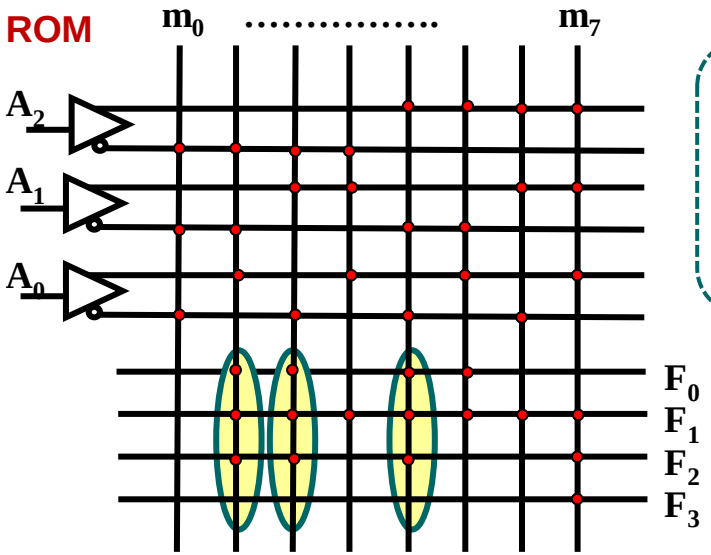
Unit 13

——Programmable Logic Devices

张彦航

School of Computer Science
Zhangyanhang@hit.edu.cn

PLA及其应用



ROM的缺点

没有充分利用半导体材料的面积，限制了使用的灵活性。

1.没有使用的最小项也占有存储单元

2.相同的内容，占用多个存储单元

ROM

A_2	A_1	A_0	F_3	F_2	F_1	F_0
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	0	1	1	1
0	1	1	0	0	1	0
1	0	0	0	1	1	1
1	0	1	0	0	1	1
1	1	0	0	0	1	0
1	1	1	1	1	1	0

PLA产生的思想

将内容相同的存储单元用一个存储单元来代替（让几个地址码读出同一存储单元的内容）

可编程逻辑阵列（PLA）的特点：

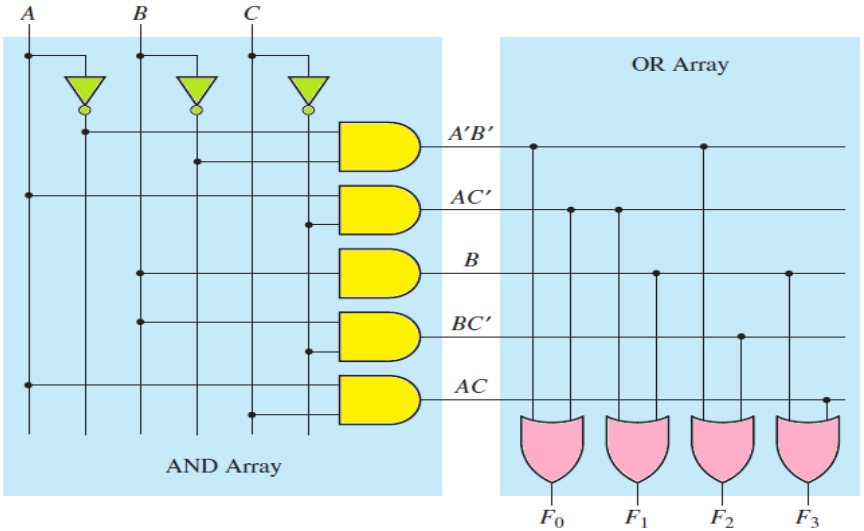
1. 与阵和或阵都可编程，每个字线不一定是完全最小项，且字数少于2^n。
2. 地址和字之间没有一一对应关系，因此一个地址可同时访问两个或两个以上的字。如：存储单元AB'CD和AD都可以用地址1011读出
3. 必须对表达式化简，即存储矩阵中是化简压缩的内容，与真值表不再有一一对应关系。
4. FPLA（现场可编程逻辑阵列）包含记忆元件（触发器网络）
5. FPLA既能实现组合逻辑又能实现时序逻辑

PLA及其应用

PLA产生的思想

将内容相同的存储单元用一个存储单元来代替（让几个地址码读出相同的内容）

输入地址000和001，读出的内容都是：1010



Product Term	Inputs A B C	Outputs F_0 F_1 F_2 F_3
$A'B'$	0 0 –	1 0 1 0
AC'	1 – 0	1 1 0 0
B	– 1 –	0 1 0 1
BC'	– 1 0	0 0 1 0
AC	1 – 1	0 0 0 1

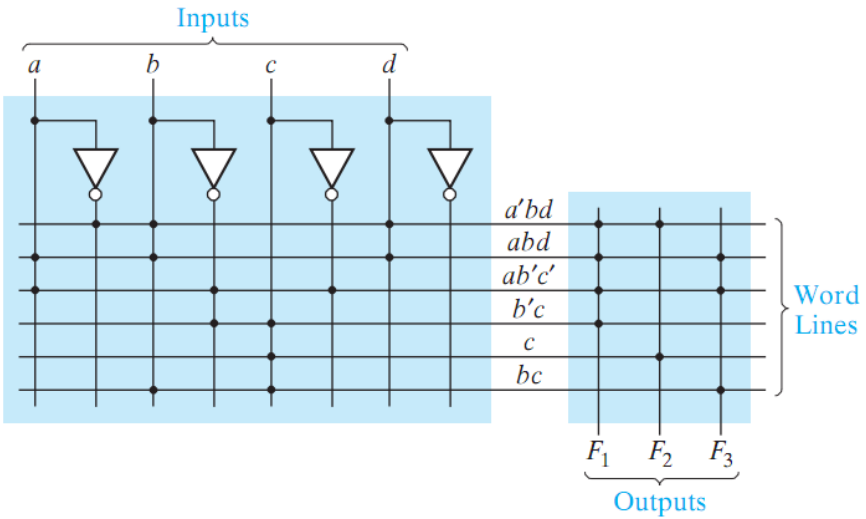
$F_0 = A'B' + AC'$

$F_1 = AC' + B$

$F_2 = A'B' + BC'$

$F_3 = B + AC$

PLA及其应用



$$f_1 = a'bd + abd + ab'c' + b'c$$
$$f_2 = c + a'bd$$
$$f_3 = bc + ab'c' + abd$$

a b c d	f_1 f_2 f_3
0 1 – 1	1 1 0
1 1 – 1	1 0 1
1 0 0 –	1 0 1
– 0 1 –	1 0 0
– – 1 –	0 1 0
– 1 1 –	0 0 1

对于ROM和PROM一个地址只能选中一个存储单元

If *abcd* =0001, 没有字线被选中

If *abcd* =1001, 只有第3行字线被选中，$f_1f_2f_3$=101.

If *abcd*= 0111,第1、5、6行字线同时被选中.

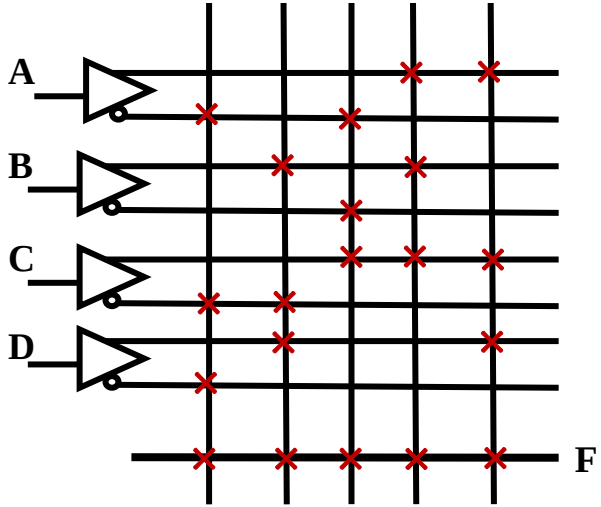
PLA：一个地址（如abcd=0111）能同时选中多个存储单元

PLA及其应用

Example

例1：利用PLA设计组合逻辑函数

$F(ABCD)=\sum(0,2,3,4,5,11,13,14,15)$



$F=\bar{A}\bar{C}\bar{D}+B\bar{C}D+\bar{A}\bar{B}C+ABC+ACD$

方法

1. 化简待设计组合逻辑函数为最简与或式。
2. 若最简表达式中包含某与项，则画出该与项对应的字线，并在或阵列输出线与该字线交点处打X

PLA: 需要5 个存储单元

PROM: 需要9 个存储单元

AB \ CD	00	01	11	10
00	1	0	1	1
01	1	1	0	0
11	0	1	1	1
10	0	0	1	0

PLA及其应用

Example

例2：利用PLA设计1位全加器

S_i

a_i \ b_ic_{i-1}	00	01	11	10
0	0	1	0	1
1	1	0	1	0

C_i

a_i \ b_ic_{i-1}	00	01	11	10
0	0	0	1	0
1	0	1	1	1

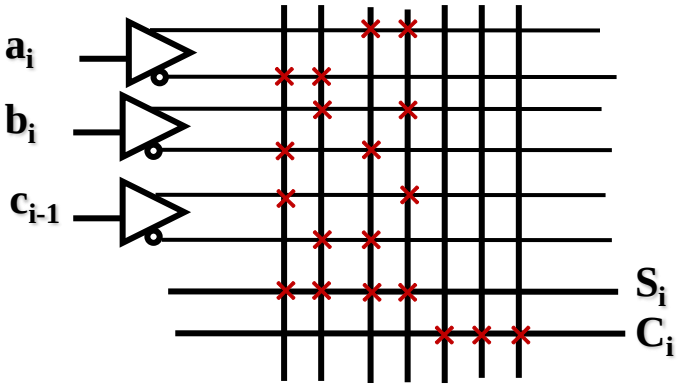
$$S_i = \bar{a}_i\bar{b}_ic_{i-1} + \bar{a}_ib_i\bar{c}_{i-1} + a_i\bar{b}_i\bar{c}_{i-1} + a_ib_ic_{i-1}$$

$$C_i = a_ic_{i-1} + a_ib_i + b_ic_{i-1}$$

全加器设计汇总

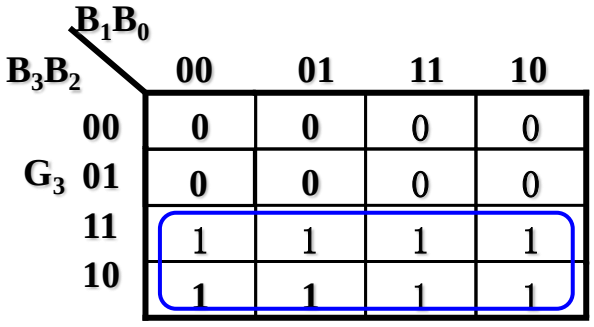
方法1：利用两个半加器
方法2：利用单一逻辑门（如与非门）
方法3：利用译码器芯片74138
方法4：利用数据选择器芯片74153
方法5：利用PROM
方法6：利用PLA
方法7：利用基本逻辑门（与、或、非）
方法8：利用异或门、与门、非门、或门等

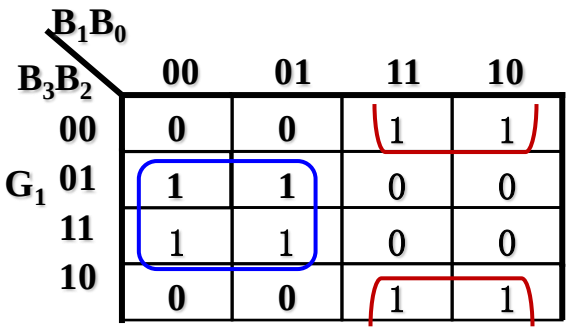
设计方案不唯一

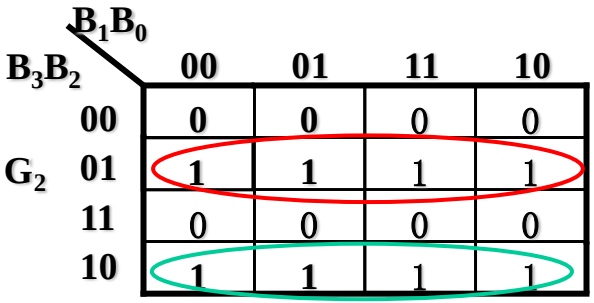


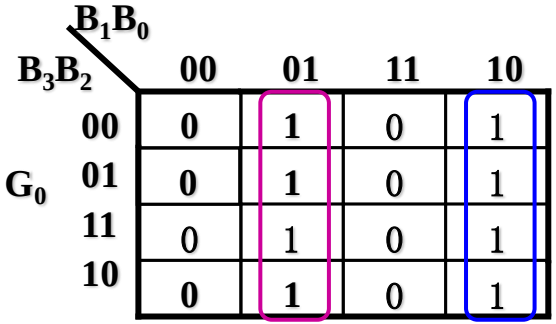
Example

例3：利用PLA设计4位二进制到格雷码转换器









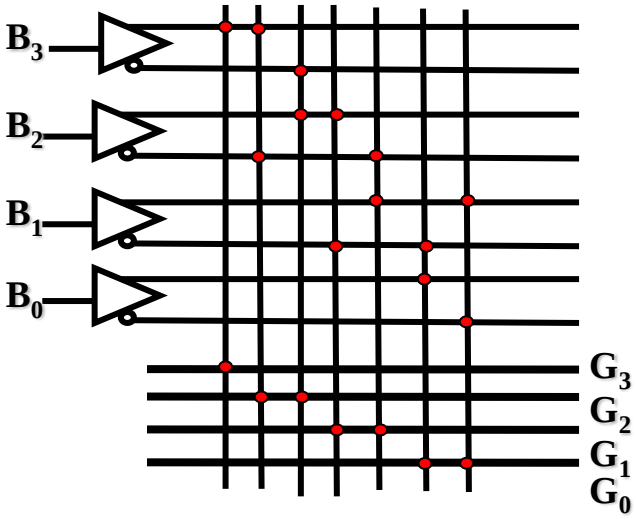
二进制数（存储地址）				葛雷码（存储数据）			
B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

四位二进制码转换为葛雷码的真值表

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

例3：利用PLA设计4位二进制到格雷码转换器

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



用PLA实现

用PROM实现