

第七章 可编程逻辑器件 PLD

第一节 基本内容

一、基本知识点

(一) 可编程逻辑器件 PLD 基本结构

可编程逻辑器件 PLD 是 70 年代发展起来的新型逻辑器件,相继出现了只读存储器 ROM、可编程只读存储器 PROM、可编程逻辑阵列 PLA、可编程阵列逻辑 PAL、通用阵列逻辑 GAL 和可擦写编程逻辑器件 EPLD 等多个品种,它们的组成和工作原理基本相似。PLD 的基本结构由与阵列和或阵列构成。与阵列用来产生有关与项,或阵列把所有与项构成“与或”形式的逻辑函数。在数字电路中,任何组合逻辑函数均可表示为与或表达式,因而用“与门—或门”两级电路可实现任何组合电路,又因为任何时序电路是由组合电路加上存储元件(触发器)构成的,因而 PLD 的“与或”结构对实现数字电路具有普遍意义。

在 PLD 中,输入电路中为了适应各种输入情况,每一个输入信号都配有一缓冲电路,使其具有足够的驱动能力,同时产生原变量和反变量输出,为与门阵列提供互补信号输入。输出电路的输出方式有多种,可以由或阵列直接输出,构成组合方式输出,也可以通过寄存器输出,构成时序方式输出。输出既可以是低电平有效,也可以是高电平有效;既可以直接接外部电路,也可以反馈到输入与阵列,由此可见 PLD 的输出电路根据不同的可编程逻辑器件有所不同。

(二) 可编程逻辑器件分类

1. 按编程部位分类

PLD 有着大致相同的基本结构,根据与阵列和或阵列是否可编程,分为三种基本类型:

- (1) 与阵列固定,或阵列可编程
- (2) 与或阵列均可编程
- (3) 与阵列可编程,或阵列固定

归纳上述 PLD 的结构特点,列于表 7-1。

表 7-1 各种 PLD 的结构特点

类 型	阵 列		输出方式
	与	或	
PROM	固定	可编程	TS, OC
PLA	可编程	可编程	TS, OC, H, L
PAL	可编程	固定	TS, I/O寄存器
GAL	可编程	固定	用户定义

2. 按编程方式分类

- (1) 掩膜编程
- (2) 熔丝与反熔丝编程
- (3) 紫外线擦除、电可编程
- (4) 电擦除、电可编程
- (5) 在系统编程 (Isp)

(三) 高密度可编程逻辑器件 HDPLD

通常衡量可编程逻辑器件芯片的密度是以芯片能容纳等效逻辑门的数量，一般是以 2000 为界限，即芯片容纳等效逻辑门小于 2000 门，称它为低密度可编程逻辑器件或简单的可编程逻辑器件 (SPLD)，若大于 2000 等效逻辑门，称为高密度可编程逻辑器件 (HDPLD)。在前面按编程部位分类可编程逻辑器件中提及的通用阵列逻辑 (GAL) 的等效逻辑门一般不超过 2000 门，习惯上称其为低密度可编程逻辑器件。

通用阵列逻辑 GAL 是在 PAL 基础上发展起来的一种具有较高可靠性和灵活性的新型可编程逻辑器件，它采用 E²CMOS 工艺和灵活的输出结构，能将数片中小规模集成电路集成在芯片内部，并具有电擦写反复编程的特性。在基本阵列结构上仍是与阵列可编程，或阵列固定的结构。GAL 在输出结构配置了 8 个可以任意组态的输出逻辑宏单元 OLMC (Output Logic Macro Cell)，适当地为输出逻辑宏单元进行编程组态，GAL 就可以在功能上代替编程阵列逻辑 PAL。

输出逻辑宏单元 OLMC 由或门、异或门、D 触发器、多路选择器 MUX、时钟控制、使能控制和编程元件等组成。

高密度可编程逻辑器件 HDPLD (High Density Programmable Logic Device) 从芯片密度上有了很大的改进，单片芯片内可以集成成千上万个等效逻辑门，因此在单片高密度可编程逻辑器件内集成数字电路系统成为可能。HDPLD 器件在结构上仍延续 GAL 的结构原理，因而还是电擦写、电编程的 EPLD 器件。

(四) 现场可编程逻辑器件 FPGA

可编程逻辑器件基本组成是与阵列、或阵列和输出电路。对这些基本组成电路进行编程就可以实现任何积之和的逻辑函数，再加上触发器则可实现时序电路。现场可编程门阵列的编程单元是基于静态存储器 (SRAM) 结构，不像 PLD 那样受结构的限制，它可以靠门与门的连接来实现任何复杂的逻辑电路，更适合实现多级逻辑功能。

现场可编程门阵列 FPGA 与 HDPLD 相比较特点如下：

(1) FPGA 的编程单元是基于 SRAM 结构，可以无限次编程，它为易失性元件，掉电后芯片内信息丢失。通电之后，要为 FPGA 重新配置逻辑。

(2) FPGA 中实现逻辑功能的 CLB 比 HDPLD 实现逻辑功能的 OLMC 规模小，制作一个 OLMC 的面积可以制作多个 CLB，因而 FPGA 内的触发器要多于 HDPLD 的触发器，使得 FPGA 在实现时序电路时要强于 HDPLD。

(3) HDPLD 的信号汇总于编程内连矩阵，然后分配到各个宏单元，因此信号通路固定，系统速度可以预测。而 FPGA 的内连线是分布在 CLB 周围，而且编程的种类和编程点很多，使得布线相当灵活。

(4) 由于 FPGA 的 CLB 规模小，可分为两个独立的电路，又有丰富的连线，所以系统综合时可进行充分的优化，以达到逻辑最高的利用。

(五) 随机存取存储器 RAM

随机存取是指可以随时将数据存入 (称写入)，和取出 (称读出)。随机存储器 (RAM) 的主要指标是存储器容量和存取时间 (周期)。存储容量表示一片 RAM 存储数据的能力。存放一个二进制数码需要一个存储单元，所以存储容量常用存储单元的总数 (bit) 来表示。存取时间表示从存储器中开始存取第一个字到能够存取第二个字为止所需的时间，或称为存取周期。存取时间越短，表示存储器的存取速度越高。

RAM 的基本结构可以分为三个部分：存储矩阵，地址译码器及读写控制电路。存储矩阵

是用来存储要存放的代码，矩阵中每个存储单元都用一个二进制码给以编号，以便查询此单元。译码器可以将输入地址译为电平信号，以选中存储矩阵中的相应的单元。

存储器根据工作原理的不同可分为静态 RAM 和动态 RAM 两大类。

(1) 静态 RAM

静态 RAM 是在触发器的基础上附加控制线或门控制管构成的，它们是靠电路状态来存储数据。根据使用的器件不同，静态存储单元又分为 MOS 型和双极型两种。

(2) 动态 RAM

动态 RAM 是利用 MOS 管栅级电容能够存储电荷的原理制成的。电路结构比较简单，但由于栅极电容的容量很小，而漏电流不可能为零，所以电荷的存储时间有限。为了及时补充泄露掉的电荷以避免存储信号丢失，必须定时给栅极电容补充电荷。通常把这种操作叫做刷新或再生。因此，工作时必须辅以比较复杂的刷新电路。

二、重点

1.可编程逻辑器件 PLD 的基本结构与工作原理基于任何组合逻辑函数均可化为与或式,从而实现与门—或门两级电路实现，而任何时序电路又都是由组合电路加上存储元件（触发器）构成。

2.可编程逻辑器件 PLD 按编程部位分类、编程方法分类的基本概念及其特征。

3.多次可擦写的可编程逻辑器件 PLD 主要基于浮栅技术，这种编程方法是一只多晶硅浮栅浮于控制栅和衬底之间的半导体中。当控制栅上的电压加大时，产生很强的电场，足以使电子获得能量穿过半导体进入浮栅住留。这样 MOS 管因为浮栅上存储负电荷作用使开启电压改变，从而达到逻辑编程“0”和“1”的目的。

4.单片可编程逻辑器件容量总是有限的，所以在设计时，应考虑利用多片 PLD，按一定方法连接以扩展其容量。

(1) 字长扩展

字长又称为数据位数，对字长的扩展即是地址的位数保持不变，而对数据位增加。

(2) 字扩展

字又称为地址位数，对字的扩展即是数据的位数保持不变，而对地址位增加。

在实际应用中，往往需要同时进行地址扩展和数据扩展，例如存储器总容量为 16K X 16bit，用 2K X 8bit 芯片构成存储器时，必须同时进行地址扩展和数据扩展，用 16 片 2K X 8bit 的芯片，依据一定的连接方式连接，便可得到总容量为 16K X 16bit=256Kbit 的存储器。

5.可编程逻辑阵列 PLA 电路的分析方法：

(1) 根据题意或者电路图，写出逻辑与-或表达式；

(2) 若时序电路，则写出激励、驱动和输出方程；

(3) 写出真值表或者状态图；

(4) 根据真值表或者状态图分析其工作原理。

6.可编程阵列逻辑 PAL 和通用阵列逻辑 GAL 的基本组成和 PAL 的各种组态、GAL 的输出宏单元各种组态及其特点。

7.高密度可编程逻辑器件 HDPLD 的基本组成和其工作原理。

8.现场可编程门阵列 FPGA 工作原理是靠门与门的连接实现任何复杂的逻辑电路，较适于多级逻辑设计。重点在于掌握工作原理及其特点。

9. 可编程逻辑器件设计实质上是设计专用集成电路 ASIC，整个设计过程必须伴随着电子

设计自动化 (EDA) 软件平台。本教材第二篇指导读者利用 Foundation 软件设计平台设计高密度 PLD 器件, 完成可编程逻辑器件的设计。

三、难点

1. 可编程逻辑器件的不同分类方法及其各种基本概念。

2. PLA 电路的设计方法:

- (1) 根据题意写出真值表或者状态图;
- (2) 选择触发器;
- (3) 写出驱动、激励和输出方程;
- (4) 画出 PLA 电路图。

3. 现场可编程门阵列 FPGA 中的可配置逻辑块 CLB 不仅可以完成组合逻辑、时序逻辑电路的功能, 而且还可以作为 RAM 使用。当作为 RAM 使用时, 不仅可以配置成电平触发的 16 位双口或 32 位单口 RAM, 而且还可以配置成边沿触发的 16 位双口或 32 位单口 RAM。

第二节 典型题解

例 7-1 试用 SD805 32×8bit PROM 构成容量为 32×32bit 的 PROM。

解: 此题为扩展存储器的数据位 (字长), 方法比较简单。在驱动器的负载能力允许之内, 将每个存储器地址输入端对应连接, 且允许输入端 S 接在一起既可。采用四片 SD805 PROM 构成 32×32bit 的 PROM, 其电路连接图如图 7-1 所示, 因为每片 SD805 输出 8 位, 其中 SD805- (1) 输出 0~7 位 (32 位中的低 8 位), SD805- (4) 输出 24~31 位 (32 位中的高 8 位)。

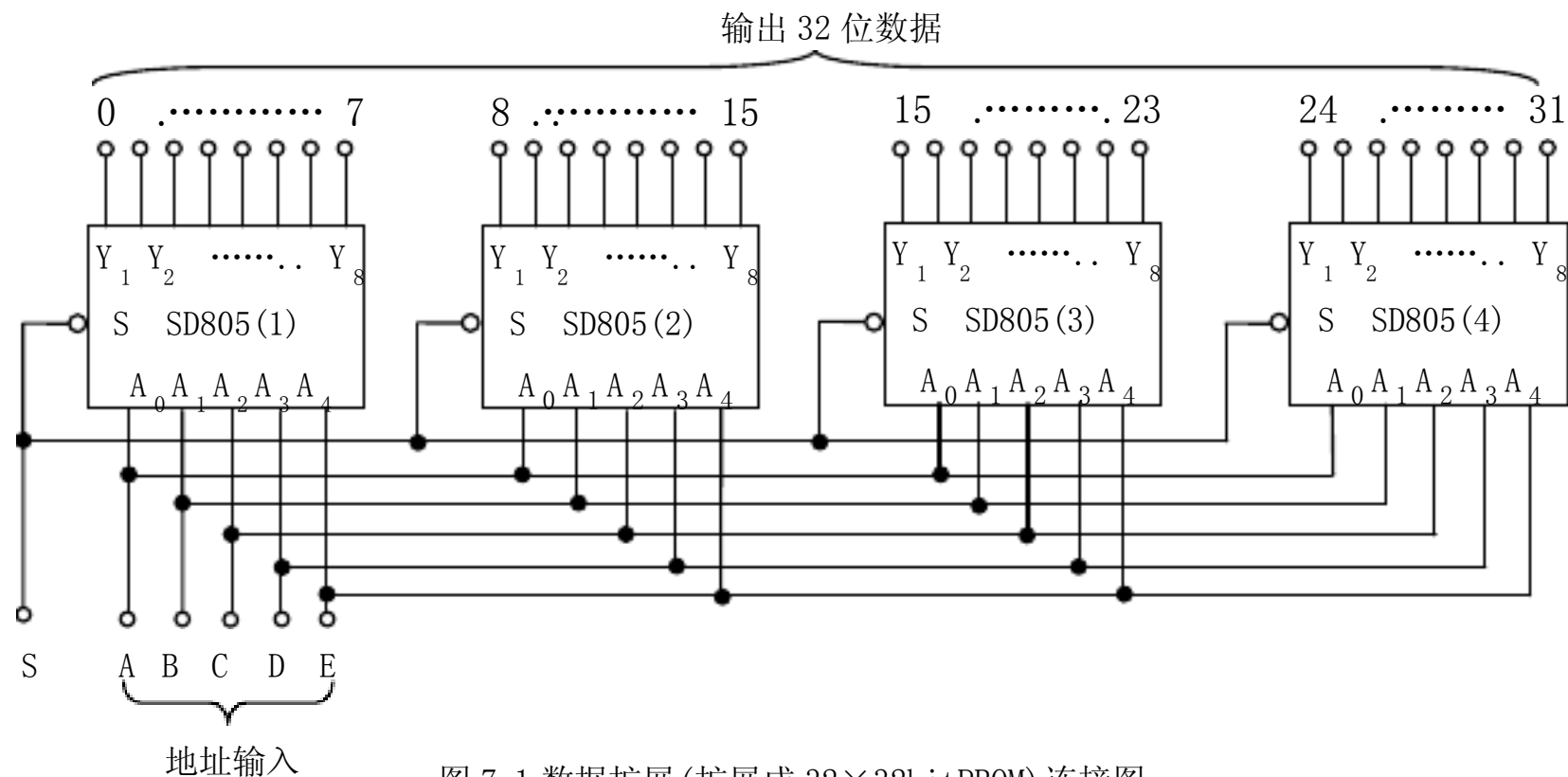


图 7-1 数据扩展 (扩展成 32×32bit PROM) 连接图

例 7-2 试用 SD805 32×8bit PROM 构成容量为 512×8bit PROM。

解: 此题为扩大存储器的地址 (字数), 一个 SD805 容量是 32 字×8 位, 可利用允许输入来扩展字数, 即每片一个字组, 通过外加译码器 T1154 来分别选中每一片, 也就将该字组的 32 个字选中, 这样 SD805 本来只有五位地址输入, 可选中 32 个字, 现采用 16 片 SD805

与一片译码器 T1154，经扩展成九位地址输入后，可选中 512 个字，其电路连接图如图 7-2 所示。九位地址码 ABCDEFGHI 中，I 为最高位，A 为最低位。当 T1154 译码器输入 FGHI=0000 时， Y_0 输出有效，选中 SD805- (1)，决定 0~31 字，当 FGHI=1000 时，选中 SD805- (2)，决定 32~63 字.....，其余类推。当 FDHI=1111 时，选中 SD805- (16)，即决定 479~511 字。

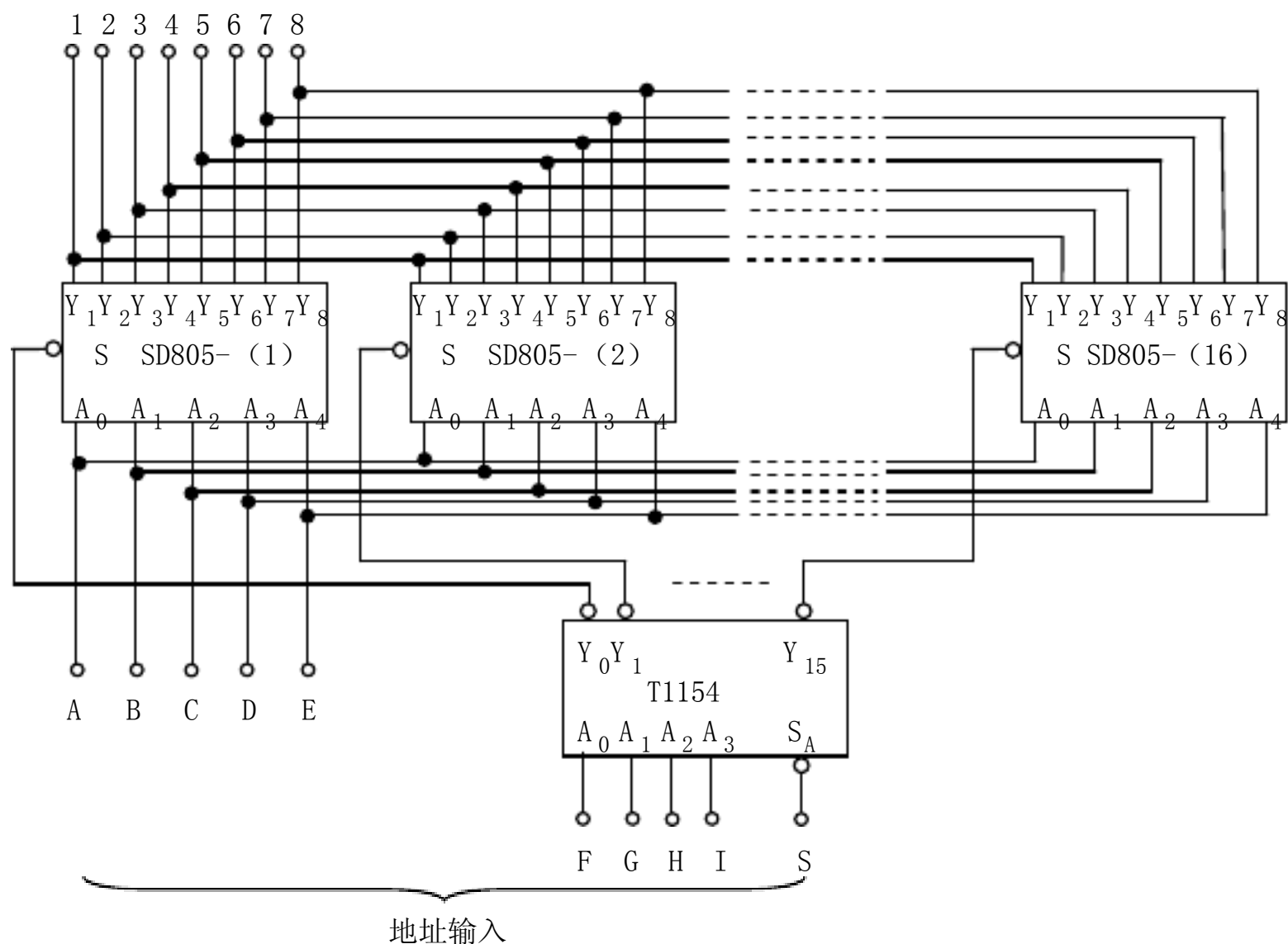


图 7-2 扩大存储器地址连接图

例 7-3 试用 SD805 32×8bit PROM 构成容量为 128×2 bit PROM。

解：此题是增加地址（字数）、减少位数。可用一片 SD805 外加双 4 选 1 数据选择器 T1153 来实现。因为 SD805 容量为 32 字×8 位，即有 32×8=256 个存储单元，正好满足 128 字×2 位 ROM 的容量，电路连接图如图 7-3 所示。通过七位地址输入端 ABCDEFG 对 128 寻址，其中 A 是最低位，G 是最高位，字选地址表如表 7-2 所示。

例 7-4 有两个两位二进制数，它们都是正整数，试用 ROM 实现对这两个数的乘法运算。

解：此例是利用 ROM 实现组合逻辑电路的设计问题。依照所要求的电路功能，可按两个 2 位二进制数的乘法运算列出真值表。在用 ROM 实现电路时，可用输入信号取值组成的代码作为地址代码构成地址译码器，其输出即是存储器的字线，译码器的结构用与阵列表示。对应于输入取值经乘法运算后所得的二进制数，作为一个字的信息存储在相应字线指示的存储单元中。每 1 位二进制数对应一条位线，各条位线的函数关系用或阵列表示，位线通过输出电路输出。依照题意，分别设这两个二进制数是 A_1A_0 和 B_1B_0 ，设输出函数 F，因为输出函数 F 是十进制数 9，所以应该用四位二进制数表示输出函数 F。列出电路真值表如表 7-3 所示，利用 ROM 实现的乘法器的与或阵列图如图 7-4 所示。

表 7-2 字选地址表

地址		AB		0 0	1 0	0 1	1 1
		C D E F G	$Y_1 Y_5$	$Y_2 Y_6$	$Y_3 Y_7$	$Y_4 Y_8$	
0 0 0 0 0		W_1	W_2	W_3	W_4		
1 0 0 0 0		W_5	W_6	W_7	W_8		
⋮		⋮	⋮	⋮	⋮		
1 1 1 1 1		W_{125}	W_{126}	W_{127}	W_{128}		

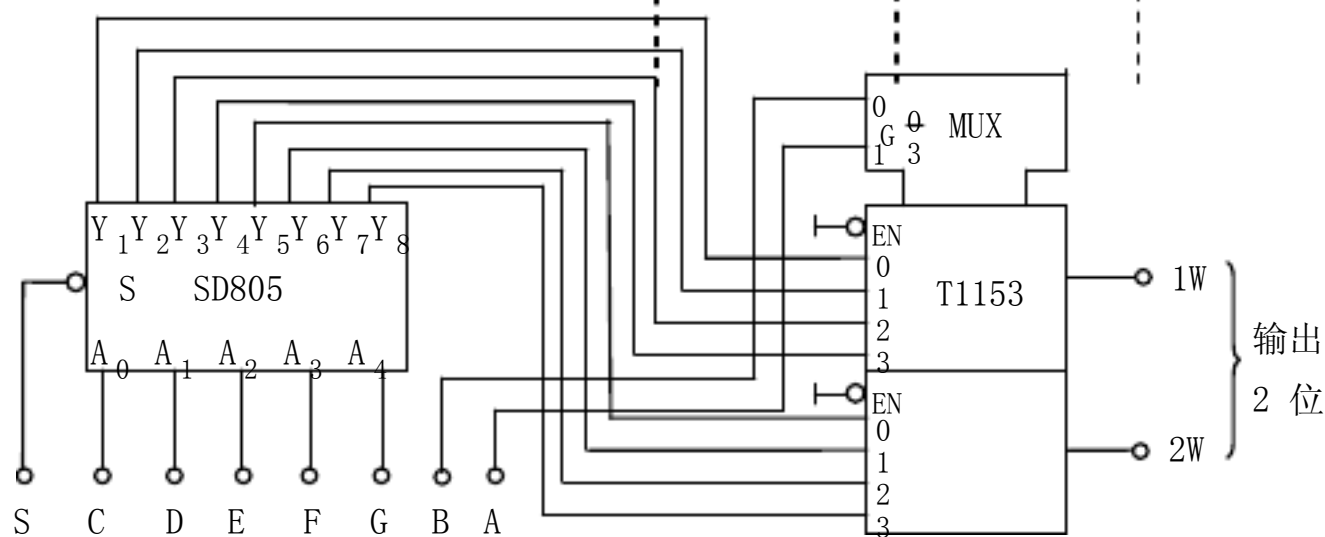


图 7-3 用 SD805 实现 128×2 位 ROM

表 7-3 例 7-4 电路真值表

A_1	A_0	B_1	B_0	F_3	F_2	F_1	F_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

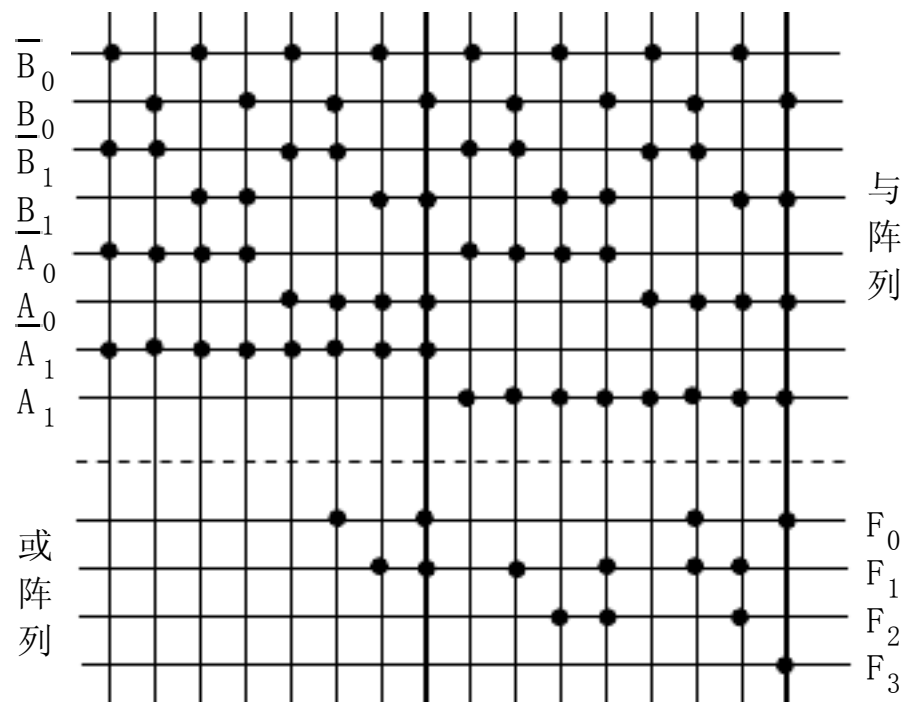


图 7-4 例 7-4 阵列图

例 7-5 芯片 CT4161 功能和 PROM 组成图 7-5 所示电路。要求：

- (1) 分析 CT4161 功能，说明电路的计数长度。
- (2) 分析 W、X、Y、Z 的函数表达式。
- (3) 在 CP 作用下，分析 W、X、Y、Z 端顺序输出的 8421BCD 码的状态，并说明电路的功能。

解：(1) CT4161 是同步 16 进制计数器， Q_D 、 Q_C 、 Q_B 、 Q_A 状态由 0000，0001 到 1111，

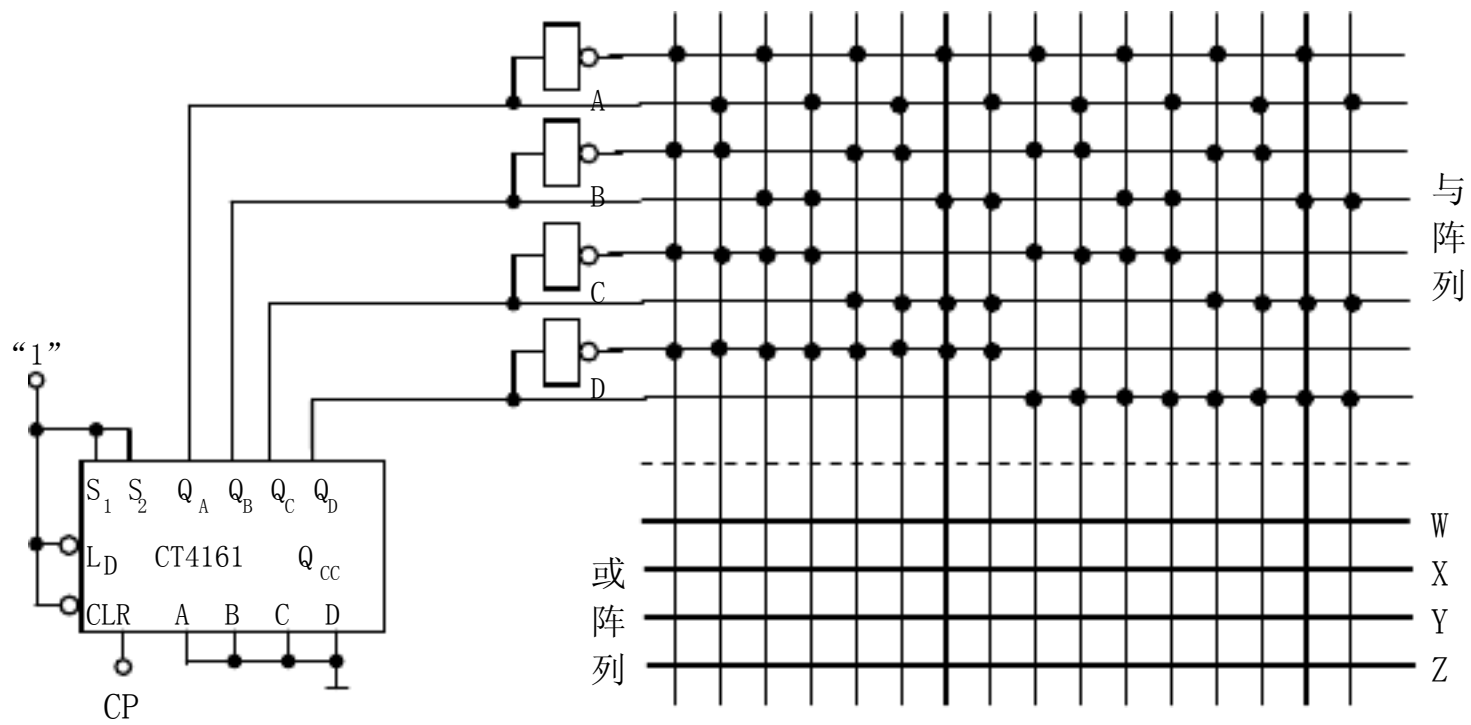


图 7-5 例 7-5 阵列逻辑图

再重复。

(2) W、X、Y、Z 的函数表达式为：

$$W = \sum m(5, 11, 12, 14)$$

$$X = \sum m(2, 4, 7, 8, 10, 13)$$

$$Y = \sum m(0, 6, 7, 9, 13, 15)$$

$$Z = \sum m(0, 1, 3, 4, 5, 8, 9, 10, 12, 13, 14, 15)$$

(3) WXYZ 端顺序输出为 3141592653589793 的 8421BCD 码。因此该电路是一个能产生 16 位的 π 函数发生器。

例 7-6 用 EPROM 设计一个字符发生器。

解：字符发生器是显示器中常用的逻辑部件。它将各种字母、数字及符号预先存储在 ROM 中，只要给出适当地址码，就能将这些字符读出来，并驱动显示器显示这些字符。图 7-6 给出了用 7×5 字符发生器存储字符“E”的原理。图中存储体有七行五列，构成 7×5 点阵。

根据字符的形状可在存储单元中存入 1 或 0，然后顺序地给出地址码，就可以读出各行的内容，每读一行，原来存储“1”的地方出现光点，全部光点就组成一个字符。

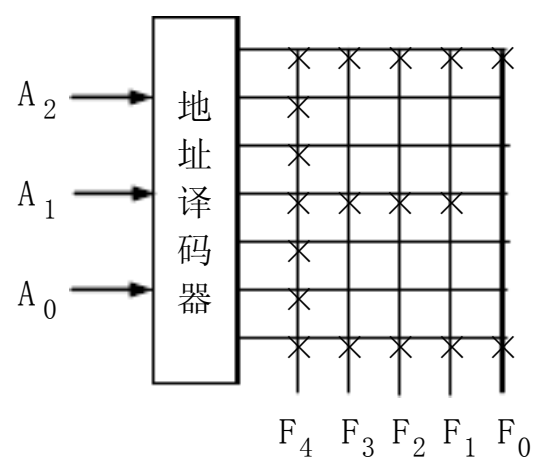


图 7-6 7×5“E”字符发生

例 7-7 试设计产生图 7-7 所示四路周期信号的逻辑电路（采用 ROM 设计电路）。

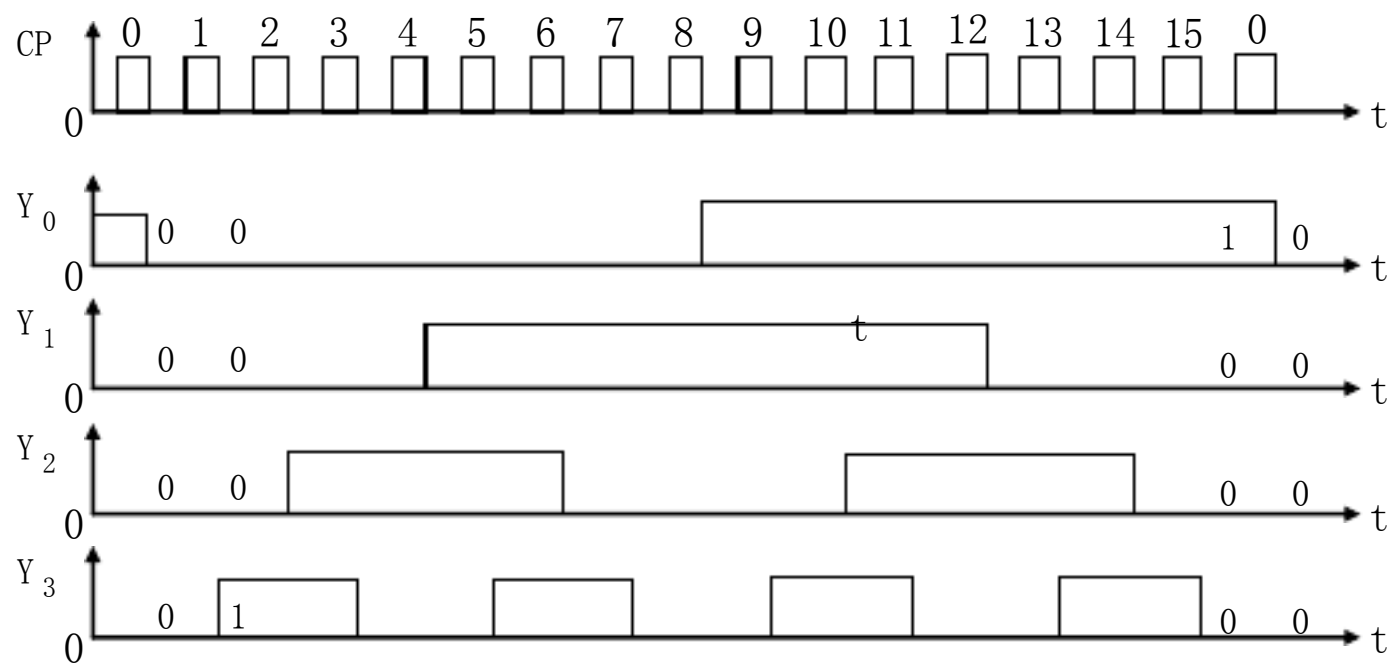


图 7-7 四路周期信号

解：由图 7-7 看出，要求产生的四路信号是周期为 16 的四组同步序列，如表 7-4 所示。

用一个模 16 同步加法计数器产生四位地址，计数器状态由状态 0~15 循环转换，每个状态便给出一组四位地址。随着计数器状态的循环转换，地址循环选通，从 ROM 输出端就得到四组同步序列。为了使四组同步序列符合真值表 7-4，必须依据序列要求给 ROM 正确编程，为此，由表 7-4 得

$$\begin{aligned}
 Y_3 &= W_1 + W_2 + W_5 + W_6 + W_9 + W_{10} + W_{13} + W_{14} \\
 Y_2 &= W_2 + W_3 + W_4 + W_5 + W_{10} + W_{11} + W_{12} + W_{13} \\
 Y_1 &= W_4 + W_5 + W_6 + W_7 + W_8 + W_9 + W_{10} + W_{11} \\
 Y_0 &= W_8 + W_9 + W_{10} + W_{11} + W_{12} + W_{13} + W_{14} + W_{15}
 \end{aligned}$$

表 7-4 例 7-7 真值表

计数器状态				字线序列输出				
Q_3	Q_2	Q_1	Q_0	W	Y_3	Y_2	Y_1	Y_0
0	0	0	0	W_0	0	0	0	0
0	0	0	1	W_1	1	0	0	0
0	0	1	0	W_2	1	1	0	0
0	0	1	1	W_3	0	1	0	0
0	1	0	0	W_4	0	1	1	0
0	1	0	1	W_5	1	1	1	0
0	1	1	0	W_6	1	0	1	0
0	1	1	1	W_7	0	0	1	0
1	0	0	0	W_8	0	0	1	1
1	0	0	1	W_9	1	0	1	1
1	0	1	0	W_{10}	1	1	1	1
1	0	1	1	W_{11}	0	1	1	1
1	1	0	0	W_{12}	0	1	0	1
1	1	0	1	W_{13}	1	1	0	1
1	1	1	0	W_{14}	1	0	0	1
1	1	1	1	W_{15}	0	0	0	1

上式很容易用二极管或多发射极晶体管构成的存储矩阵予以实现，所以用一片中规模四位二进制计数器和一个 16 字×4 位 ROM 就可以实现题意功能，逻辑框图如图 7-8 所示。

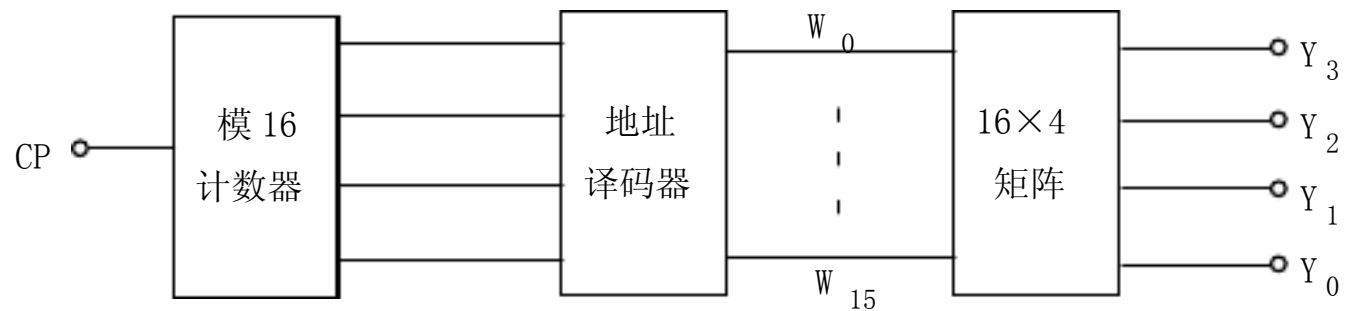


图 7-8 例 7-7 逻辑框图

例 7-8 用 PLA 设计一个四位自然二进制码——格雷码的转换电路。

解：四位自然二进制码与格雷码之间的转换关系如表 7-5 所示，这是一个多输出函数的问题。对表 7-5 中的真值表所示的函数进行简化，得到：

$$\begin{aligned}
 G_3 &= B_3 \\
 G_2 &= B_2 \bar{B}_3 + \bar{B}_2 B_3 \\
 G_1 &= B_1 \bar{B}_2 + \bar{B}_1 B_2 \\
 G_0 &= B_0 \bar{B}_1 + \bar{B}_0 B_1
 \end{aligned}$$

由上面逻辑函数表达式看出，它们包含七个“与”项，即 B_3 , $B_2 \bar{B}_3$, $\bar{B}_2 B_3$, $B_1 \bar{B}_2$, $\bar{B}_1 B_2$, $B_0 \bar{B}_1$, $\bar{B}_0 B_1$ 用 PLA 实现上述函数时，其阵列图如图 7-9 所示。

表 7-5 8421 码和格雷码对照真值表

B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

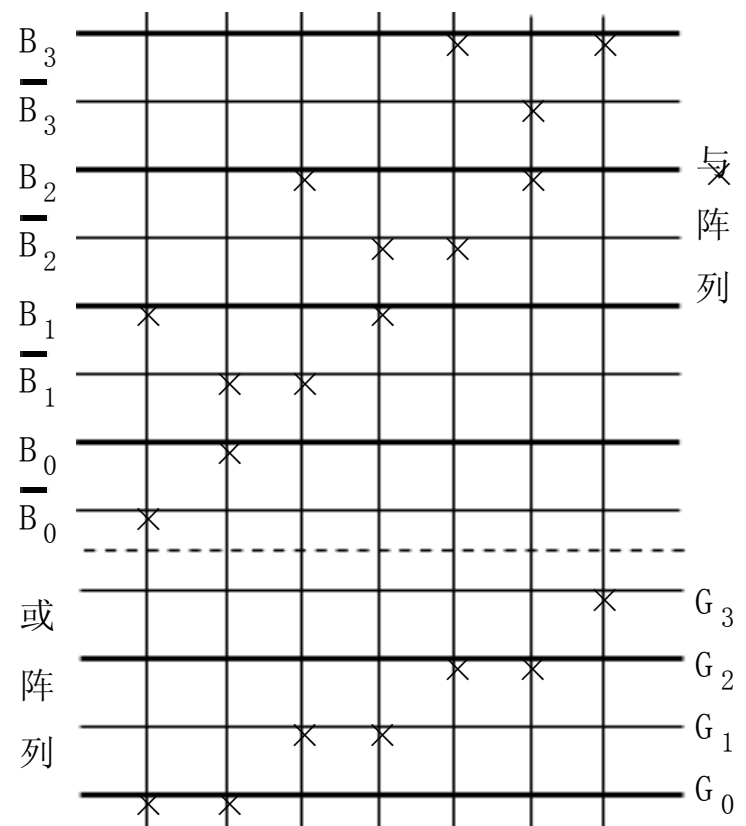


图 7-9 例 7-8 PLA 阵列图

例 7-9 PLA 和 D 触发器组成的同步时序电路如图 7-10 所示。要求：

- 写出电路的驱动方程、输出方程。
- 分析电路功能，画出电路的状态转换图。

解：(1) 根据 PLA 与一或阵列的输入/输出关系，可直接得到各触发器的激励方程及输出方程：

$$\begin{aligned}
 D_0 &= \bar{Q}_0 + \bar{Q}_1 Q_0 & D_1 &= \bar{Q}_1 Q_0 + Q_1 Q_0 \\
 D_2 &= \bar{Q}_0 \bar{Q}_2 + Q_2 Q_0 \\
 Q_{CC} &= \bar{Q}_0 Q_1 Q_2 + Q_0 \bar{Q}_1 \bar{Q}_2
 \end{aligned}$$

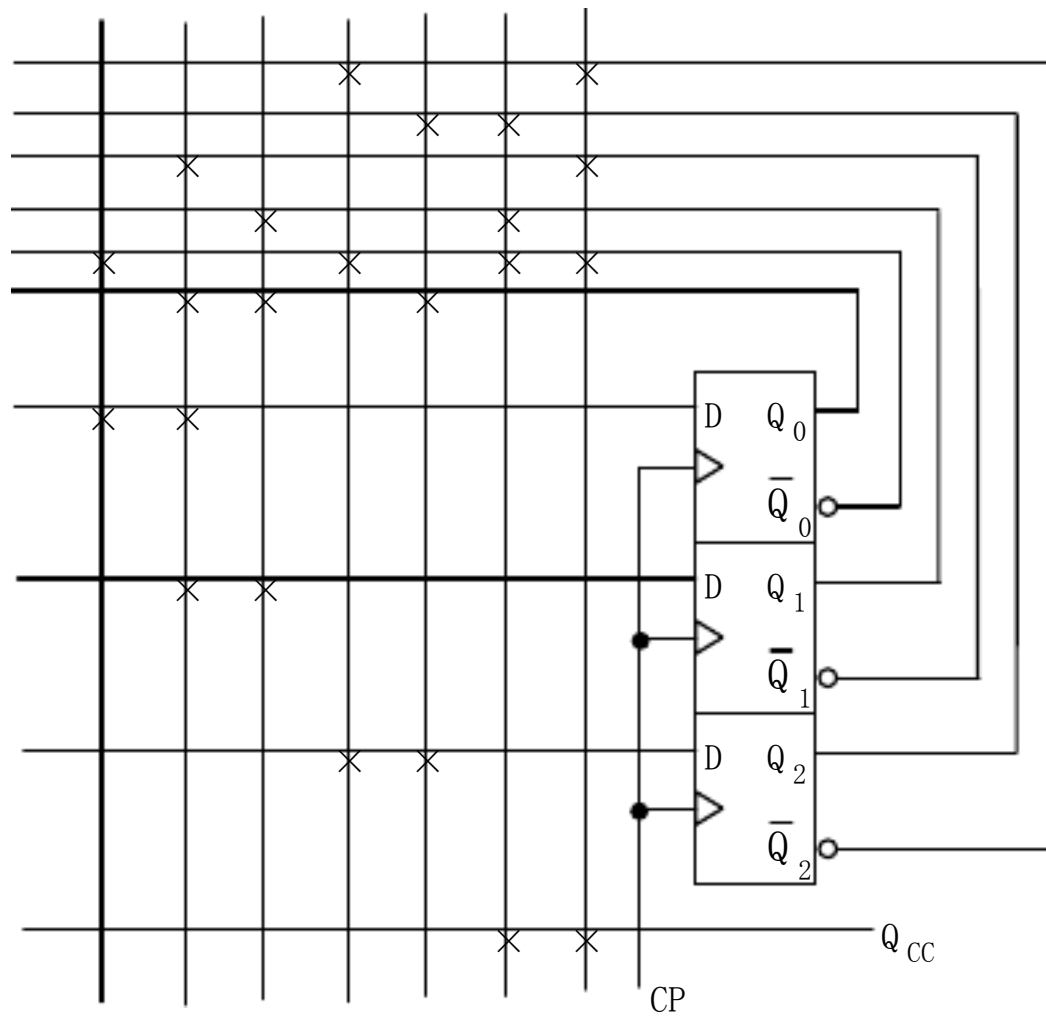


图 7-10 例 7-9 PLA 同步时序电路

(2) 先设定电路的状态，根据触发器的激励方程和输出方程，可列出表 7-6 所示的电路状态转换表，并画出图 7-11 所示的电路状态转换图。

该电路是能够自启动的同步六进制计数器。

表 7-6 例 7-9 电路状态转换表

Q_2	Q_1	Q_0	D_2	D_1	D_0	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}	Q_{cc}
0	0	0	1	0	1	1	0	1	1
0	0	1	0	1	1	0	1	1	0
0	1	0	1	0	1	1	0	1	0
0	1	1	0	1	0	0	1	0	0
1	0	0	0	0	1	0	0	1	0
1	0	1	1	1	1	1	1	1	0
1	1	0	0	0	1	0	0	1	1
1	1	1	1	1	0	1	1	0	0

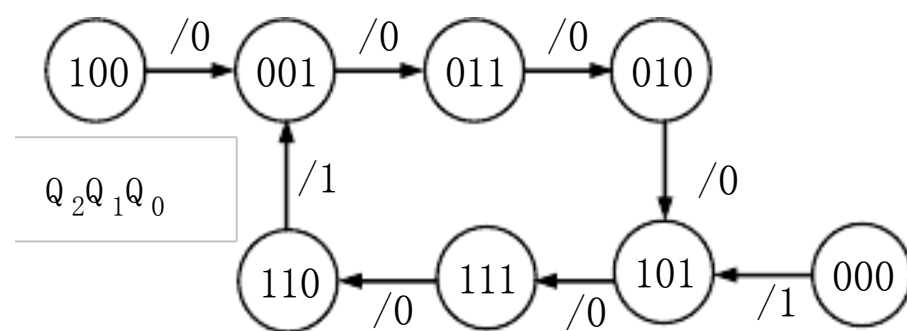


图 7-11 例 7-9 电路状态转换

例 7-10 试用 PLA 和 D 触发器设计一个时序逻辑电路，电路的状态转换图如图 7-12 所示。当输入控制变量 $C=0$ 时，状态变化按顺时针方向，当 $C=1$ 时，状态变化方向按逆时针方向。 Q_{cc} 为电路的进位位。

(1) 写出电路的驱动方程 D_0 、 D_1 、 D_2 和输出方程 Q_{cc} 。

(2) 画出相应的电路图。

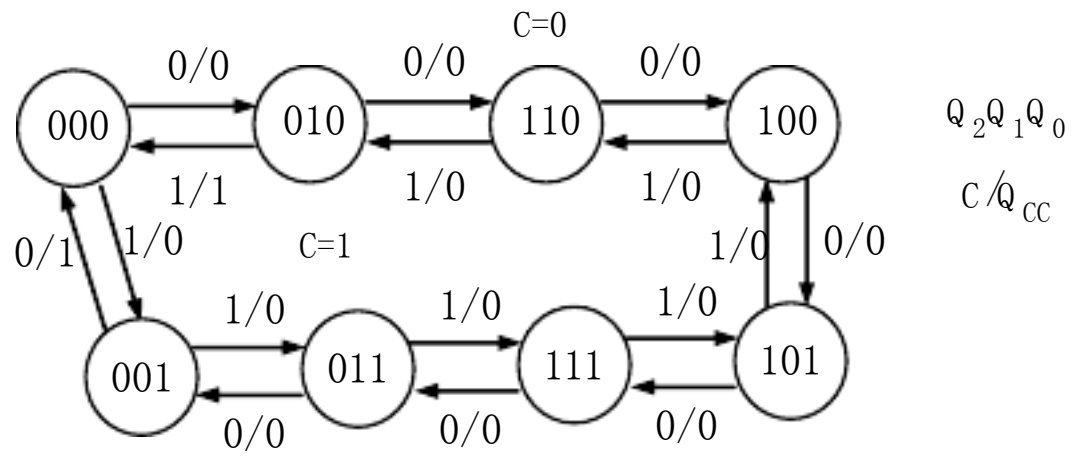


图 7-12 例 7-10 状态转换图

解：(1) 按同步可控计数器的设计方法，可得图 7-13 所示的 Q^{n+1} 卡诺图。

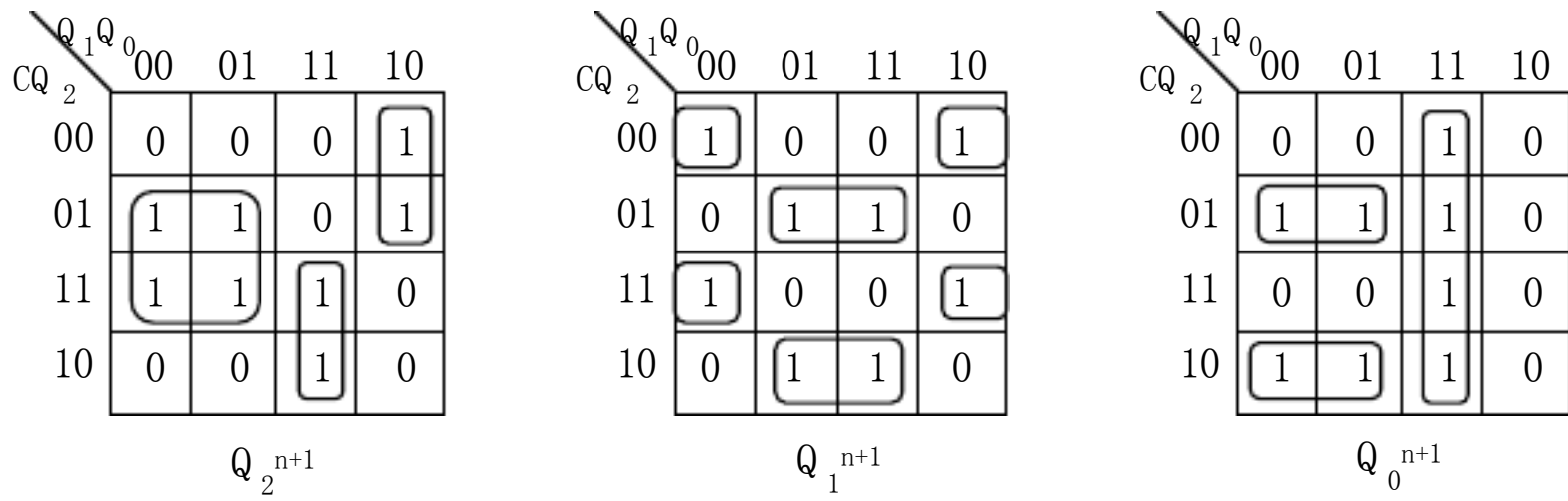


图 7-13 例 7-10 Q_2^{n+1} 、 Q_1^{n+1} 、 Q_0^{n+1} 、卡诺图

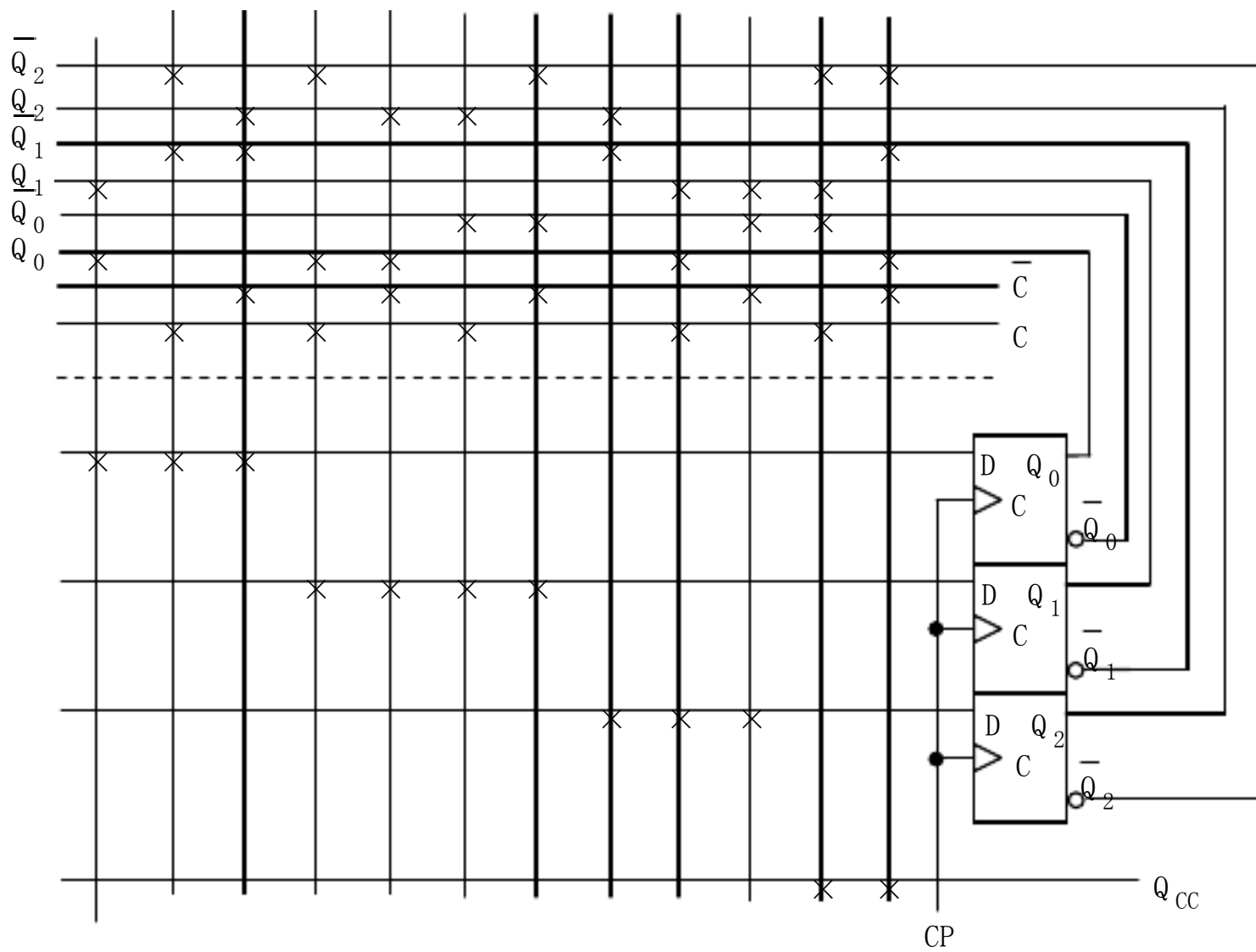


图 7-14 例 7-10 PLA 阵列图

按卡诺图合并规则，可得各触发器的驱动方程，即有

$$D_0 = \overline{Q_1}Q_0 + \overline{CQ_2}\overline{Q_1} + \overline{CQ_2}Q_1$$

$$D_1 = \overline{CQ_2}Q_0 + \overline{CQ_2}Q_0 + \overline{CQ_2}Q_0 + \overline{CQ_2}Q_0$$

$$D_2 = \overline{Q_2}Q_1 + CQ_1Q_0 + \overline{CQ_1}Q_0$$

电路的输出方程 Q_{cc} 为

$$Q_{cc} = \overline{CQ_2}Q_1Q_0 + \overline{CQ_2}Q_1Q_0$$

(2) 用 PLA 和 D 触发器设计的电路图如图 7-14 所示。

例 7-11 用 PLA 和 D 触发器设计 8421BCD 码转换七段字形译码器。

解：七段字形如图 7-15 (a) 所示，七段笔划形状与数字的关系示于图 7-15 (b)，根据图 7-15 作真值表如表 7-7 所示。其中“1”为该亮的字段，“0”表示不该亮的字段。再按多输出函数的简化方法，先在每个卡诺图上按单输出函数进行简化，然后再在 7 个函数的卡诺图上找出公共项，见图 7-16 所示。

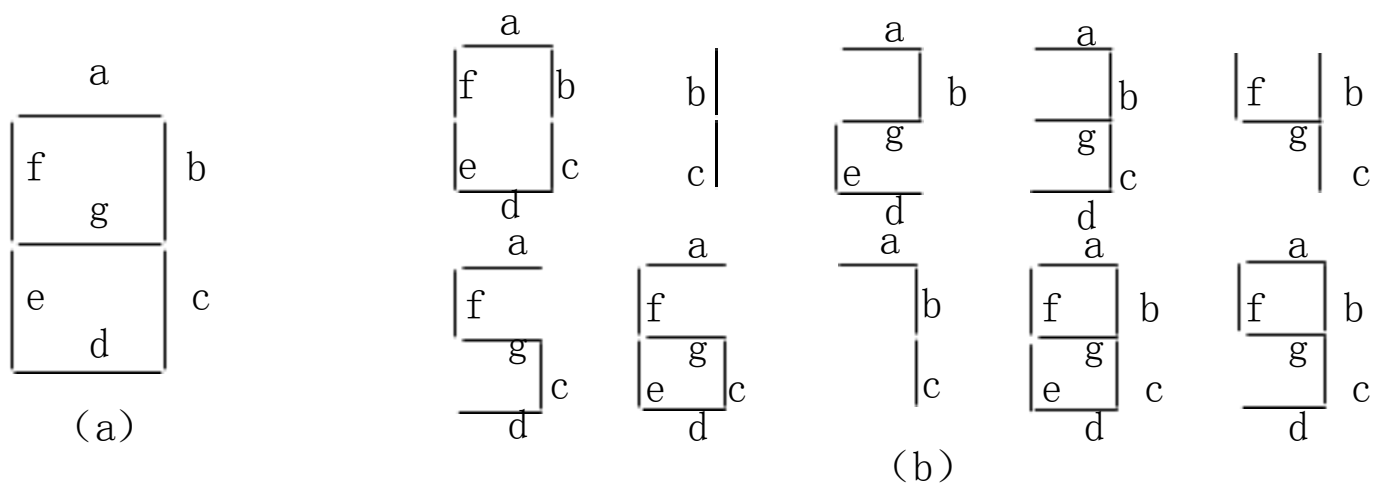


图 7-15 七段字形图与数字关系

表 7-7 8421BCD 码对应七段译码器真值表

B_8	B_4	B_2	B_1	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ
1	0	1	1	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ
1	1	0	0	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ
1	1	0	1	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ
1	1	1	0	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ
1	1	1	1	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ

通过卡诺图简化后得到逻辑表达式如下：

$$a = \overline{B_8} + \overline{B_4}\overline{B_1} + B_2B_1 + \overline{B_4}B_2B_1 + B_4B_2\overline{B_1}$$

$$b = \overline{B_4}\overline{B_1} + \overline{B_2}B_1 + B_2B_1 + \overline{B_4}B_1$$

$$c = \overline{B_2}B_1 + \overline{B_4}B_1 + \overline{B_2}B_1 + \overline{B_4}B_2B_1 + \overline{B_4}B_2\overline{B_1}$$

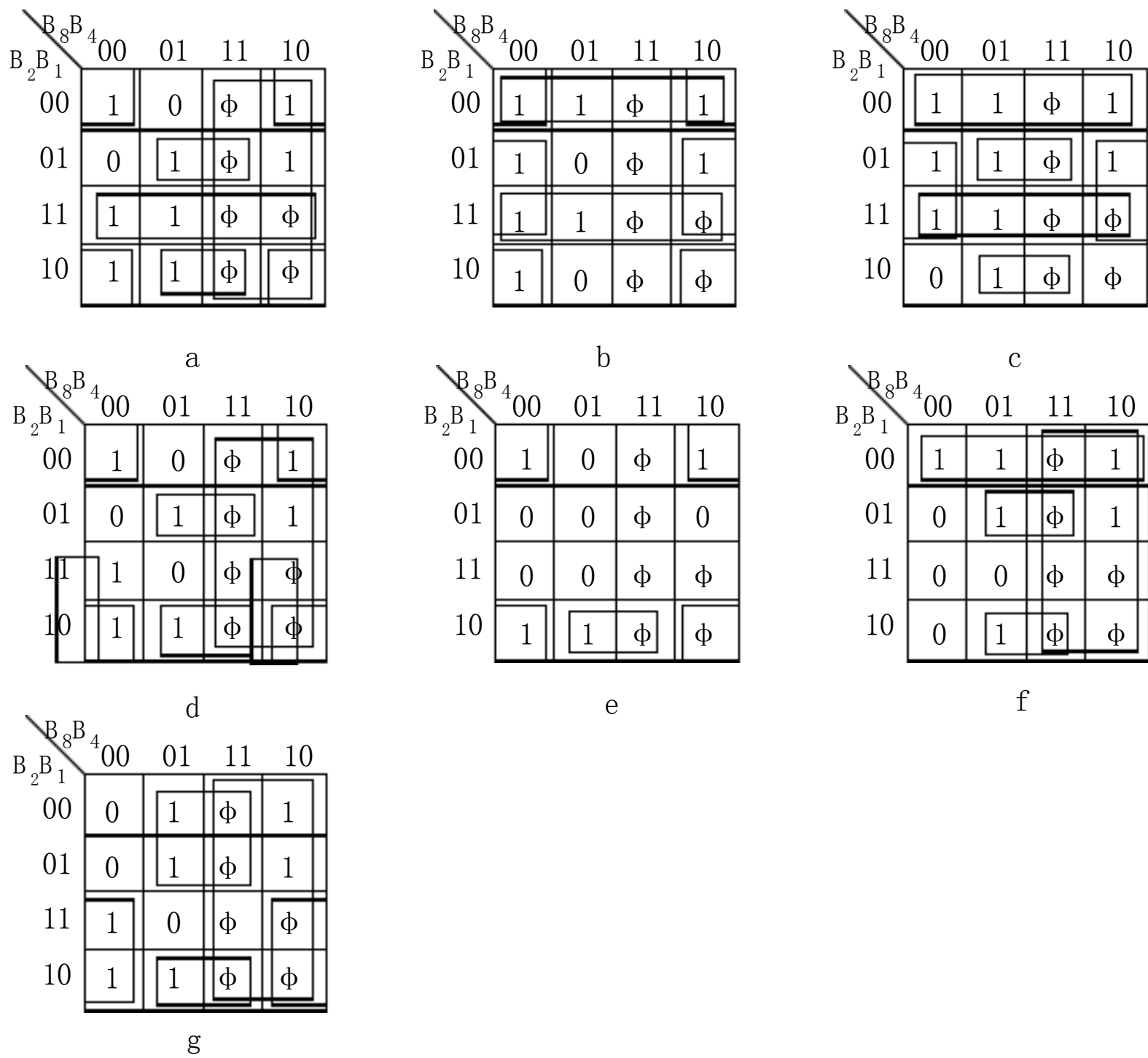


图 7-16 例 7-10 a、b、c、d、e、f、g 卡诺图

$$d = \overline{B_8} \overline{B_4} B_2 + \overline{B_8} B_4 \overline{B_2} + B_8 \overline{B_4} B_2 + B_8 B_4 \overline{B_2} + B_8$$

$$e = \overline{B_8} \overline{B_4} + B_8 B_4 \overline{B_2}$$

$$f = \overline{B_8} + \overline{B_4} \overline{B_2} + \overline{B_4} B_2 B_1 + B_8 \overline{B_2} \overline{B_1}$$

$$g = \overline{B_8} + \overline{B_4} \overline{B_2} + \overline{B_4} B_2 + B_8 \overline{B_2} \overline{B_1}$$

这些表达式中总共有九个与项，即

$$P_0 = \overline{B_8}; P_1 = \overline{B_4} \overline{B_2}; P_2 = \overline{B_4} B_2$$

$$P_3 = \overline{B_8} \overline{B_4} \overline{B_2}; P_4 = \overline{B_8} \overline{B_4} B_2; P_5 = \overline{B_8} B_4 \overline{B_2}$$

$$P_6 = \overline{B_8} B_4 B_2; P_7 = \overline{B_8} \overline{B_4} \overline{B_2}; P_8 = \overline{B_8} \overline{B_4} B_2$$

由各个与项构成的函数为：

$$a = P_0 + P_1 + P_2 + P_3 + P_4$$

$$b = P_1 + P_2 + P_5 + P_6$$

$$c = P_2 + P_3 + P_4 + P_5 + P_6$$

$$d = P_1 + P_3 + P_4 + P_7 + P_8$$

$$e = P_1 + P_4$$

$$f = P_0 + P_3 + P_4 + P_5$$

$$g = P_0 + P_4 + P_7 + P_8$$

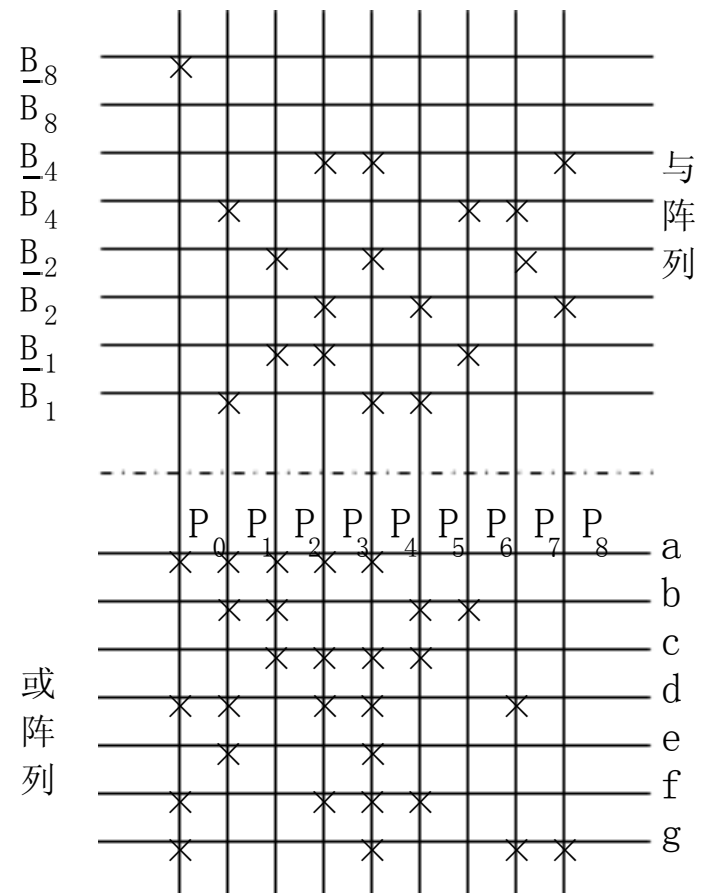


图 7-17 例 7-11 阵列逻辑图

根据上述逻辑函数和各个与项构成的阵列逻辑图示于图 7-17。

例 7-12 用 PAL 器件设计一个 3 位循环码计数器，状态表如表 7-8 所示。RESET 复位信号可使该计数器初始化为 $Q_3Q_2Q_1=000$ 状态。 \overline{OC} 是使能控制信号。

解：(1) 选择 PAL 器件：3 位循环码计数器有 3 个状态变量 $Q_3Q_2Q_1$ 输出，根据题意该计数器只有一个输入复位信号 RESET。显然，可采用带有寄存器输出的 PAL16R4 芯片进行设计。PAL16R4 引脚及引脚分配如图 7-18 (a) 所示。

(2) 电路状态方程：由状态表可得状态方程

$$Q_3^{n+1} = \overline{R}Q_3^nQ_1^n + \overline{R}Q_2^n\overline{Q_1}^n$$

$$Q_2^{n+1} = \overline{R}Q_2^n\overline{Q_1}^n + \overline{R}Q_3^nQ_1^n$$

$$Q_1^{n+1} = RQ_3^nQ_1^n + \overline{R}Q_2^nQ_1^n$$

(3) 按照图 7-18 (b) 对 PAL16R4 编程，即可实现该循环码计数器。本例中若用中、小规模通用器件设计，则至少要使用 1 片 74175 和 2 片 7400，这里一片 PAL16R4 就代替了若干片 SSI、MSI 集成器件，显然是一种较理想的实施方案。

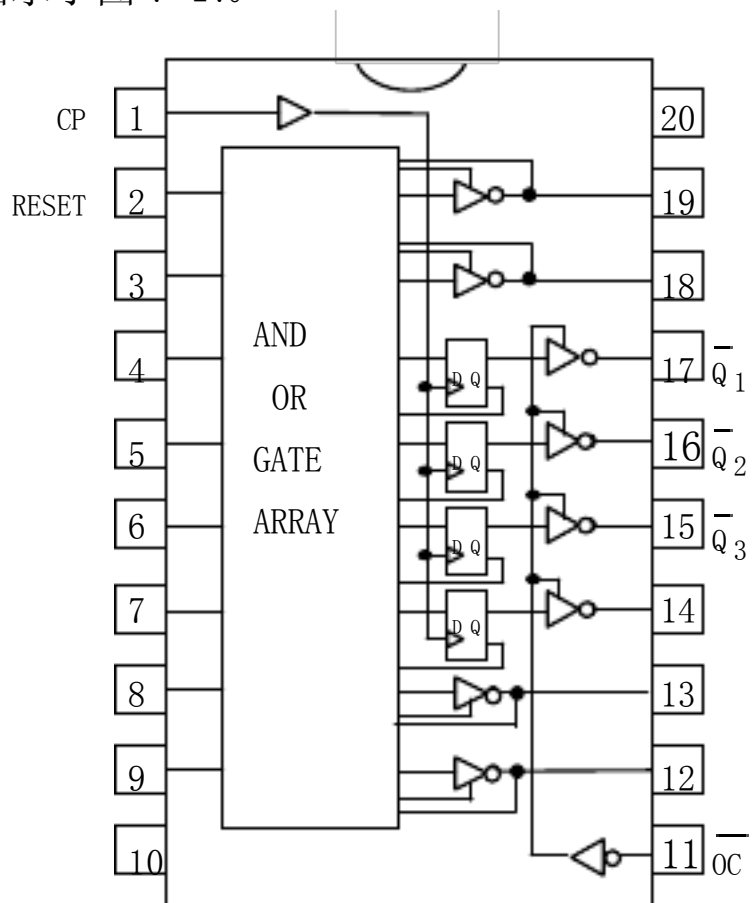


图 7-18 (a) PAL16R4 实现循环码计数器的引脚分配图

表 7-8 3 位循环码计数器状态表

CP	R (RESET)	Q_3^n	Q_2^n	Q_1^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}
↑	0	0	0	0	0	0	1
↑	0	0	0	1	0	1	1
↑	0	0	1	1	0	1	0
↑	0	0	1	0	1	1	0
↑	0	1	1	0	1	1	1
↑	0	1	1	1	1	0	1
↑	0	1	0	1	1	0	0
↑	0	1	0	0	0	0	0
↑	1	Φ	Φ	Φ	0	0	0

例 7-13 用一片 GAL16V8，实现图 7-19 (a) 组合逻辑电路。

解：该电路包括 6 个基本逻辑门，它们是：或非门、或门、与非门、与门、异或门和同或门。逻辑方程是：

$$F_1 = \overline{A_1} \cdot \overline{B_1}$$

$$F_2 = \overline{A_2} + \overline{B_2}$$

$$F_3 = \overline{A_3} + \overline{B_3}$$

$$F_4 = \overline{A_4} \cdot \overline{B_4}$$

$$F_5 = \overline{A_5} \oplus \overline{B_5}$$

$$F_6 = \overline{A_6} \odot \overline{B_6}$$

实现这些逻辑方程共需 12 个输入端和 6 个输出端，因此可将 GAL16V8 的 6 个逻辑宏单元 OLMC 组态为专用组合输出结构，引脚 13、14、15、16、17 和 18 分别为输出端 F_6 、 F_5 、 F_4 、 F_3 、 F_2 和 F_1 ；考虑到待实现的电路为 12 输入的组合函数，故除了用专门的输入端 2、3、

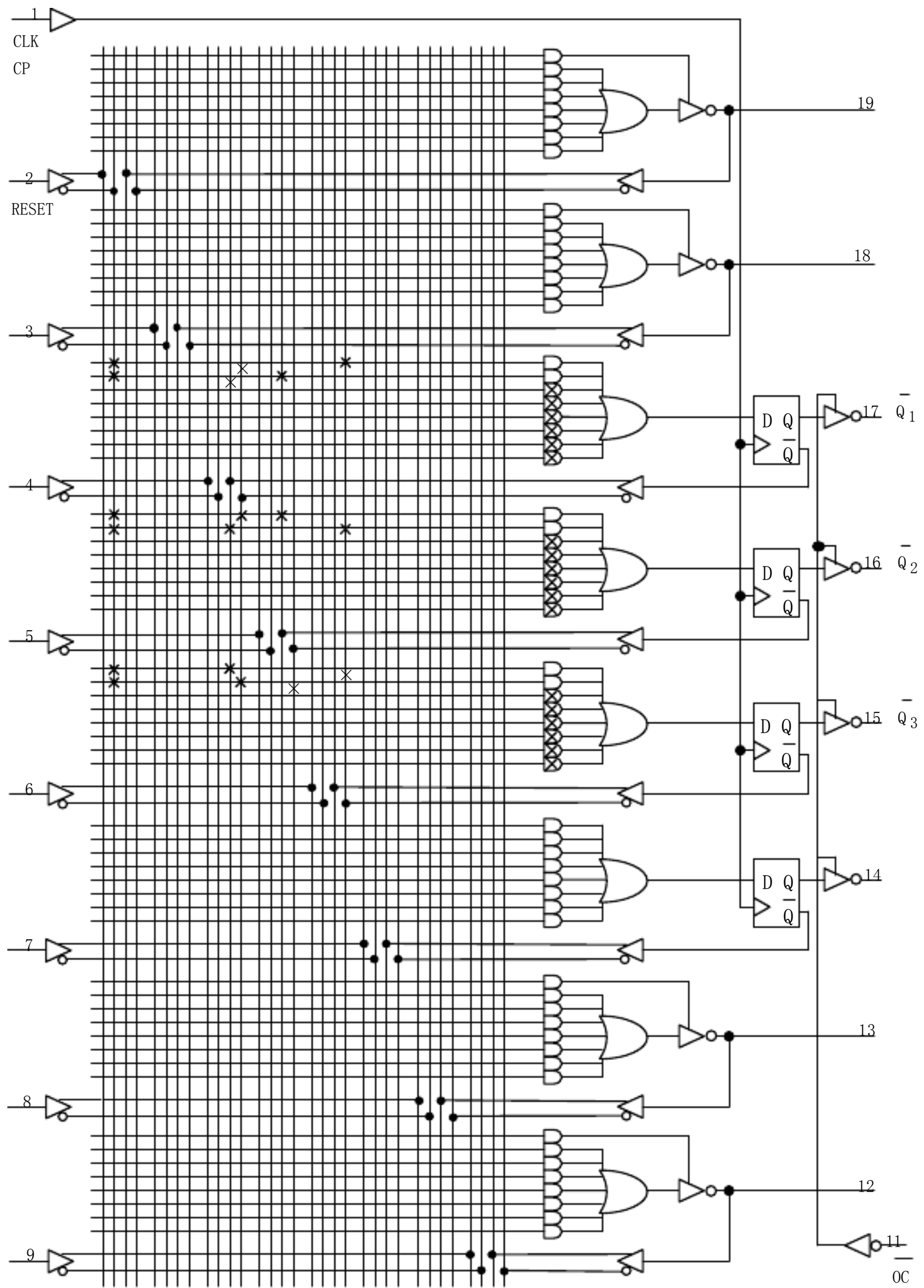


图 7-18 (b) PAL16R4 实现循环码计数器的阵列图

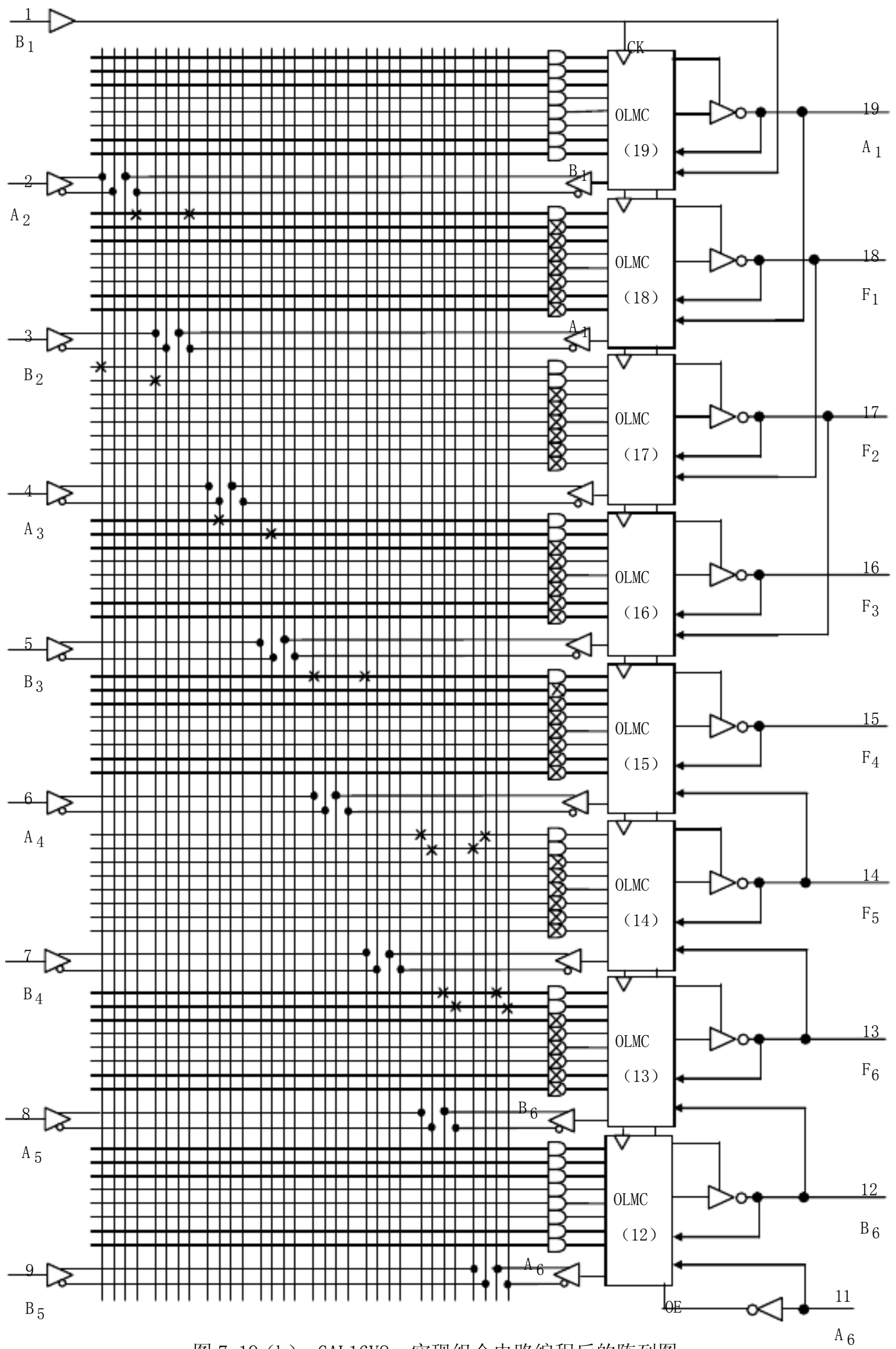


图 7-19 (b) GAL16V8 实现组合电路编程后的阵列图

4、5、6、7、8 和 9 作为信号输入外，又将 CP（引脚 1）和 OE（引脚 11）作为两个信号输入端，同时还将剩余的两个 OLMC（引脚 12、19）组态为专用输入结构，恰好满足了待设计电路的输入端数量的要求，用 GAL16V8 实现编程后的阵列图如图 7-19（b）所示。

例 7-14 图 7-20 是 XC4000E 系列芯片中的一个 CLB，它实现了 16×2 电平单口 RAM，试分析其逻辑功能。

解：组合逻辑输入 $F_1 \sim F_4$ 和 $G_1 \sim G_4$ 作为 RAM 的地址线输入，可以通过地址的改变来选中存储单元的信息，此时 CLB 控制信号 Din、S/R 和 H_1 分别作为数据信号 D_1 、 D_0 和写使能信号 WE， F' 和 G' 作为 RAM 的数据输出。

当 RAM 写数据时，写使能信号（WE）为高电平选通写译码器，使写译码器使能。地址 $F_1 \sim F_4$ 通过地址译码选中 16×1 阵列中的某一单元。如当 $F_1 \sim F_4$ 地址为 0000 时，并且 WE 有效，数据 D_0 进入 16×1 存储矩阵最上面的存储单元。当 RAM 读出数据时，写使能信号 WE 为低电平，禁止数据写入。地址信号直接到数据输出选择器选中所需单元从 F' 读出，形成 16×1 电平单口 RAM，同样 $G_1 \sim G_4$ 对应 G' 组成另一个 16×1 电平单口 RAM。

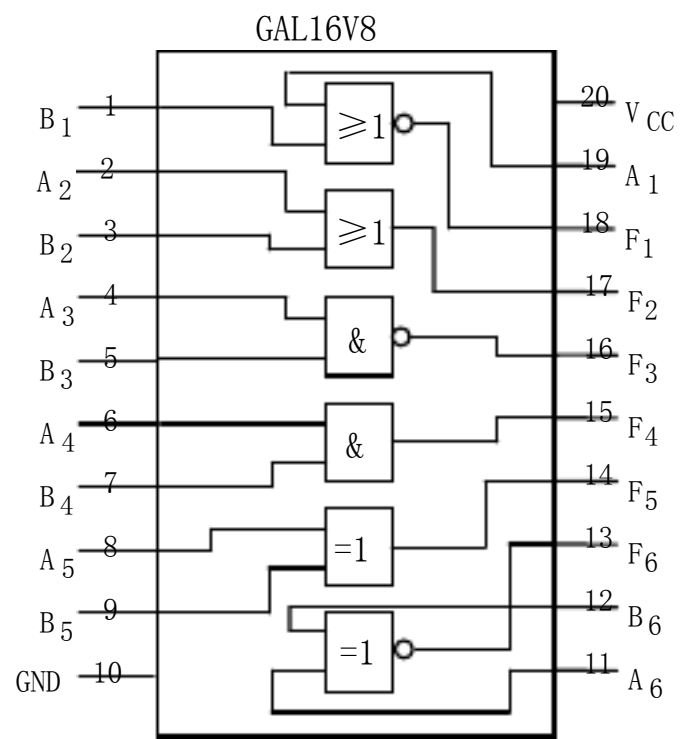


图 7-19 (a) GAL16V8 实现例 7-13 组合电路的引脚分配图

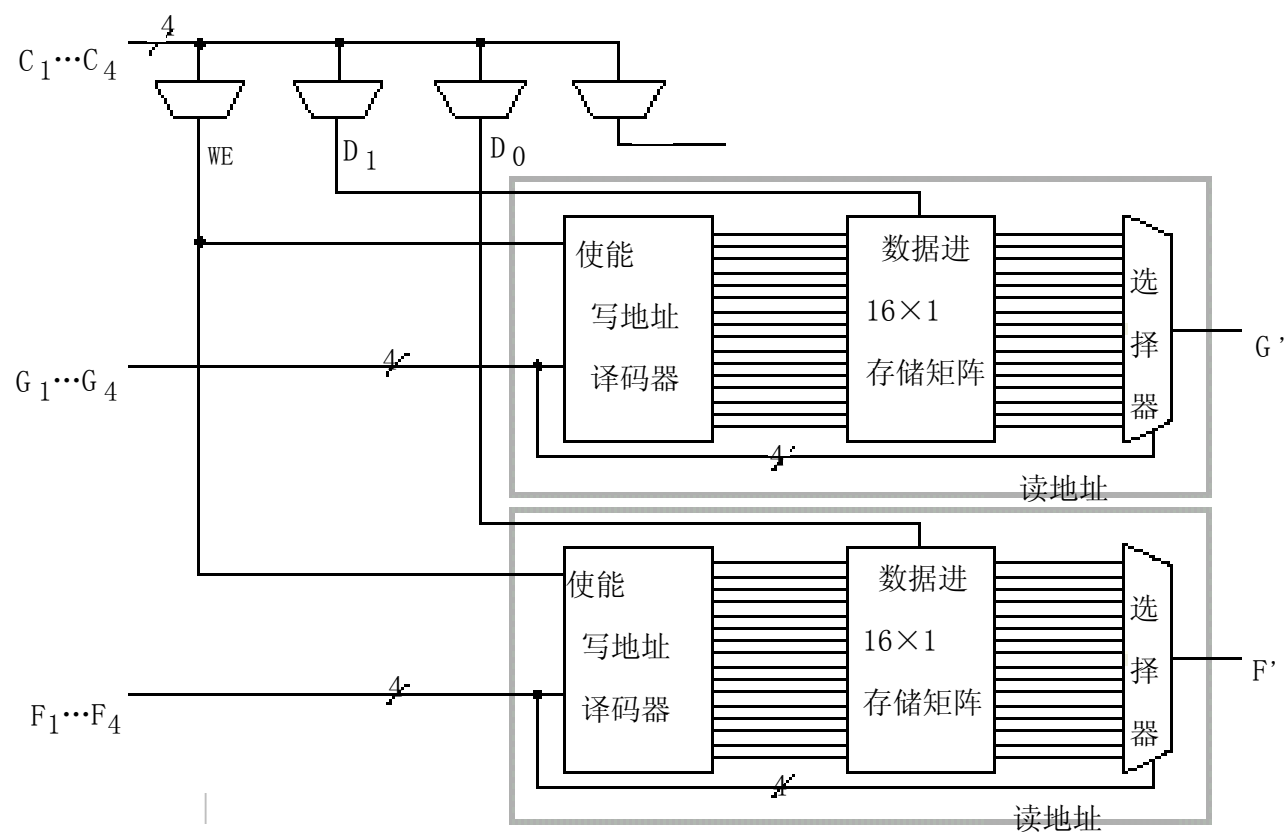


图 7-20 16×2 电平触发单口 RAM

例 7-15 试用 XC4000E 系列芯片中的 CLB 组成 16×8 的电平触发单口 RAM。

解：在例 7-14 中的一个 CLB 可以形成 16×2 的电平触发单口 RAM，如要组成 16×8 的电平触发单口 RAM 则需要 4 个 CLB，如图 7-21 所示。

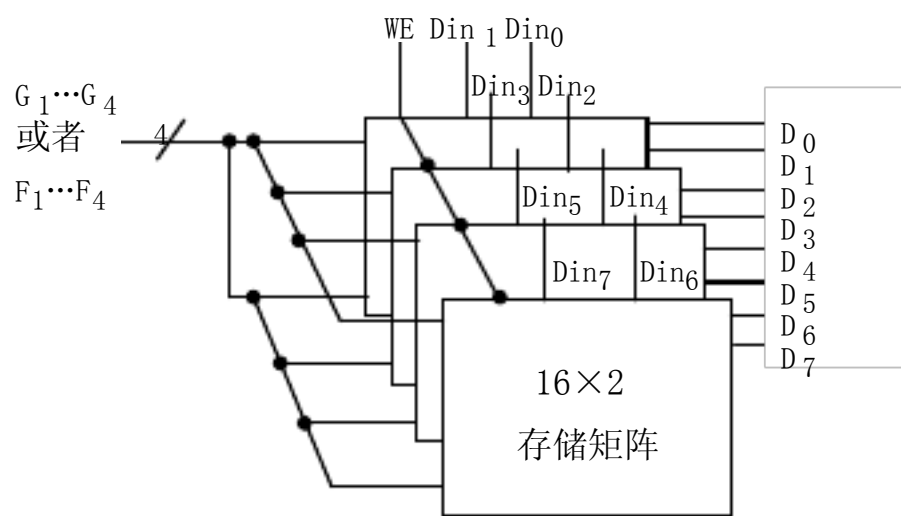


图 7-21 16×8 电平触发单口 RAM

例 7-16 试用 5G2112 256×4bitRAM 构成容量为 512×8bitRAM。5G2112 的逻辑示意图如图 7-22 所示。

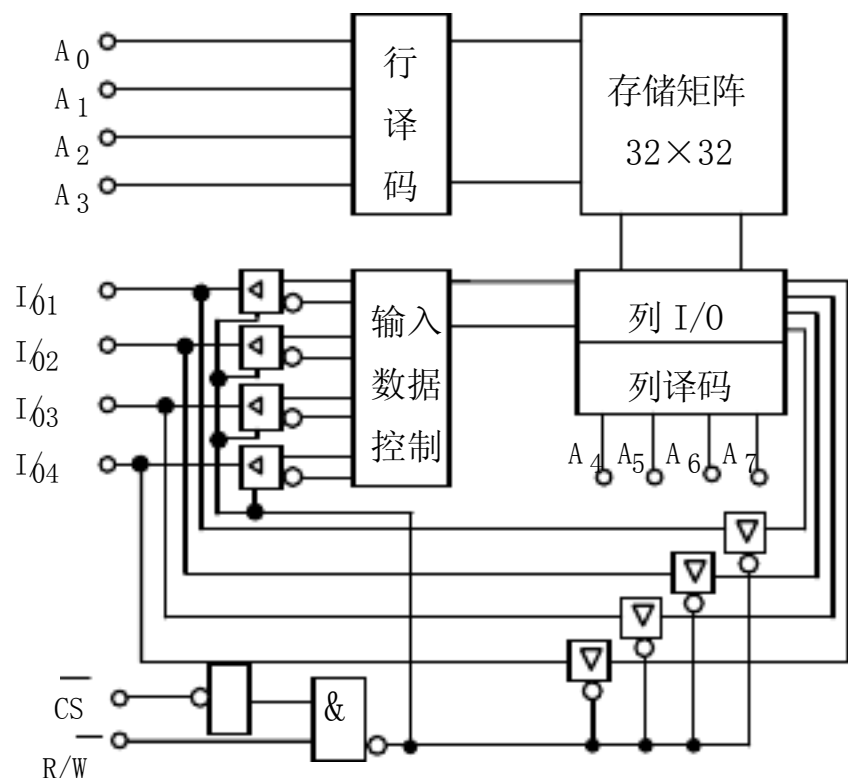


图 7-22 5G2112 逻辑示意图

解：5G2112 是 256×4bit 静态 RAM，由图 7-22 看出，5G2112 是采用二元寻址和三态输出结构。片选信号 \overline{CS} 是低电平有效，当电路选通之后，若要写入，则令读/写控制端 $R/\overline{W}=0$ ，输入三态门打开，数据便写入存储器；与此同时，输出三态门被关闭，切断了输出与数据总线的联系。若要读出，则令 $R/\overline{W}=1$ ，输入三态门被关闭，而输出三态门被打开，因而存储数据被读出（注意：输入三态门是高电平选通，输出三态门是低电平选通）。

用四片 5G2112RAM 构成 512×8bitRAM 的电路连接示意图如图 7-23 所示。

先进行字长扩展，为此将片（1）与片（2）的地址 $A_0 \sim A_7$ 、片选 \overline{CS} 、读/写控制端 R/\overline{W} 对应并接起来，数据端 I/O 分别接到数据总线的低四位 $D_1 \sim D_4$ 和高四位 $D_5 \sim D_8$ ，这样一来就扩展成 256×8bitRAM；片（3）与片（4）也进行同样的处理。然后再将它们进行地址扩展，这里不必增加译码器，因为只增加一个地址变量 A_8 ，所以用一个非门，将非门的输入端 A_8 和输出端 $\overline{A_8}$ 分别接到片（1）、（2）和片（3）、（4）的 \overline{CS} 端，并将 R/\overline{W} 端、 $A_0 \sim A_7$ 、数据端 I/O 等对应连接，这样就得到 512×8bitRAM。

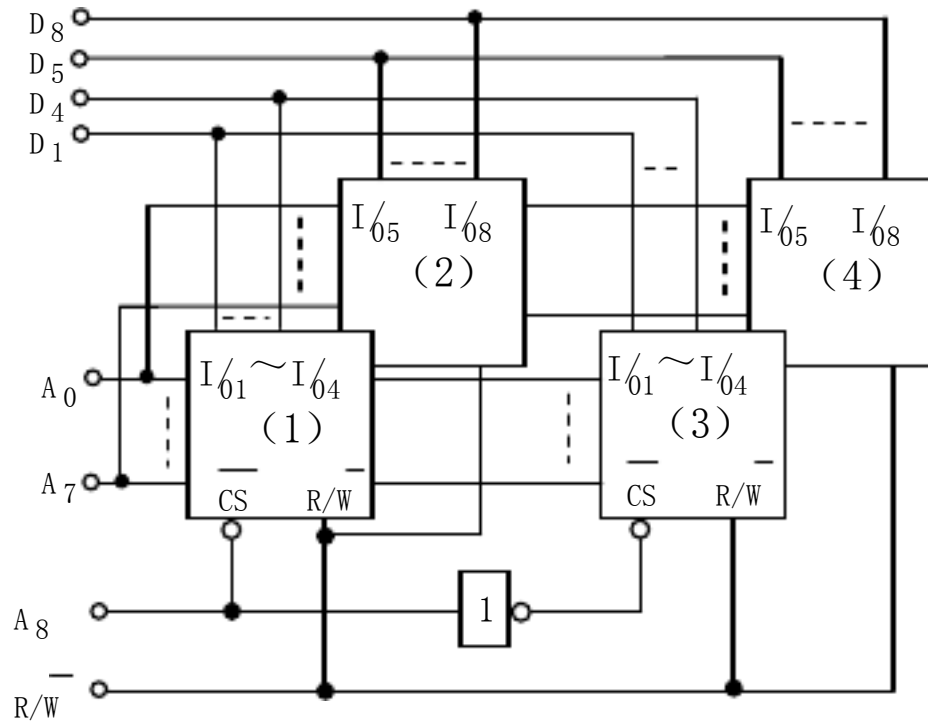


图 7-23 512×8bitRAM 电路连接示意图

例 7-17 RAM6116 功能分析和应用。

CMOS 静态 RAM6116 (2K×8) 芯片的逻辑符号如图 7-24 所示。

试用 6116 芯片和 CT4138 组成 8K×8 的 RAM，内存单元的寻址范围为：8000~87FFH，9800~9FFFH，C000~C7FFH，C800~CFFFH。画出包括和 CPU 接线在内的 8K×8 的 RAM 的接线图。

解：用 2K×8 的 RAM 组成 8K×8 的 RAM 时，需四片芯片。

将 16 条地址线和内存单元地址排列成表 7-9 所示的形式。由表 7-9 中内存单元地址和地址线的对应关系，可将 RAM6116 的 11 根地址线 $A_{10} \sim A_0$ 直接和 CPU 的低 11 位地址线 $A_{10} \sim A_0$ 相连。其他的接线可参阅图 7-25。图中 CPU 的高 5 位地址线 $A_{15} \sim A_{11}$ 和 CT4138 的选通输入和地址输入端相连。其中 A_{15} 接 S_1 端， A_{13} 接 S_3 ， S_2 ， A_{14} 、 A_{12} 、 A_{11} 分别接 $A_2 \sim A_0$ 。CT4138 的输出端 \bar{Y}_0 、 \bar{Y}_1 、 \bar{Y}_5 、 \bar{Y}_6 分别接四片 RAM6116 的 CS 端。这样四片 RAM6116 的内存单元地址即为：8000~87FFH，9800~9FFFH，C000~C7FFH，C800~CFFFH，满足了题目的要求。

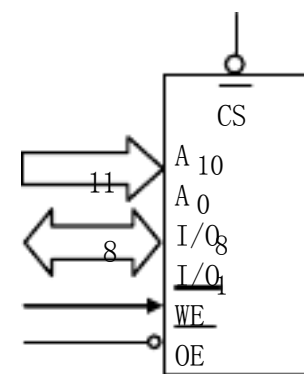


图 7-24 RAM6116 逻辑符号

表 7-9 内存单元对应 CPU 分配图

地址线		A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
内存单元地址	8000H	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	87FFH	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
	9800H	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
	9FFFH	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
	C000H	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	C7FFH	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1
	C800H	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	CFFFH	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1

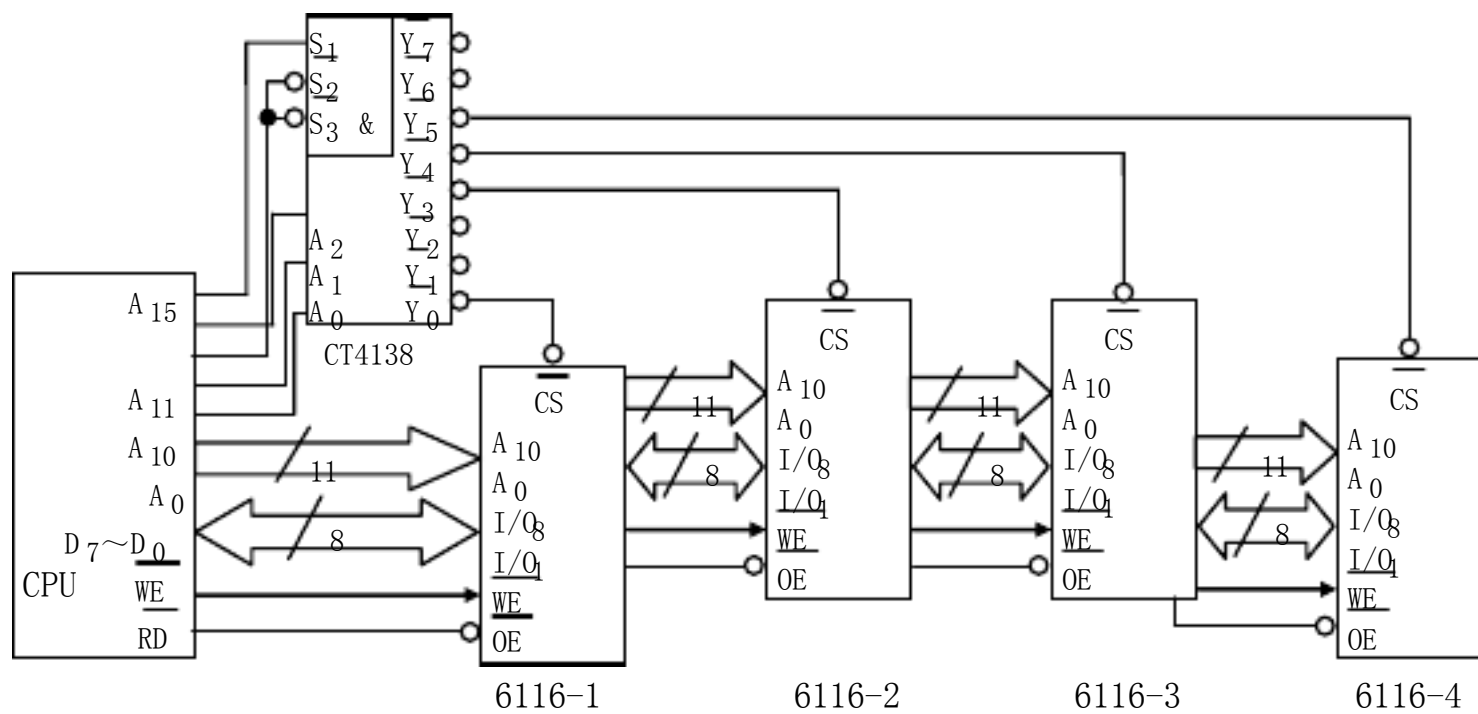


图 7-25 RAM6116 和 CPU 连接图

RAM6116 芯片的 \overline{OE} 为输出允许端，在读出内存单元的内容时， \overline{OE} 应加低电平，故应和 CPU 的 \overline{RD} 端相接， \overline{RD} 在读操作时发出低电平，能满足对 6116RAM 读取数据的时序要求。

需注意：RAM 扩展时，除考虑容量（位扩展，字扩展或位、字同时扩展）和内存单元地址的分配方式（采用全译码方式或部分译码方式，后者可能出现地址重叠）以外，还必须考虑速度的配合，即 CPU 和 RAM 的存取速度要合适，这主要涉及 CPU 和 RAM 之间存取数据的时序问题，此例未作讨论。

例 7-18 试分析图 7-26 所示的随机存取存储器（RAM）电路。

- (1) 存储器的总容量和字长是多少？
- (2) 指出当 $R/\overline{W}=1$ ，地址码为 16H 时，哪些 RAM 芯片将数据送到数据线上。
- (3) 指出 RAM0，RAM1，RAM2，RAM3 的存储地址范围各是多少？

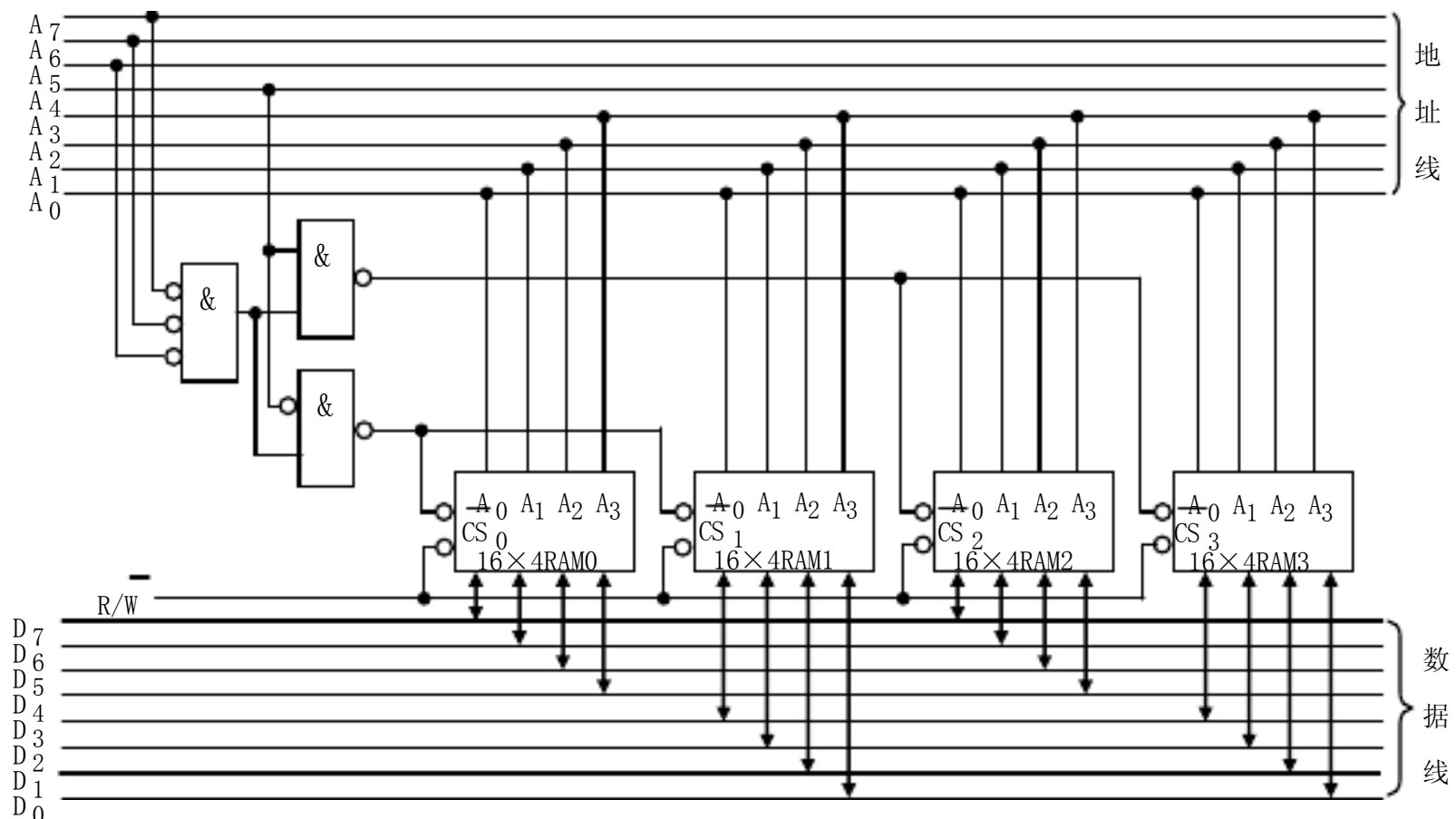


图 7-26 由 RAM 芯片组成的存储器

解：本题采用 4 片 16×4 的 RAM 芯片组成一个容量为 32×8 位的存储器。由于 RAM 芯片容量为 16×4 位，字数及字长均不能满足要求，因此，将 4 片容量为 16×4 的 RAM 进行两两组合，进行位扩展，组成两个容量为 16×8 的 RAM。然后再用这两个容量为 16×8 的 RAM 进行字扩展，组成容量为 32×8 的 RAM。位扩展及字扩展均通过 RAM 片选信号 \overline{CS} 来扩展。

数据线有 8 位 ($D_7 \sim D_0$)，地址线有 8 位 ($A_7 \sim A_0$)，地址范围从 00H 到 FFH，故最多有 256 个字。地址线的高 4 位 $A_7 \sim A_4$ 通过门电路构成两个 16×8 RAM 的片选信号，低四位 $A_3 \sim A_0$ 则作为 16×8 RAM 自身的地址。片选信号由下式决定：

$$\overline{CS}_0 = \overline{CS}_1 = A_7 + A_6 + A_5 + A_4$$

$$\overline{CS}_2 = \overline{CS}_3 = A_7 + A_6 + A_5 + A_4$$

可见只有当 $A_7 A_6 A_5 A_4 = 0000$ 时， $\overline{CS}_0 = \overline{CS}_1 = 0$ RAM0，RAM1 选中工作；当 $A_7 A_6 A_5 A_4 = 0001$ 时， $\overline{CS}_2 = \overline{CS}_3 = 0$ RAM2，RAM3 选中工作。

(1) 此 RAM 电路的总容量为 32×8 ，字长为 8 位。

(2) $R/\overline{W}=1$ 表示发出读存储器的命令，当地址为 10H 时，即

$$A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0 = 00010000 \text{ 所以 } \overline{CS}_0 = \overline{CS}_1 = 1 \text{ RAM0，RAM1 被封锁。}$$

$\overline{CS}_2 = \overline{CS}_3 = 0$ RAM2，RAM3 被选中工作，并将地址 10H 的 8 位数据读出后送到数据线上。

(3) RAM0，RAM1 的存储地址范围为 00H ~ 0FH；RAM2，RAM3 的存储地址为 10H ~ 1FH。

第三节 习题解答

习题 7-1 试述 PROM、EPROM 和 E²PROM 的特点。

答：共同之处：(1) 均为可以进行编程的只读存储器；

(2) 属于非易失性元件，即掉电之后，所存储的信息不丢失；

(3) 利用了浮栅编程技术；

(4) 芯片为有限次写入。

不同之处：

(1) PROM 为一次编程器件；EPROM 为电编程、紫外线擦除器件、E²PROM 是电编程、电擦写器件；

(2) PROM 编程元件为容丝或者是 PLICE 介质；EPROM 是用 SIMOS 浮栅作为编程元件；E²PROM 是用 FLOTOX 浮栅作为编程元件。

习题 7-2 试述非易失性元件的种类及特点。

答：非易失性元件应该是可编程逻辑器件 PLD，包括只读存储器 ROM、编程只读存储器 PROM、电编程、紫外线擦除只读存储器 EPROM、电编程、电擦除只读存储器 E²PROM、在系统编程 ISP、编程逻辑阵列 PLA、编程阵列逻辑 PAL、通用阵列逻辑 GAL 和高密度可编程逻辑器件 HDPLD。它们有相似的基本结构：输入、与阵列、或阵列和输出电路。

按编程部位可将 PLD 分为：

(1) 与阵列可编程、或阵列固定，代表器件为 ROM、PROM、EPROM、E²PROM。

(2) 与阵列可编程、或阵列可编程，代表器件为 PLA。

(3) 与阵列固定、或阵列可编程，代表器件为 PAL、GAL 和 HDPLD 等。

按编程方法可将 PLD 分为：

(1) 固定编程：是由集成电路制造厂家编程后出厂的 ROM 芯片。

(2) 一次编程：是由用户进行编程，且只能编程一次。它的编程单元是容丝或者是反容丝等结构，代表器件为 PROM 、 PAL 。

(3) 多次编程：是由用户多次进行编程。它的编程单元采用的是浮栅技术，代表器件为 EPROM 、 E²PROM 、 PLA 、 GAL 和 HDPLD 等。

特点：(1) 减小系统体积；(2) 增强逻辑设计的灵活性；(3) 缩短设计周期；(4) 提高系统处理速度；(5) 降低系统成本；(6) 提高系统的可靠性；(7) 系统具有加密功能。

习题 7-3 简述 EPROM 实现不同规模逻辑函数的特点。

答：EPROM 是与阵列固定，输入信号的每个组合都固定连接（不管这个组合是否会被使用），所以与门阵列为全译码阵列，它经常被用来作为数据存储器。还可方便地用 EPROM 来实现简单的逻辑函数。若实现复杂的逻辑函数，则会随着输入信号的增加，使得芯片面积增大，利用率和工作速度降低等情况发生，例如，输入信号有 10 个，所需要的函数乘积项仅有 40 个的时候，由于固定的与阵列所产生的 10 个信号的乘积项有 $2^{10}=1024$ 个，所以将所有的乘积项（1024）减去所需的乘积项（40）就有 984 个乘积项被空闲。实际上，大多数组合逻辑函数的最小项不超过 40 个，则使得 PROM 芯片的面积利用率不高，功耗增加。

习题 7-4 用 EPROM 实现下列多输出函数

$$\text{解： } F_1 = \overline{A} \overline{B} \overline{C} + \overline{B} C + AB$$

$$F_2 = \overline{A} + B + C$$

$$F_3 = \overline{A} B + \overline{A} \overline{B}$$

$$F_4 = (A + B + C)(A + B + C) + \overline{A} BC$$

解：输入信号是 3 个 A、B、C，输出是 4 个 F_1 、 F_2 、 F_3 、 F_4 。阵列规模为 $6 \times 8 + 8 \times 4$ 。

输入 A、B、C 在与阵列进行全译码，产生所有的最小项。

在或阵列上，输出 F_1 、 F_2 、 F_3 、 F_4 根据给定的逻辑函数，得到最小项。然后在或阵列进行编程，得到图 7-27 所示的与或阵列图。

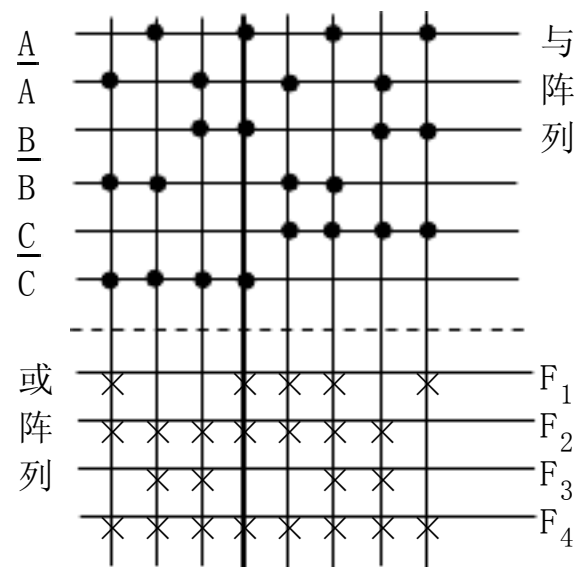


图 7-27 习题 7-4ROM 阵列图

习题 7-5 用适当规模的 EPROM 设计两位二进制数乘法器，输入乘数和被乘数分别为

表 7-10 习题 7-5 电路真值表

A_2	A_1	B_2	B_1	C_4	C_3	C_2	C_1
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

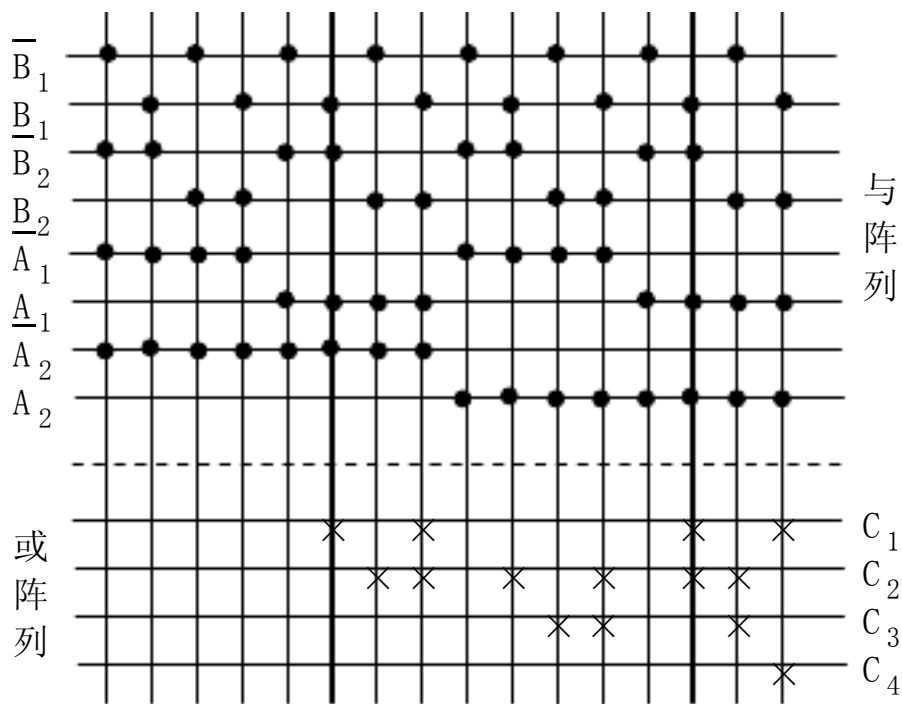


图 7-28 习题 7-5 二位乘法器阵列图

A_2A_1 和 B_2B_1 , 输出为四位二进制数 C_4, C_3, C_2, C_1 , 并且说明 EPROM 的容量。

解: 本题是利用紫外线擦除、电编程的 EPROM 实现组合逻辑电路的设计问题。依照所要求的电路功能, 可按两个 2 位二进制数的乘法运算列出真值表。参照例题 7-4 中表 7-3 和图 7-4 改写 A_1A_0 为 A_2A_1 (作为乘数)、 B_1B_0 为 B_2B_1 (作为被乘数) 和输出乘积 $F_3F_2F_1F_0$ 为 $C_4C_3C_2C_1$, 即列出电路真值表如表 7-10 所示, 利用 EPROM 实现的乘法器的与或阵列图如图 7-28 所示。

习题 7-6 用 EPROM 实现以下码制的变换

(1) 8421 码至 2421 码的变换;

(2) 8421 码至 5421 码的变换。

解: 利用 EPROM 实现 8421 码至 2421 码的变换和 8421 码至 5421 码的变换也是采用与阵列固定或阵列编程。表 7-11 列出了 8421BCD 码转换成 5421BCD 码和 2421BCD 码的对应真值表。

(1) 参照表 7-11, 画出 8421BCD 码转换成 5421BCD 码阵列逻辑图, 如图 7-29 所示。

(2) 参照表 7-11, 画出 8421BCD 码转换成 5421BCD 码阵列逻辑图, 如图 7-30 所示。

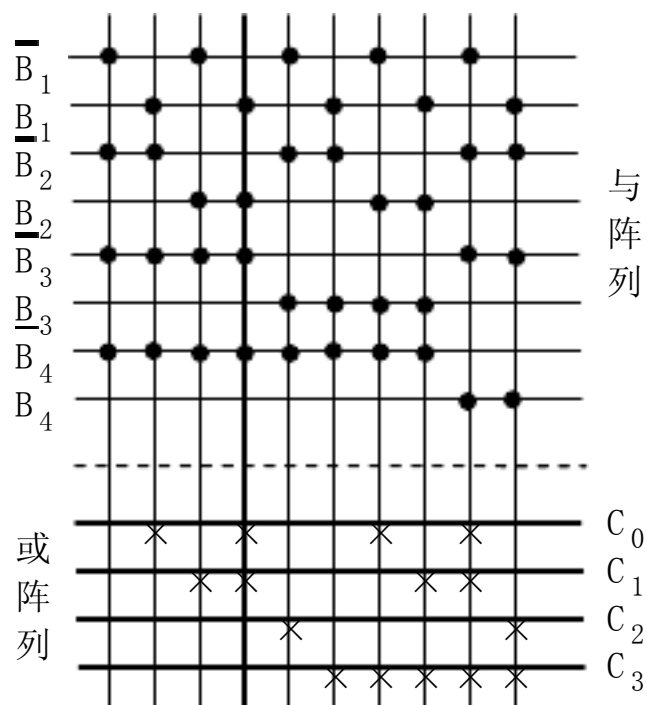


图 7-29 8421 至 2421 码阵列图

8421				5421				2421			
B_3	B_2	B_1	B_0	C_3	C_2	C_1	C_0	C_3	C_2	C_1	C_0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0	1	0	0	0
0	0	1	1	0	0	1	1	1	0	0	1
0	1	0	0	0	1	0	0	1	0	1	0
0	1	0	1	1	0	0	0	1	0	1	1
0	1	1	0	1	0	0	1	1	1	0	0
0	1	1	1	1	0	1	0	1	1	0	1
1	0	0	0	1	0	1	1	1	1	1	0
1	0	0	1	1	1	0	0	1	1	1	1

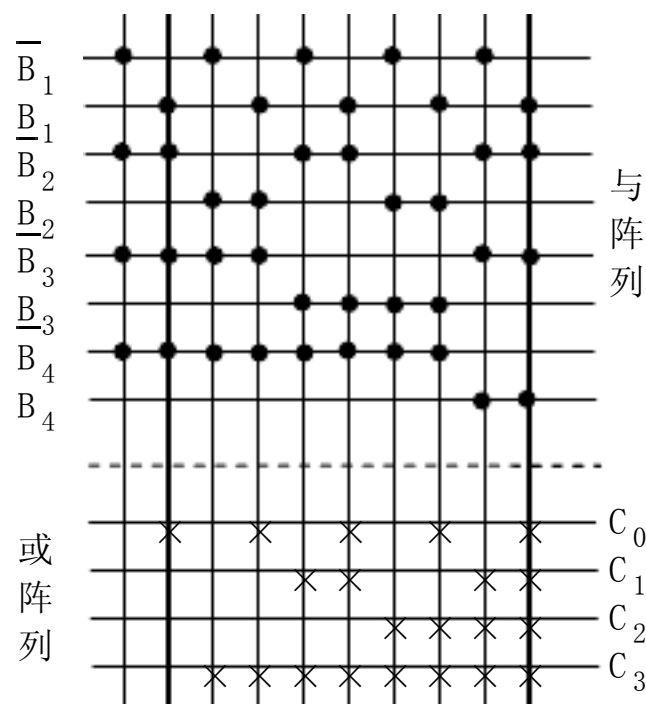


图 7-30 8421 至 5421 码阵列图

习题 7-7 试问 256 字 \times 32 位的 EPROM 的地址线、数据线、字线各有多少根?

答: 256 字 \times 32 位的 EPROM 的地址线为 8 根、数据线为 32 根、字线 256 根。

习题 7-8 用 EPROM2716 构成 4K \times 8 位的 EPROM, 共需多少片? 画出扩展的 EPROM 逻辑图。

解: EPROM2716 是 2K \times 8 位的 EPROM, 若扩展成 4K \times 8 位的 EPROM, 只需扩展地址位, 将地址位加倍, 共需 2 片 EPROM2716。扩展的 EPROM 逻辑图如图 7-31 所示。

习题 7-9 试用 EPROM 设计一字符发生器, 发生的字符为 H。

解: 字符发生器是显示器中常用的逻辑部件。它将各种字母、数字及符号预先存储在 ROM 中, 只要给出适当地址码, 就能将这些字符读出来, 并驱动显示器显示这些字符。图 7-32 给

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/395323010031012011>