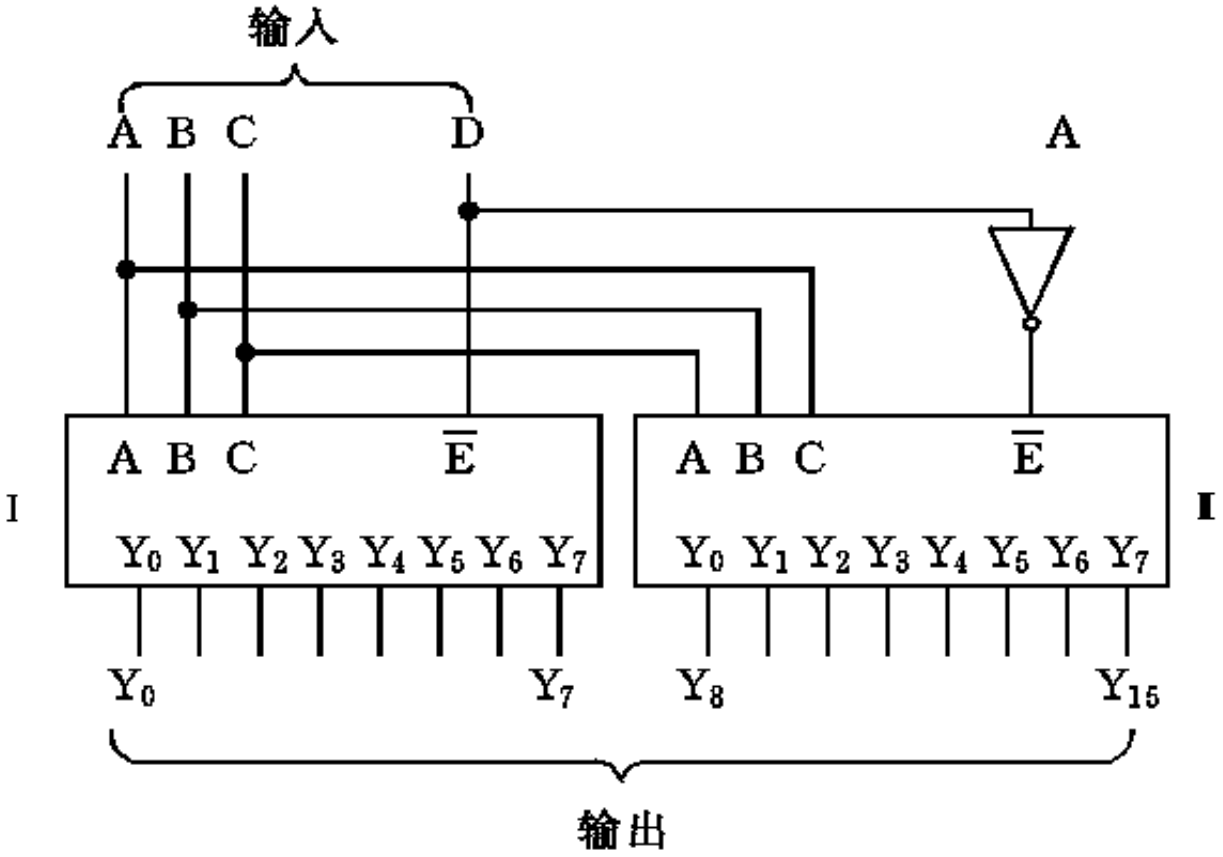


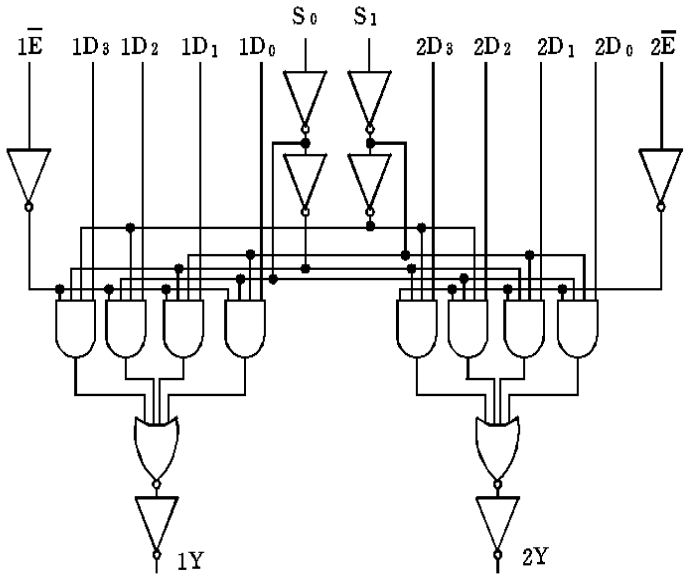
图2.14 两块三输入变量译码器扩展成四输入译码器

2.4.4 数据选择器

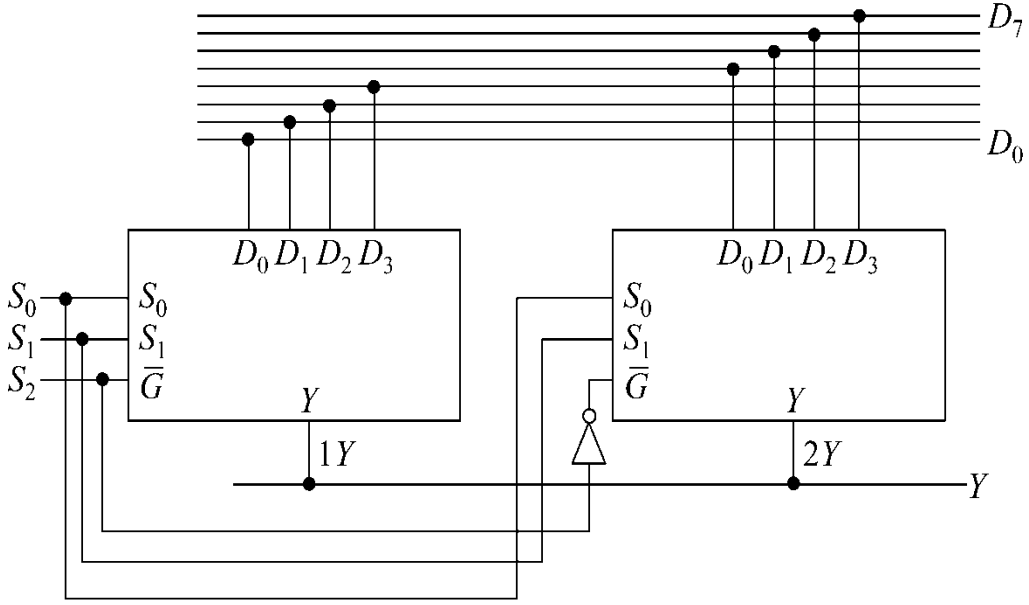
•**数据选择器又称多路开关**，是以“与或”门或“与或非”门为主的电路。它能在选择信号的作用下，从多个输入通道中选择某一个通道的数据作为输出。

功能表

S_1	S_0	D_3	D_2	D_1	D_0	$\overline{E}$	Y
×	×	×	×	×	×	1	0
1	1	D_3	×	×	×	0	D_3
1	0	×	D_2	×	×	0	D_2
0	1	×	×	D_1	×	0	D_1
0	0	×	×	×	D_0	0	D_0



双四通道选一数据选择器


D_7
D_0
$D_0\,D_1\,D_2\,D_3$
$D_0\,D_1\,D_2\,D_3$
S_0
S_1
S_2
S_0
S_1
$\overline{G}$
Y
$1Y$
S_0
S_1
$\overline{G}$
Y
$2Y$
Y

8选1数据选择器

2.5 时序逻辑电路

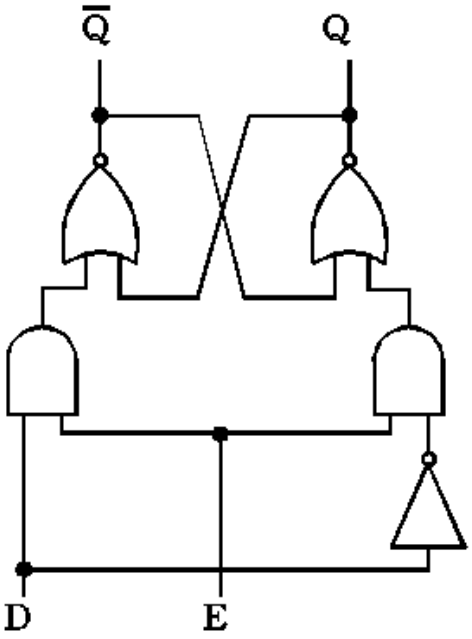
时序逻辑电路:逻辑电路的输出状态不但和**当时的输入状态**有关，而且还与电路**在此以前的输入状态**有关。时序电路内必须要有能存储信息的记忆元件——**触发器**。**触发器**是构成时序电路的基础。

2.5.1 触发器

- **按时钟控制方式来分:**电位触发、边沿触发、主从触发等方式。
- **按功能分:**，R-S型、D型、J-K型等功能。
- **同一功能触发器可以由不同触发方式来实现。**
- **对使用者来说，在选用触发器时，触发方式是必须考虑的因素。**因为相同功能触发器，若触发方式选用不当，系统是不能达到预期设计要求的。

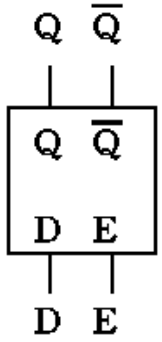
1. 电位触发方式触发器

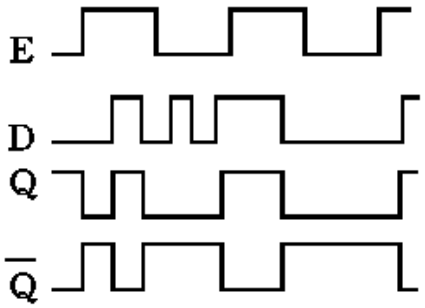
- 当触发器的同步控制信号E为约定“1”或“0”电平时，触发器接收输入数据，此时输入数据D的任何变化都会在输出Q端得到反映；当E为非约定电平时，触发器状态保持不变。鉴于它接收信息的条件是E出现约定的逻辑电平，故称它为电位触发方式触发器，简称**电位触发器。**
- 图2.16给出了被称为**锁定触发器(又称锁存器)**的电位触发器的逻辑图。
- 电位触发器具有结构简单的优点。在计算机中常用它来组成**暂存器。**



功能表

E	D	Q	$\overline{Q}$
1	0	0	1
1	1	1	0
0	×	Q_0	$\overline{Q}_0$





(a) 逻辑图和功能表　　(b) 图形符号　　(c) 典型波形图

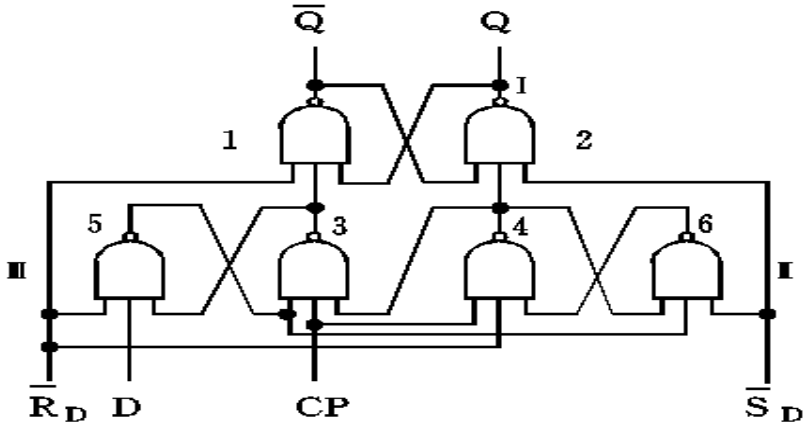
图2.16 锁存器

2. 边沿触发方式触发器

•边沿触发方式触发器(简称边沿触发器):触发器接收的是时钟脉冲CP的某一约定跳变(正跳变或负跳变)来到时的输入数据。在CP=1及CP=0期间以及CP非约定跳变到来时，触发器不接收数据。

•常用的正边沿触发器是D触发器，图2.17给出了它的逻辑图及典型波形图。

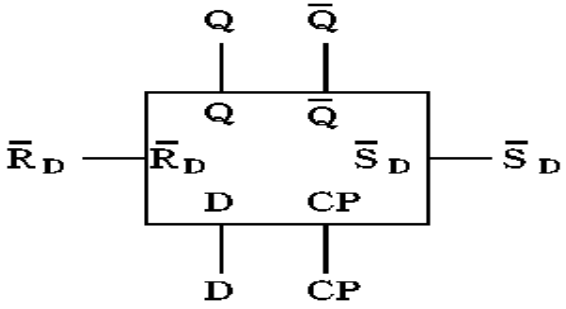
•边沿触发器在CP正跳变(对正边沿触发器)以外期间出现在D端的数据变化和干扰不会被接收，因此有很强的抗数据端干扰的能力而被广泛应用，它除用来组成寄存器外，还可用来**组成计数器和移位寄存器等。**



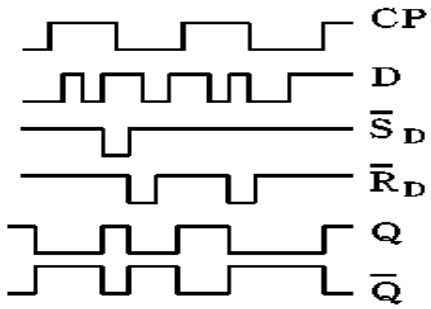
功能表

$\overline{R}_D$	$\overline{S}_D$	CP	D	Q	$\overline{Q}$
0	1	×	×	0	1
1	0	×	×	1	0
1	1	↑	0	0	1
1	1	↑	1	1	0

（a）*D*触发器逻辑图



（b）*D*触发器图形符号

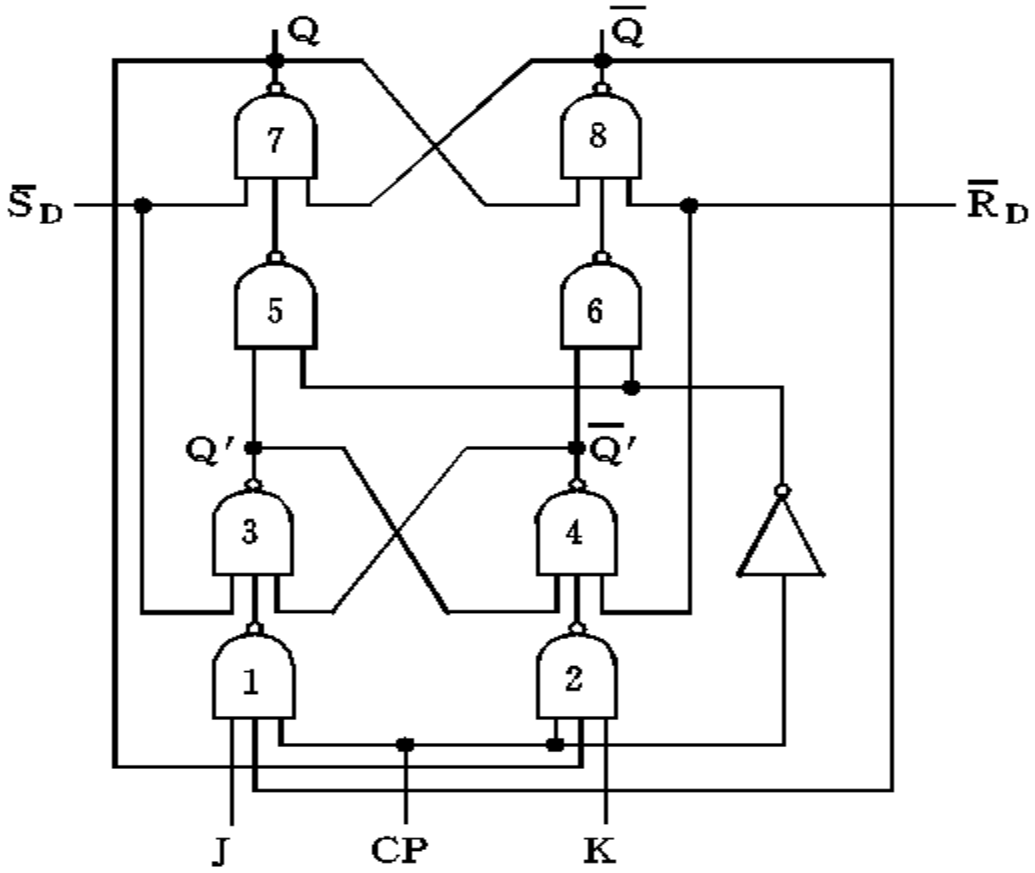


（c）波形图

图2.17 D触发器

3. 主-从触发方式触发器(简称主-从触发器)

- **主-从触发器基本上是由两个电位触发器级联而成的**，接收输入数据的是主触发器，接收主触发器输出的是从触发器，主、从触发器的同步控制信号是互补的(CP和$\overline{CP}$)。
- **图2.18(a)是主-从J-K触发器的原理图，**触发器的输出Q，$\overline{Q}$分别和接收K，J数据的输入门相连。在CP＝1期间主触发器接收数据；在CP负跳变来到时，从触发器接收主触发器最终的状态。
- **主从触发器由于有计数功能，常用于组成计数器。**



(a)

功能表

$\overline{R}_D$	$\overline{S}_D$	CP	J	K	Q	$\overline{Q}$
0	1	×	×	×	0	1
1	0	×	×	×	1	0
0	0	×	×	×	1*	1*
1	1	⊓	0	0	Q_0	$\overline{Q}_0$
1	1	⊓	1	0	1	0
1	1	⊓	0	1	0	1
1	1	⊓	1	1	$\overline{Q}_0$	Q_0

* 指状态不定

(b)

图2.18 主-从J-K触发器图

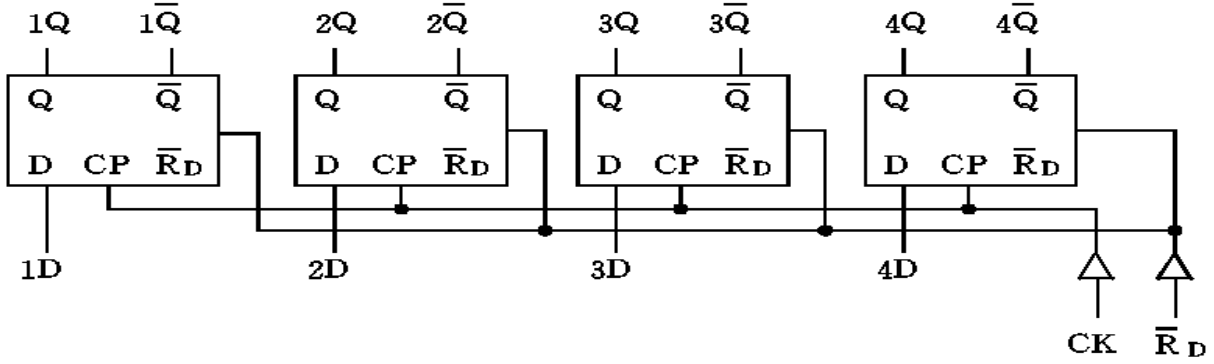
2.5.2 寄存器和移位寄存器

•寄存器用于**暂存数据、指令等**。它**由触发器和一些控制门组成**。常用的是正边沿触发D触发器和锁存器。

•移位寄存器:具有移位功能。如在进行乘法时，要求将部分积右移；在将并行传送的数转换成串行数时也需移位。

功能表

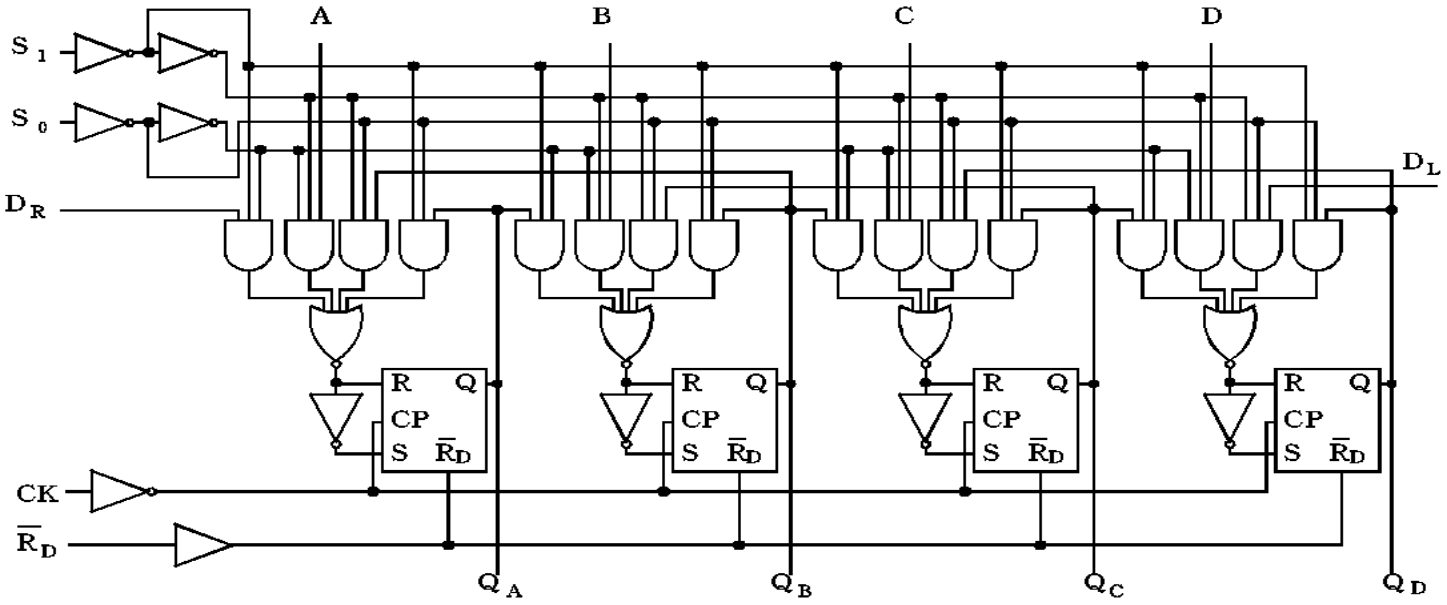
$\overline{R}_D$	CK	1D	2D	3D	4D	1Q	2Q	3Q	4Q
1	↑	1D	2D	3D	4D	1D	2D	3D	4D
0	×	×	×	×	×	0	0	0	0



四D寄存器

功能表

$\overline{R}_D$	S_0	S_1	CK	功　能
0	×	×	×	置“0”
1	0	0	↑	保　持
1	1	0	↑	右　移
1	0	1	↑	左　移
1	1	1	↑	并行输入



并行输入数据的四位移位寄存器

2.5.3 计数器

- 计数器按**时钟作用方式来分**，有**同步计数器**和**异步计数器**两大类。
- **异步计数器：**高位触发器的时钟信号是由低一位触发器的输出来提供的，结构简单。
- **同步计数器：**触发器的时钟信号是由同一脉冲来提供的，各触发器是同时翻转的，它的工作频率比异步计数器高，但结构较复杂。
- 计数器按**计数顺序来分**，有**二进制、十进制**两大类。
- 这里着重介绍有**并行输入数据功能的正向同步十进制计数器**。

功能表

P	T	L	$\overline{R}_D$	CK	功　　能
1	1	1	1	⎍	计　　数
×	×	0	1	⎍	并行输入数据
0	1	1	1	×	保　　持
×	0	1	1	×	触发器保持，RC＝0
×	×	×	0	×	异步清“0”

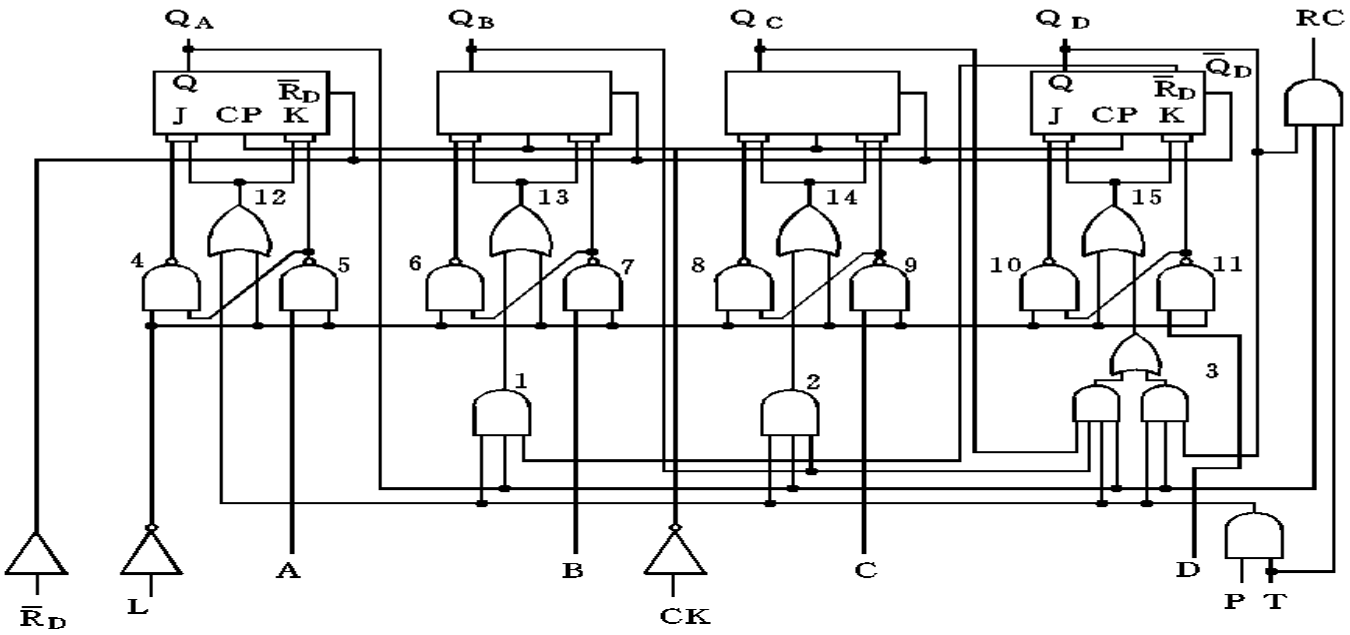


图2.23 十进制同步计数器

当L=1，执行同步计数

$$\begin{cases} J_A=K_A=1 \\ J_B=K_B=Q_A\cdot\overline{Q_D} \\ J_C=K_C=Q_A\cdot Q_B \\ J_D=K_D=Q_A\cdot Q_B\cdot Q_C+Q_A\cdot Q_D \end{cases}$$

L=0，执行预置数。

计数器扩展应满足以下条件:

（1）有标志计数器已计至最大数的进位输出端RC

二进制计数器： $RC=Q_AQ_BQ_CQ_D$

十进制计数器： $RC=Q_AQ_D$

（2）计数器应有保持功能。计数器中设置了“计数允许”端P和T，用来控制计数器快速进位电路和RC形成门。有了RC，P，T端，就可以方便地对计数器进行扩展。

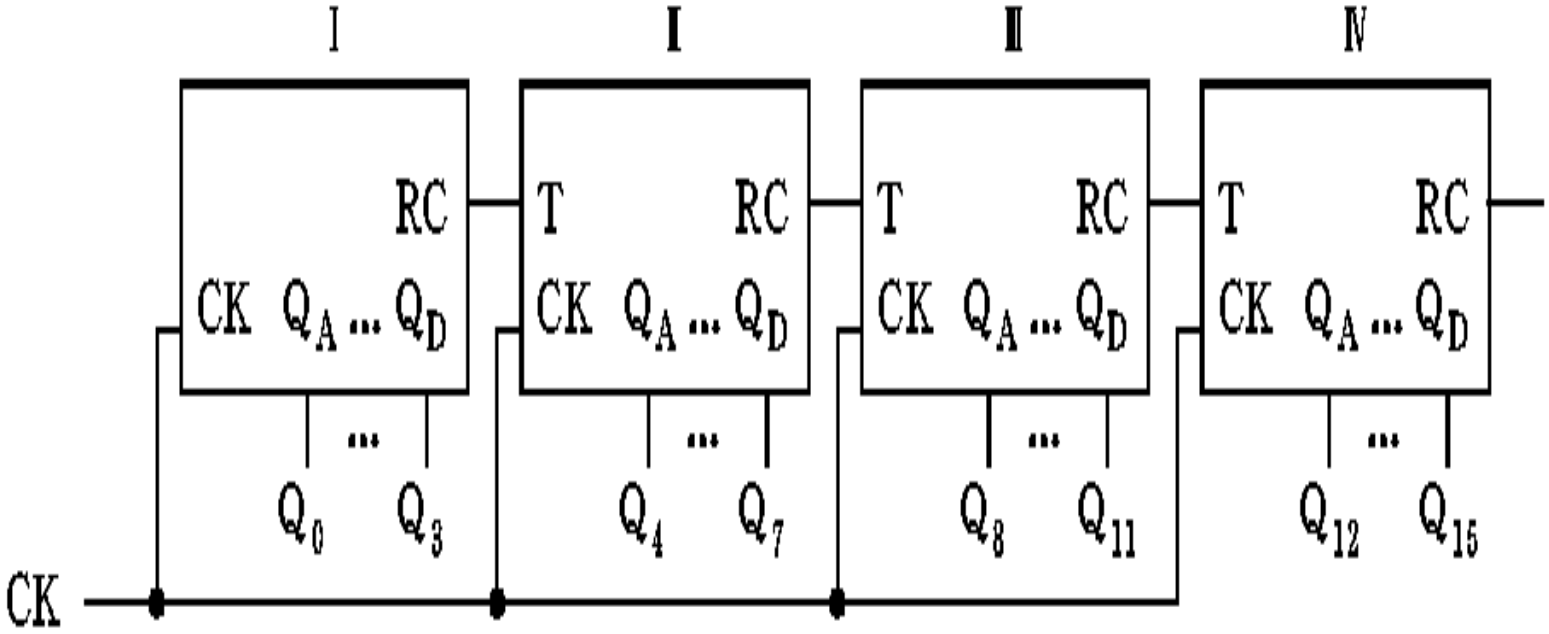


图2.24 同步计数器的扩展方法

2.6 阵列逻辑电路

- **阵列逻辑电路**:指逻辑元件在硅芯片上以阵列形式排列

优点:设计方便、芯片面积小、产品成品率高、用户自编程、减少系统的硬件规模。

- **读 / 写存储器(random access memory,简称RAM)**:存储单元排列成阵列形式。RAM在使用时能按给定的单元地址把信息存入或取出。

- **只读存储器(read only memory,简称ROM)**:存储固定的信息(如监控程序、函数、常数等)。在使用前把信息存入其中，使用时读出已存入的信息，而不能写入新的信息。ROM主要由**全译码的地址译码器**和**存储单元体组成**，前者是一种**“与”阵列**，后者则是**“或”阵列**，它们都以阵列形式排列。存储体中写入的信息是由用户事先决定的，因此是**“用户可编程”的**，而地址译码器则是**“用户不可编程”的。**

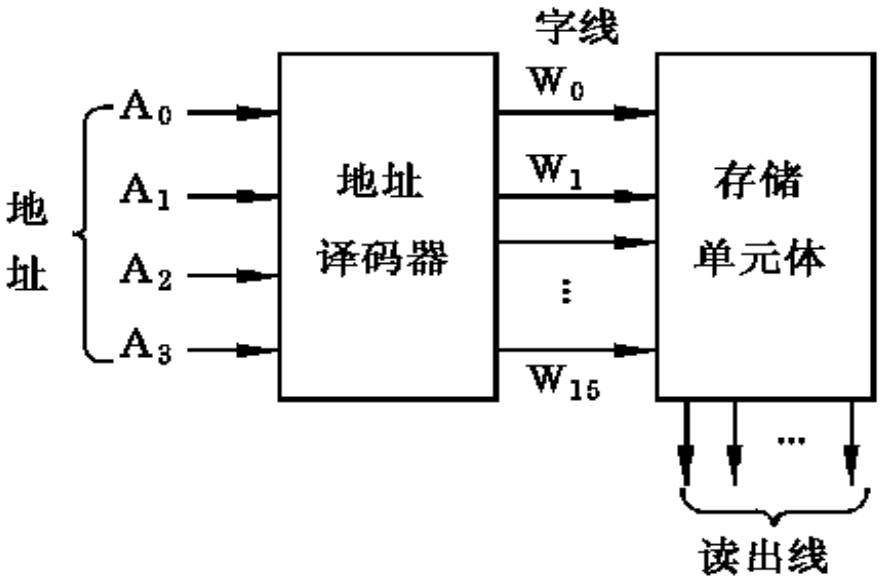
- **可编程序逻辑阵列**(programmable logic array，简称PLA):一种新型的ROM。与ROM不同之处是**PLA的与阵列、或阵列都是用户可编程的。PLA在组成控制器、存储固定函数以及实现随机逻辑**中有广泛的应用。

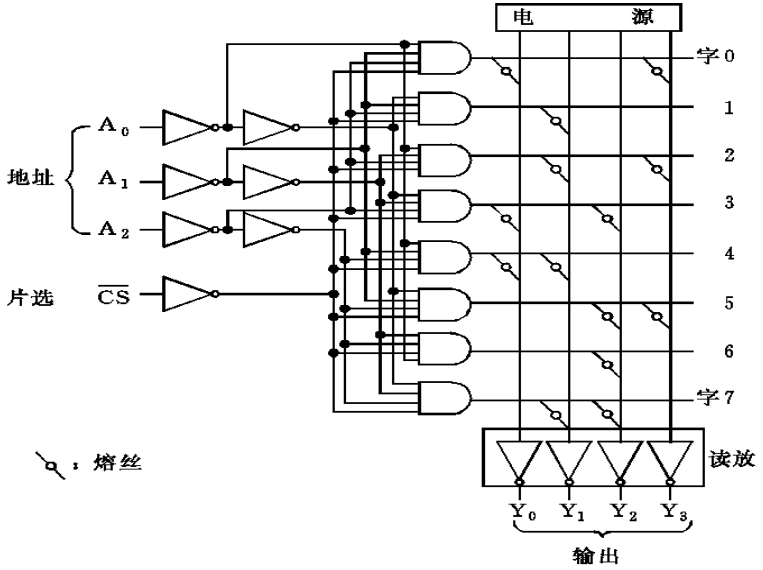
可编程序阵列逻辑(programmable array logic,简称PAL)也是ROM的变种，与ROM不同处是PAL的**与阵列是用户可编程的**，而**或阵列是用户不可编程的。**

- **通用阵列逻辑**(general array logic,简称GAL):输出有一个**逻辑宏单元**，通过对它的编程，可以获得多种输出形式。
- **门阵列**(gate array，简称GA)：在芯片上制作了排成阵列形式的**门电路**，根据用户需要对门阵列中的门电路进行互连设计，再通过集成电路制作工艺来实现互连，以实现所需的逻辑功能。
- **宏单元阵列**(macrocell array,简称MA)：芯片上排列成阵列的除门电路外还有**触发器、加法器、寄存器以及ALU等**。
- **可编程门阵列**(field programmable gate array，FPGA)：与GA，MA的一个区别在于，FPGA内部按阵列分布的**宏单元块**都是**用户可编程**的。即用户所需逻辑可在软件支持下，由用户自己装入来实现的，而无需集成电路制造工厂介入，并且这种装入是可以修改的，其连接十分灵活。
- **一般把除读／写存储器的阵列逻辑电路统称为可编程序逻辑器件**

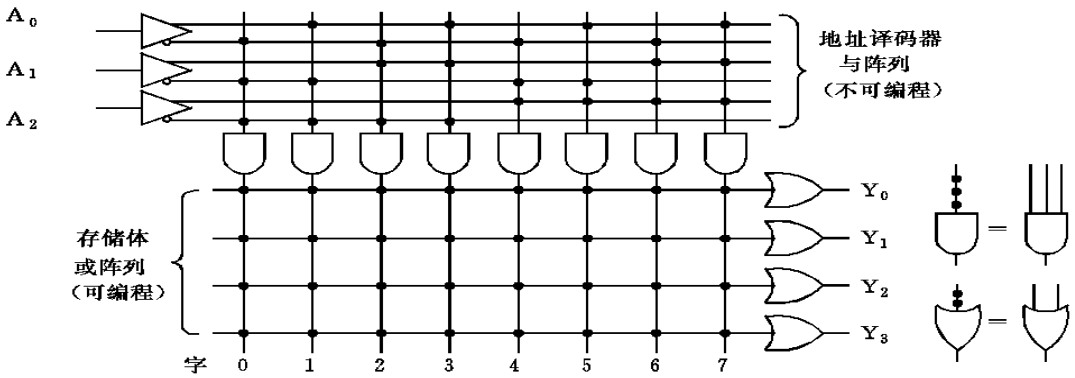
2.6.1 只读存储器（ROM）

存储器中存放信息的单元是**存储单元**，它是由若干个二进制信息组成的，叫做**“字”**，每个二进制信息称为**“位”**。为了寻找存入存储器中的字，给每个字以编号，称为地址码，简称**地址**。





（a）熔丝型8×4ROM原理图



（b）ROM结构的另一种表示形式

图2.26

2.6.2 可编程序逻辑阵列(PLA)

- 当用户要存入ROM的**字数少于ROM所能提供的字数**时，ROM中有许多存储单元便会闲置不用，因而造成管芯面积的浪费。
- 在ROM中，**地址和字之间有一一对应关系**，对任何一个给定地址，只能读出一个字，因此，即使有若干个字的内容一样，也无法节省单元。
- **PLA是一种特殊的只读存储器，它较好地解决了ROM的上述缺点。**它用较少的存储单元就能存储大量的信息。

表2.1 一张信息表

输　入				输　出							
I_3	I_2	I_1	I_0	F_7	F_6	F_5	F_4	F_3	F_2	F_1	F_0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	1	0	0	1
0	1	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	1	1	1	0	0	1
0	1	1	0	0	0	0	0	0	1	0	0
0	1	1	1	0	0	1	1	0	0	0	1
1	0	0	0	0	1	0	0	0	0	0	0
1	0	0	1	0	1	0	1	0	0	0	1
1	0	1	0	0	1	0	0	0	1	0	0
1	0	1	1	0	1	0	1	1	0	0	1
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	0	0	1	0	0	1
1	1	1	0	1	1	1	0	0	1	0	0
1	1	1	1	1	1	1	0	0	0	0	1

先把表2.1用逻辑表达式写出，进行化简，可得：

$$
\begin{cases}
F_0 = \times\times\times I_0 = P_0 \\
F_1 = 0 \\
F_2 = \times\times I_1 \bar{I}_0 = P_1 \\
F_3 = \times I_2 \bar{I}_1 I_0 + \times\ I_2 \bar{I}_1 I_0 = P_2 + P_3 \\
F_4 = \times I_2 \bar{I}_1 \bar{I}_0 + \quad I_3 I_2 \bar{} \times I_0 + I_3 I_2 \times \bar{I}_0 = P_4 + P_5 + P_6 \\
F_5 = \quad \bar{I}_3 I_2 \times I_0 + I_3 I_2 I_1 \times = P_5 + P_7 \\
F_6 = I_3 \bar{I}_2 \times\times + I_3 I_2 I_1 \times = P_7 + P_8 \\
F_7 = I_3 I_2 I_1 \times = P_7
\end{cases}
$$

(2.36)

其中×为任意值。P项称为乘积项，它们分别为：

$P_0 = \times\times\times I_0$　　$P_1 = \times\times I_1 \bar{I}_0$

$P_2 = \times I_2 \bar{I}_1 I_0$　　$P_3 = \times \bar{I}_2 I_1 I_0$

$P_4 = \times I_2 \bar{I}_1 \bar{I}_0$　　$P_5 = \bar{I}_3 I_2 \times I_0$

$P_6 = I_3 \bar{I}_2 \times I_0$　　$P_7 = I_3 I_2 I_1 \times$

$P_8 = I_3 \bar{I}_2 \times\times$

- 上半部分形成P项的二极管与阵列，构成译码电路，9条P线称为PLA的字线
- 下半部分形成输出F的三极管或阵列，构成存储阵列
- ROM矩阵容量16×8；PLA矩阵容量9×8

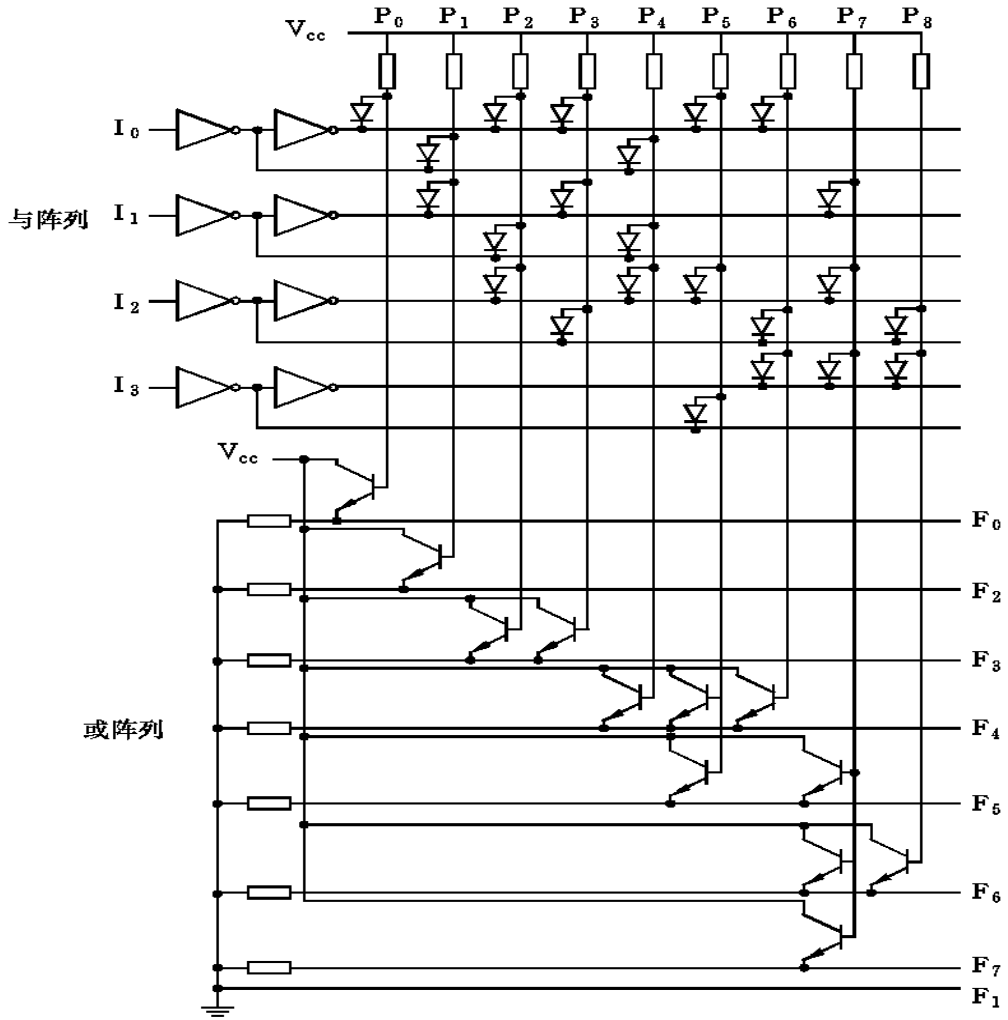


图2.27 存储表2.1所示信息表的PLA

PLA的读出过程如下：若$I_3I_2I_1I_0$=1001，则字线P_0，P_6，P_8均被选中，其余字线均未被选，再经存储矩阵，得F_0，F_4，F_6为“1”，其余输出均为“0”。

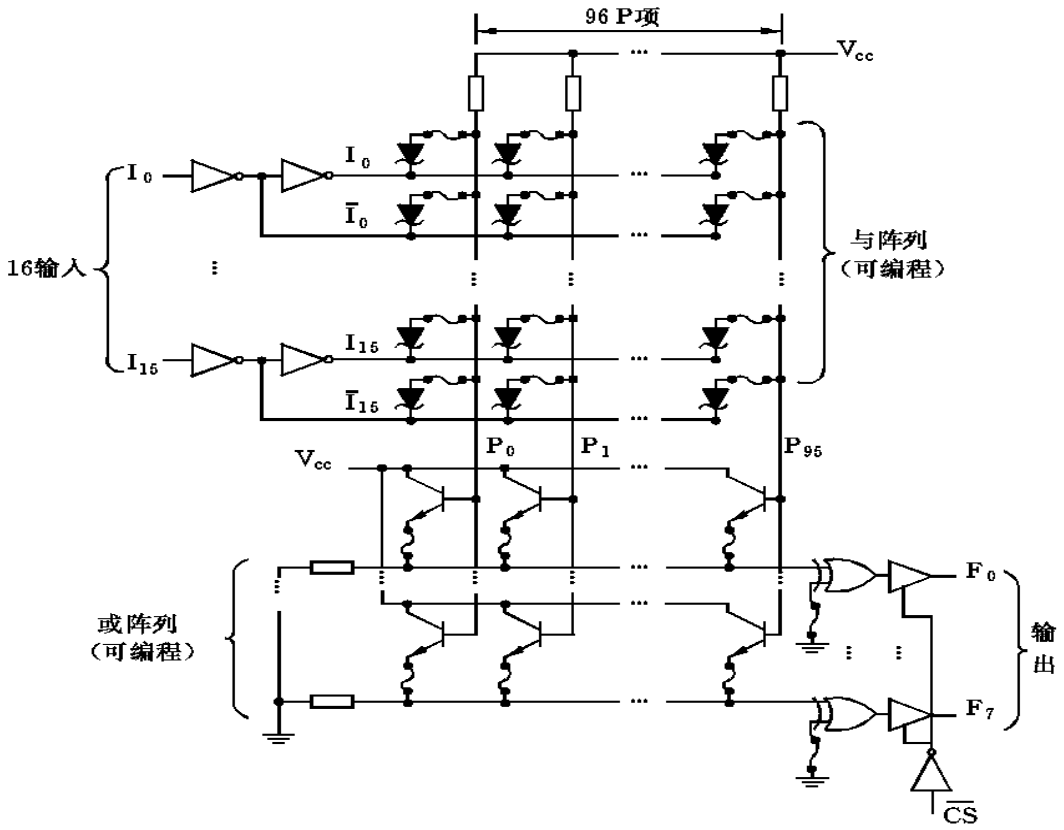
PLA的特点归结如下：

•在ROM中，地址译码器(与阵列)是**“完全”译码器**，它提供了输入的全部最小项，每个地址对应一个字，译码器是用户不可编程。**PLA**，虽然也有一个地址译码器(即与阵列)，但它是一个**非完全译码器，**它的输出不是输入变量的最小项，而是某些输入变量的乘积项，乘积项的个数小于(或等于)2^n。此外，这个译码器是用户可编程的。因此，PLA的与矩阵比ROM的与矩阵节省了很多元件。

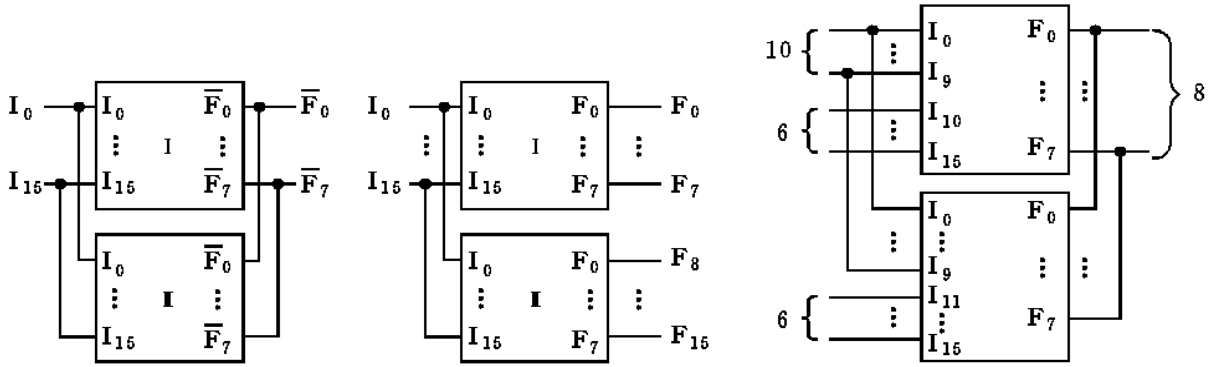
•在ROM中，**地址和字是一一对应的**，对于任一给定的地址，只能读出一个字。而在PLA中，**一个地址可以同时(即并行地)读出两个或两个以上字(即P项)**，在PLA的输出所得的是读出字的“或”。此外，**多个地址码能访问同一个P项**。这样，PLA就能用较少的单元存储较多的信息。

•在ROM中，**信息表**是原封不动地装入存储阵列中的。而在PLA中，存储信息不是原封不动地装入的，而是经过化简、压缩后装入的，它和信息表不再是简单的一一对应关系了。

16个输入端、8个输出端，96个乘积项，与阵列规模是32×96，或阵列规模是96×8



当一个PLA电路的P项数及输出端数不能满足要求时，可用几片PLA电路来扩展。



（a）PLA的P项扩展方法　（b）PLA的输出端扩展方法　（c）PLA的输入端扩展方法

例：图中每个PLA只有48个P项，若使用中要求某输出含有96个P项：

$Fi=P0+P1+\ldots+P47+P48+\ldots+P95$

熔断异或门的熔丝，

片I的Fi　$Fi=P0+P1+\ldots+P47$

片II的Fi::　$Fi=P48+P49+\ldots+P95$

片I、II的Fi端“线与”在一起，Fi的反码:

$\overline{Fi=P0+P1+\ldots+P47+P48+\ldots+P95}$

2.6.3 可编程序阵列逻辑(PAL)

PAL的**与阵列是可编程的，或阵列是不可编程的**。在某些PAL器件中还设置记忆元件，还可具有反馈功能，即输出可反馈到输入端，作为输入信号使用。

- 图2.31(a) 给出了**带触发器具有反馈功能的PAL电路**。8输入，8输出，64个P项，或门的连接是固定的，不可编程；
- 图2.31(b)**给出了不带触发器并具有反馈功能的PAL电路**。三态门的控制端由P项来控制

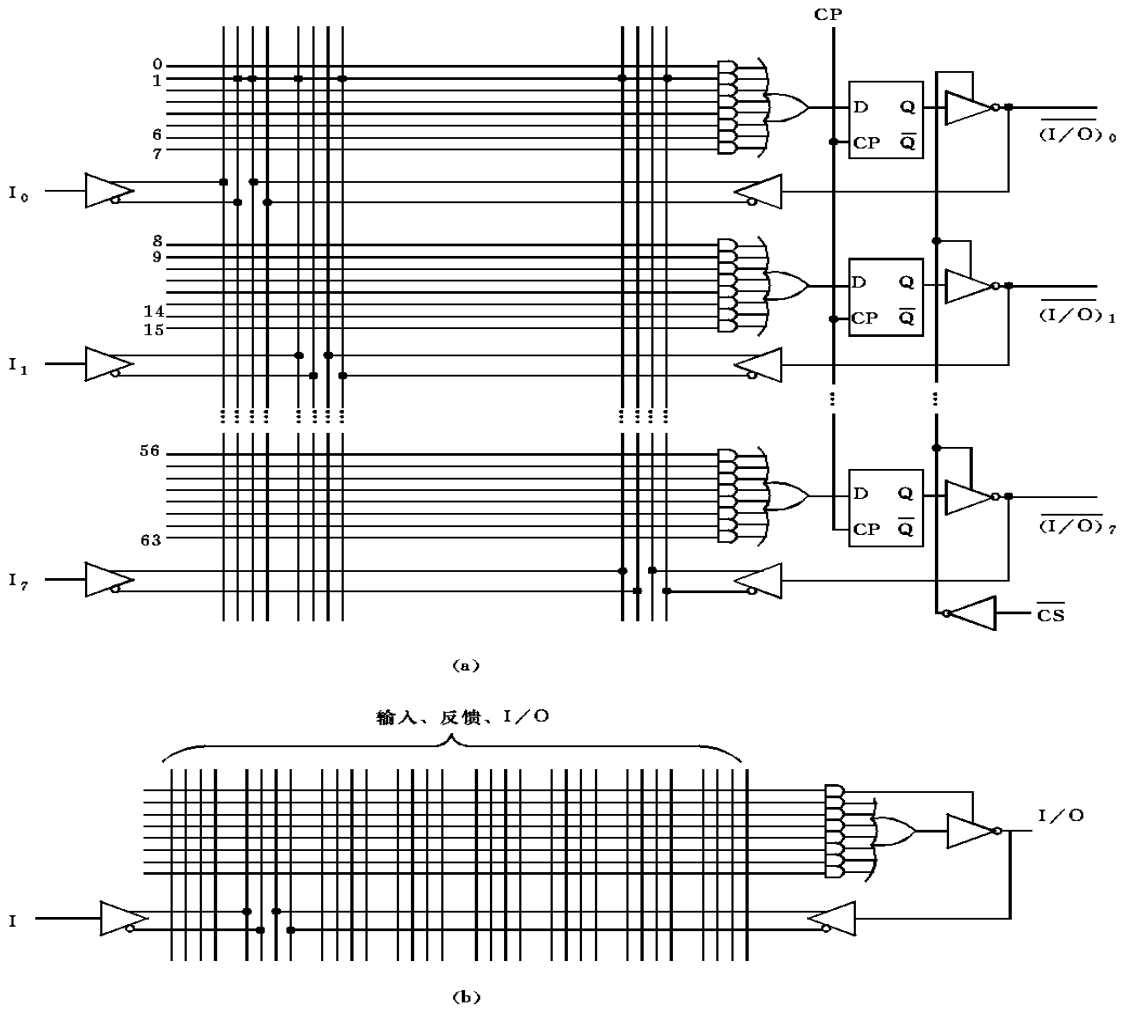


图2.31 两种带反馈的阵列型PAL

2.6.4 通用阵列逻辑(GAL)

通用阵列逻辑(generic array logic，简称GAL)器件是一种可用电擦除的，可重复编程的高速PLD。它与PAL器件的主要区别在于：

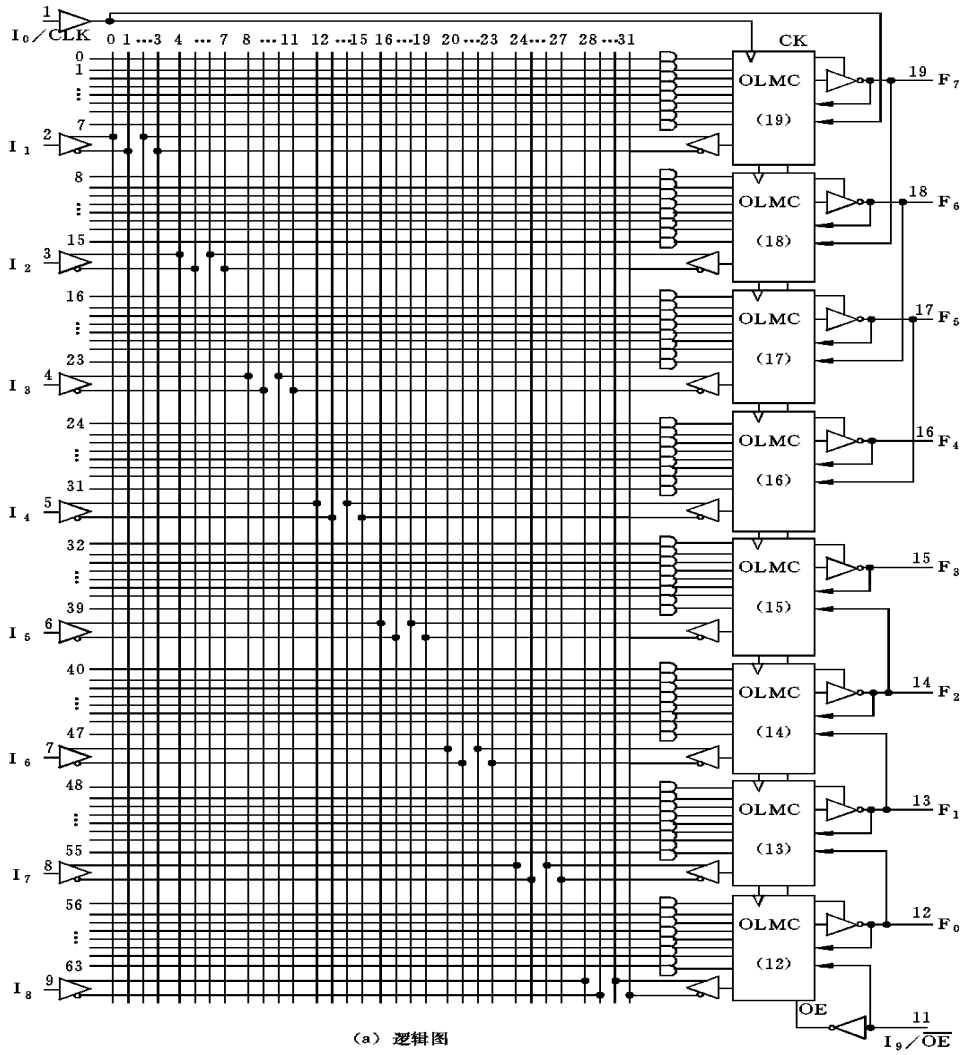
- PAL采用的是熔丝工艺，一旦编程后就不能改写，而GAL采用可用电擦除的CMOS(E^2CMOS)工艺，可擦除重写100次以上，数据可保存20年以上，在数秒钟内即可完成擦除和编程过程。

- PAL器件的应用局限性较大，对于不同的输出结构，需选用不同型号的PAL器件。而GAL的输出结构有一个输出逻辑宏单元(OLMC)，通过对它的编程，使GAL有多种输出方式：**寄存器型输出、组合逻辑输出，并可控制三态输出门**，因此显得很灵活。

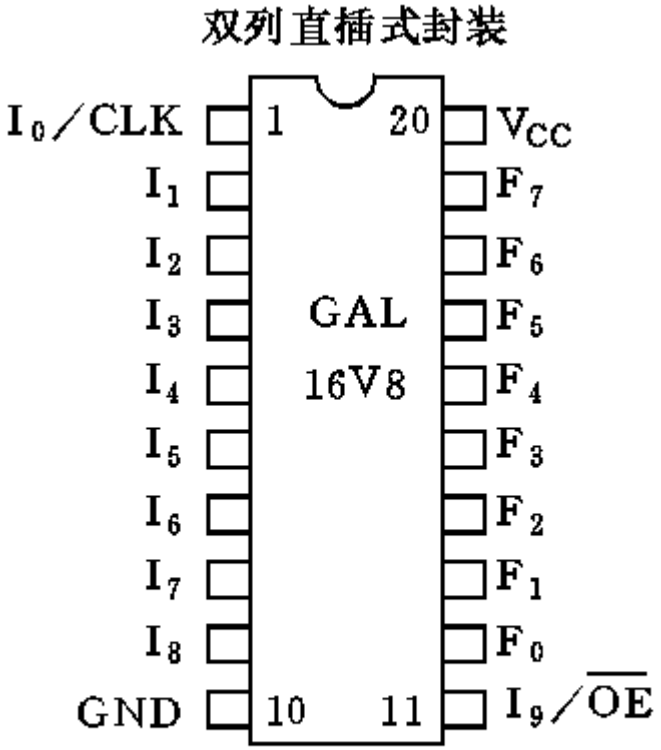
1. GAL的基本结构

以GAL 16V8型器件为例，GAL16V8包括：输入门、输出三态门、与门阵列、输出逻辑宏单元(内含或阵列)以及从输出反馈到输入的控制门等。

GAL16V8 8个固定为输入端（I_0-I_7），8个I/O端（F_0-F_7）（可以通过编程确定其为输入或输出),引出端1与11也有两种选择，可以通过编程确定。最多16个输入端（输出端为2个），最多8个输出端（输入端《=10）。


1
I0/CLK
0 1 ···3 4 ··· 7 8 ···11 12 ···15 16···19 20···23 24···27 28···31
CK
OLMC
(19)
OLMC
(18)
OLMC
(17)
OLMC
(16)
OLMC
(15)
OLMC
(14)
OLMC
(13)
OLMC
(12)
OE
I1
I2
I3
I4
I5
I6
I7
I8
I9/OE
F7
F6
F5
F4
F3
F2
F1
F0

（a）逻辑图



（b）封装图

图2.32 GAL 16V8逻辑图与封装图

2.6.5 门阵列(GA)、宏单元阵列(MA)、标准单元阵列(SCA)

1. 基本组成形式

这三种阵列电路内部的单元是以阵列形式排列--**阵列逻辑电路**。

主要实现批量较大的专用集成电路(application specific integrated circuit，简称ASIC)。由用户向集成电路生产厂家定做。

(1) 门阵列(gate array,简称GA)

- 门阵列设计利用预先制造好的“母片”来进行布图设计。**母片上通常以一定的间距成行成列的排列着基本单元电路**。
- 基本单元**一般由6—10晶体管组成**
- 门阵列设计的**优点是设计自动化程度较高**，设计周期短，设计成本低。
- **门阵列的缺点是布图密度低**，并且品种有限，为了使所有单元间的连线能布通，势必造成芯片面积利用率的下降。

(2) 宏单元阵列(macro cell array,简称MA)

对门阵列进行改进，产生宏单元阵列

- 宏单元阵列按列排列，每一列由若干个基本单元构成，在每两个基本单元之间有一个走线过道，基本单元之间的连线在垂直和水平走线通道中进行。

- **一个逻辑元件可由一个基本单元或若干个基本单元构成，称为宏单元**。

- 宏单元阵列自动设计系统有一个“宏单元库”，存有**门电路、触发器、加法器、译码器等各类逻辑元件**。由于宏单元的逻辑功能比较强，因而布图密度比门阵列高。宏单元阵列也是一种半用户器件，具有制造周期短等优点。

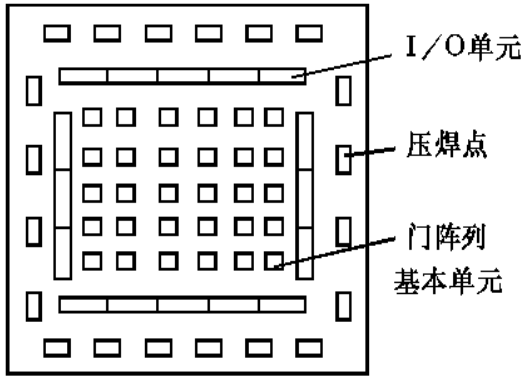


图2.34 门阵列母片芯片布图(示意图)

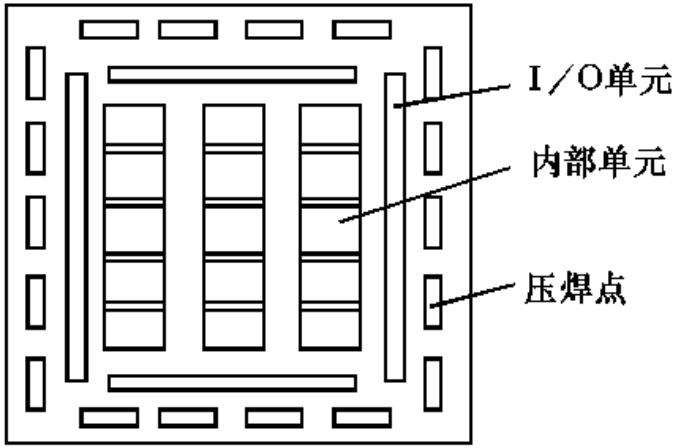


图2.35 宏单元阵列

(3) 标准单元阵列 (standard cell array,简称SCA)

- **标准单元阵列又称为多元胞阵列** (polycell array), 它以预先设计好的功能单元(称为标准单元或多元胞)为基础，**这些单元可以是门、触发器或有一定功能的功能块(如加法器)**。

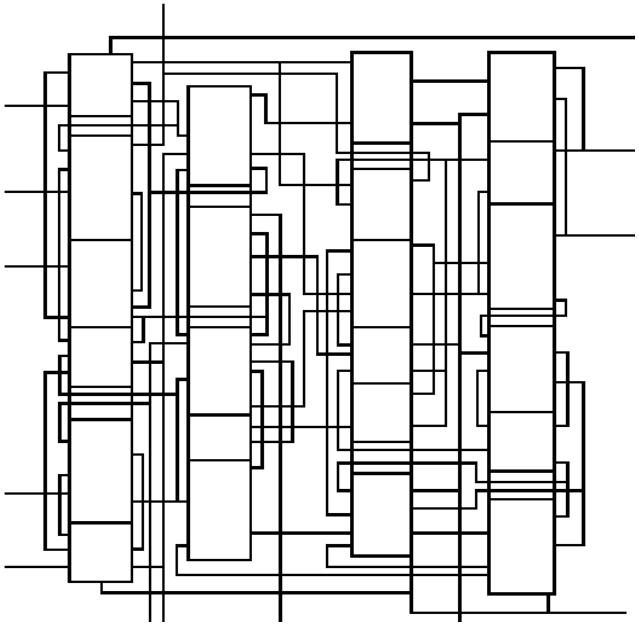


图2.36 标准单元阵列的排列

2.6.6 现场可编程序门阵列(FPGA)

主要由四个部分组成:

- **可编程序逻辑宏单元(CLB):**它以阵列形式分布在芯片的中心部位。每个CLB由**若干个触发器及一些可编程序组合逻辑部件**组成。CLB可通过编程来实现用户所需的逻辑。
- **可编程序输入输出宏单元(IOB):**它排列在CLB四周，是芯片内部CLB与芯片外部引脚间的可编程接口，每个IOB可进行边沿触发器、锁存器、上拉电阻选择、三态选择等输入输出方式控制。IOB也是通过编程来实现所需的输入输出方式控制的。
- **互连资源:**它包括可编程的互连开关矩阵、内部长线、总线等。
- **重构逻辑的程序存储器:**它以阵列形式分布在整个芯片上。

FPGA器件工作时:将用户所需实现的逻辑以某种程序形式从片外读至PGA重构逻辑的程序存储器内，该存储器的**存储单元输出直接去控制**指定的CLB，IOB等单元，从而使器件有确定的功能。**常把这一过程称为配置**。

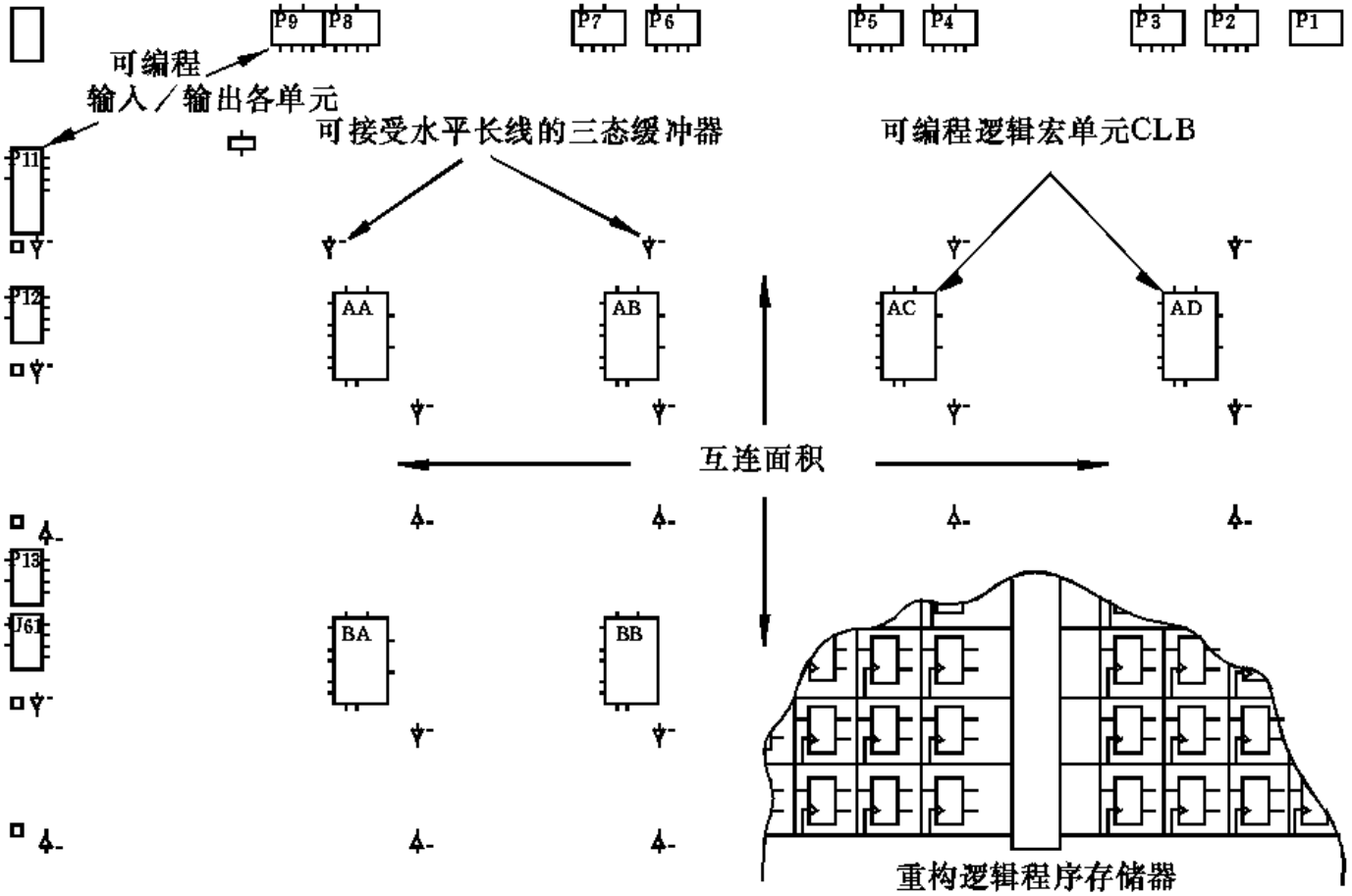


图2.47 可编程序门阵列结构图