

# 数字逻辑电路总复习

# 总成绩评定

- 课程的考核和评分方式
  - 理论考试（期末考试）占80%；
  - 平时作业占20%；

# 考试题型

- 填空或选择（10分，每个1分）
- 化简（20分，每道题5分）
- 组合逻辑设计与分析（25分左右）
- 时序逻辑设计与分析（30分左右）
- 可编程逻辑设计（0~15分）

# 前言

- 通过学习在计算机系统中用到的典型逻辑电路的设计分析，达到：
  - 1、掌握在逻辑设计中设计和分析的基本方法。
  - 2、掌握在逻辑设计中应当注意的问题。
  - 3、掌握在计算机系统中常用IC器件的性能及设计方法。
  - 4、掌握基本硬件系统的调试和测试方法。
  - 5、熟练掌握基本工具（万用表、示波器、逻辑分析仪.....）

# 课程基本情况

## ● 教学的具体内容

- 前言和逻辑代数：基本概念，卡诺图，表格化简法等；
- 门电路：三种基本逻辑运算（与或非），TTL、门电路静态特性和动态特性；
- 组合逻辑电路：定义、分析方法、设计方法，组合电路中的竞争—冒险，常用的逻辑功能的实现；
- 时序电路：触发器的原理、功能、特性，时序电路的特点、分析方法和设计方法，典型的时序逻辑电路；
- 可编程逻辑器件：可编程逻辑的工作原理及应用。

# 数字逻辑电路总复习

- 四句话加一个忠告

# 本课程总体情况

- 数字：编码，真值表，状态表.....
- 逻辑：表达式，状态方程.....
- 电路：符号图，逻辑图.....

# 数字逻辑电路总复习

- 四句话

- 全书一句话
- 组合逻辑一句话
- 时序逻辑一句话
- 可编程逻辑一句话

- 一个忠告



# 数字逻辑电路总复习

- 四句话

- ⇒ ➤ 全书一句话

- 组合逻辑一句话

- 时序逻辑一句话

- 可编程逻辑一句话

- 一个忠告

# 一句话包含全书

- 表达式+触发器是全书的核心原理
  - 组合逻辑：表达式
  - 时序逻辑：表达式+触发器
- 集成电路：表达式+触发器的具体应用
- 需要注意的是：表达式可以包含所有分析和设计的原理，但具体实现技术却各有特点

# 表达式的化简

- 化简方法

- 手工化简

- ✓ 公式法

- ✓ 卡诺图

- 计算机化简

- ✓ 表格法

- 实际设计中并不是越简越好

# 数字逻辑电路总复习

- 四句话

- 全书一句话

- ⇒ ➤ 组合逻辑一句话

- 时序逻辑一句话

- 可编程逻辑一句话

- 一个忠告

# 组合逻辑一句话

- 表达式可以表示全部组合逻辑

- 译码器

- 编码器

- 数据选择器

- 数据比较器

- 奇偶校验器

- 加法器

- 可以用其他方式表示：真值表，逻辑图.....

# 组合逻辑

- 其他表示方式

- 真值表

- 逻辑图

- 简单的设计形象直观

- 复杂的设计需要仔细分析和仔细设计

# 译码器

- 2—4译码器表达式:

输出表达式

$$Y_0 = \overline{A} \overline{B}$$

$$Y_1 = A \overline{B}$$

$$Y_2 = \overline{A} B$$

$$Y_3 = A B$$

- 3—8译码器, 4—16译码器类似

# 译码器

- 显示译码器

$$a = B D + \overline{A} C + A \overline{B} \overline{C} \overline{D}$$

$$b = B D + A \overline{B} C + \overline{A} B C$$

$$c = \overline{A} B \overline{C} + C D$$

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

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

$$f = A \overline{C} \overline{D} + A B + B \overline{C}$$

$$g = \overline{B} \overline{C} \overline{D} + A B C$$



# 编码器

## ● 4—2编码器:

$$A_0 = I_0 \overline{I_1} I_2 I_3 + I_0 I_1 I_2 \overline{I_3} = \overline{\overline{I_0} I_1 I_2 I_3 + I_0 \overline{I_1} \overline{I_2} \overline{I_3}}$$

$$A_1 = I_0 I_1 \overline{I_2} I_3 + I_0 I_1 I_2 \overline{I_3} = \overline{\overline{I_0} I_1 I_2 I_3 + I_0 \overline{I_1} I_2 I_3}$$

## ● 8421编码器

$$Y_3 = X_8 + X_9$$

$$Y_2 = X_4 + X_5 + X_6 + X_7$$

$$Y_1 = X_2 + X_3 + X_6 + X_7$$

$$Y_0 = X_1 + X_3 + X_5 + X_7 + X_9$$

# 编码器

## ● 8—3优先编码器

$$\overline{A_0} = \overline{7} + \overline{6} \overline{5} + \overline{6} \overline{4} \overline{3} + \overline{6} \overline{4} \overline{2} \overline{1}$$

$$\overline{A_1} = \overline{7} + \overline{6} + \overline{5} \overline{4} \overline{3} + \overline{5} \overline{4} \overline{2}$$

$$\overline{A_2} = \overline{7} + \overline{6} + \overline{5} + \overline{4}$$

$$A_0 = \overline{7 \ 6 \ 5 \ 4 \ 3 \ 2 \ 1 \ 0}$$

$$G_S = \overline{E_0} \overline{\overline{E_i}}$$

# 优先编码功能表

$\overline{E_i}$	0	1	2	3	4	5	6	7	$A_0$	$A_1$	$A_2$	$G_s$	$\overline{E_o}$
0	X	X	X	X	X	X	X	0	0	0	0	0	1
0	X	X	X	X	X	X	0	1	1	0	0	0	1
0	X	X	X	X	X	0	1	1	0	1	0	0	1
0	X	X	X	X	0	1	1	1	1	1	0	0	1
0	X	X	X	0	1	1	1	1	0	0	1	0	1
0	X	X	0	1	1	1	1	1	1	0	1	0	1
0	X	0	1	1	1	1	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
1	X	X	X	X	X	X	X	X	1	1	1	1	1

( $A_2, A_1, A_0$ 用反码编码,  $G_s$ 为编码输出,  $\overline{E_i}$ 为使能输入,  $E_o$  为输出指示)

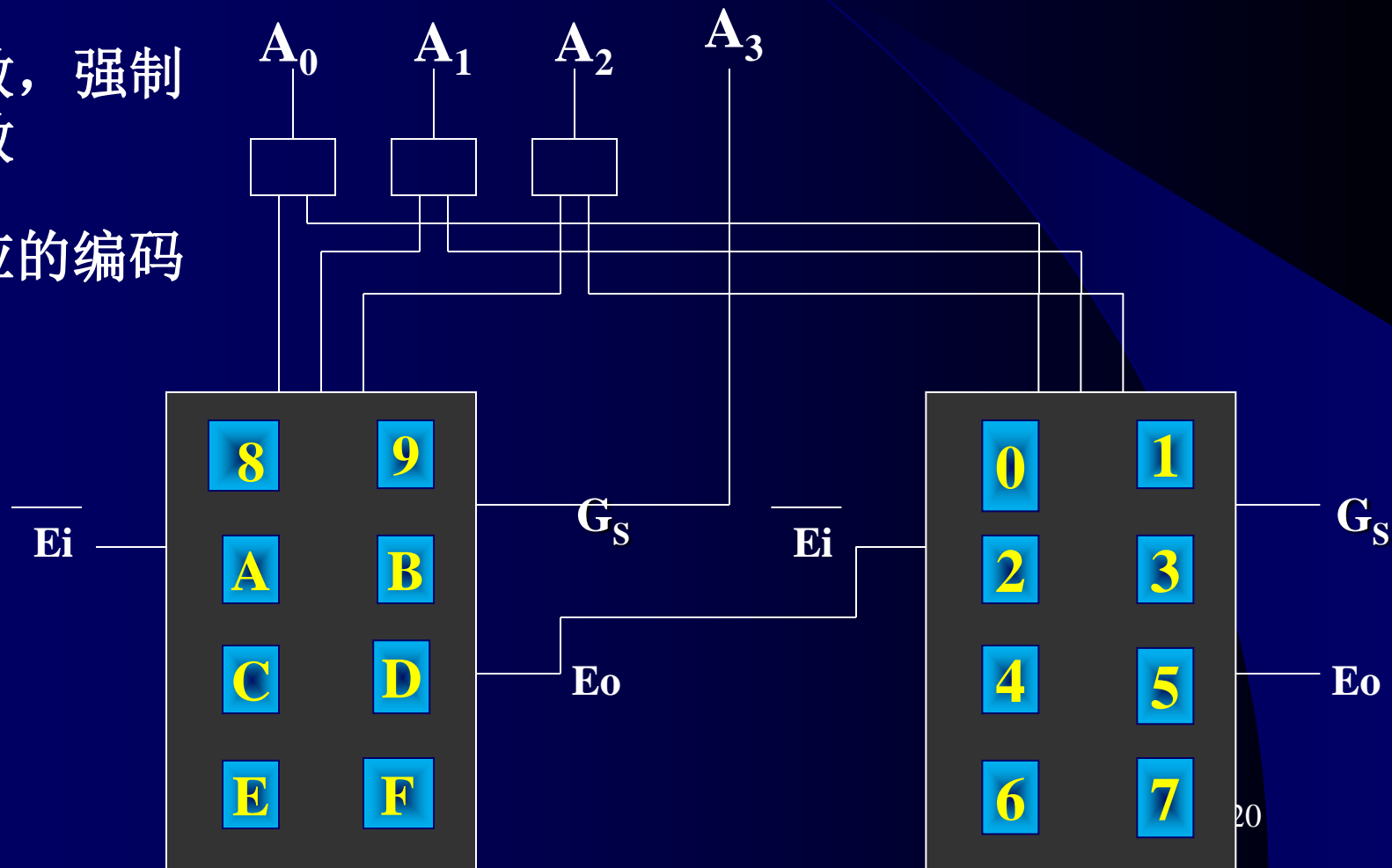
●输出编码为反码, 即用“000”表示7, 用“111”表示0

## 3.3.2 编码器（15）

● 将8-3优先编码器扩展为16-4优先编码器

高位有效，强制  
低位无效

生成相应的编码  
输出



# 数据选择器

- 4选1数据选择器

$$Y = \overline{S_0} \overline{S_1} D_0 + S_0 \overline{S_1} D_1 + \overline{S_0} S_1 D_2 + S_0 S_1 D_3$$

- 8选1, 16选1.....

# 运算器

- 一位全加器

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

$$C_n = X_n Y_n + (X_n + Y_n) C_{n-1}$$

# 运算器

- 4位串行、并行加法器
- 16位串行、并行加法器
- 关键点
  - 减少进位传输延迟的级数，是加法器设计的主要矛盾

# 运算器

## ● 4位并行加法器

➤ 求和:  $F_n = C_{n-1} \oplus (X_n \oplus Y_n)$

➤ 进位产生函数Pi和进位传递函数Gi

✓ 形成条件C1~C4

✓ 得到原理表达式

$$C_1 = G_1 + P_1 C_0$$

$$C_2 = G_2 + P_2 G_1 + P_2 P_1 C_0$$

$$C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 C_0$$

$$C_4 = G_4 + P_4 G_3 + P_4 P_3 G_2 + P_4 P_3 P_2 G_1 + P_4 P_3 P_2 P_1 C_0$$

$$G_i = X_i Y_i$$

$$P_i = X_i + Y_i$$



# 运算器

## ● 4位并行加法器

### ➤ 进位产生函数Pi和进位传递函数Gi

✓ 实用化表达式

$$C_1 = \overline{P_1} + \overline{G_1} \overline{C_0}$$

$$C_2 = \overline{P_2} + \overline{G_2} \overline{P_1} + \overline{G_2} \overline{G_1} \overline{C_0}$$

$$C_3 = \overline{P_3} + \overline{G_3} \overline{P_2} + \overline{G_3} \overline{G_2} \overline{P_1} + \overline{G_3} \overline{G_2} \overline{G_1} \overline{C_0}$$

$$C_4 = \overline{P_4} + \overline{G_4} \overline{P_3} + \overline{G_4} \overline{G_3} \overline{P_2} + \overline{G_4} \overline{G_3} \overline{G_2} \overline{P_1} + \overline{G_4} \overline{G_3} \overline{G_2} \overline{G_1} \overline{C_0}$$

### 3.3.5 运算器

#### ● 16位加法器

##### ➤ 超前进位扩展器的设计

- ✓ 用类似四位快速加法器中C1、C2、C3、C4形成的原理，去形成片间快速进位C4、C8、C12、C16

$$\begin{aligned} C_4 &= (G_4 + P_4G_3 + P_4P_3G_2 + P_4P_3P_2G_1) + P_4P_3P_2P_1C_0 \\ &= \underline{G_{m1}} + \underline{P_{m1}C_0} \end{aligned}$$

$$\underline{G_{m1} = G_4 + P_4G_3 + P_4P_3G_2 + P_4P_3P_2G_1}$$

$$\underline{P_{m1} = P_4P_3P_2P_1}$$

由超前进位扩展器产生

由四位加法器产生

### 3.3.5 运算器

●  $C_8$ 是如何产生的？

$$C_8 = G_{m2} + P_{m2} G_{m1} + P_{m2} P_{m1} C_0$$

$$G_{m2} = G_8 + P_8 G_7 + P_8 P_7 G_6 + P_8 P_7 P_6 G_5$$

$$P_{m2} = P_8 P_7 P_6 P_5$$

由超前进位扩展器产生

由四位加法器产生

### 3.3.5 运算器

●  $C_{12}$ 和 $C_{16}$ 如何产生的？

$$C_{12} = G_{m3} + P_{m3}G_{m2} + P_{m3}P_{m2}G_{m1} + P_{m3}P_{m2}P_{m1}C_0$$

$$G_{m3} = G_{12} + P_{12}G_{11} + P_{12}P_{11}G_{10} + P_{12}P_{11}P_{10}G_9$$

$$P_{m3} = P_{12}P_{11}P_{10}P_9$$

$$C_{16} = G_{m4} + P_{m4}G_{m3} + P_{m4}P_{m3}G_{m2} + P_{m4}P_{m3}P_{m2}G_{m1} + P_{m4}P_{m3}P_{m2}P_{m1}C_0$$

$$G_{m4} = G_{16} + P_{16}G_{15} + P_{16}P_{15}G_{14} + P_{16}P_{15}P_{14}G_{13}$$

$$P_{m4} = P_{16}P_{15}P_{14}P_{13}$$

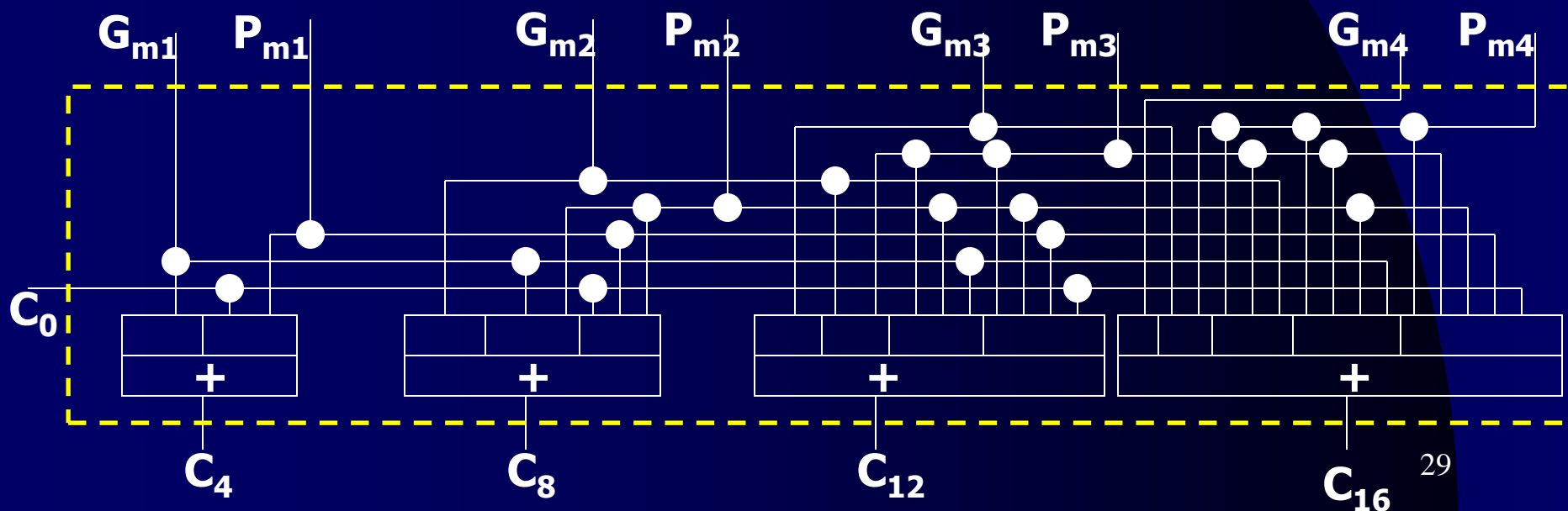
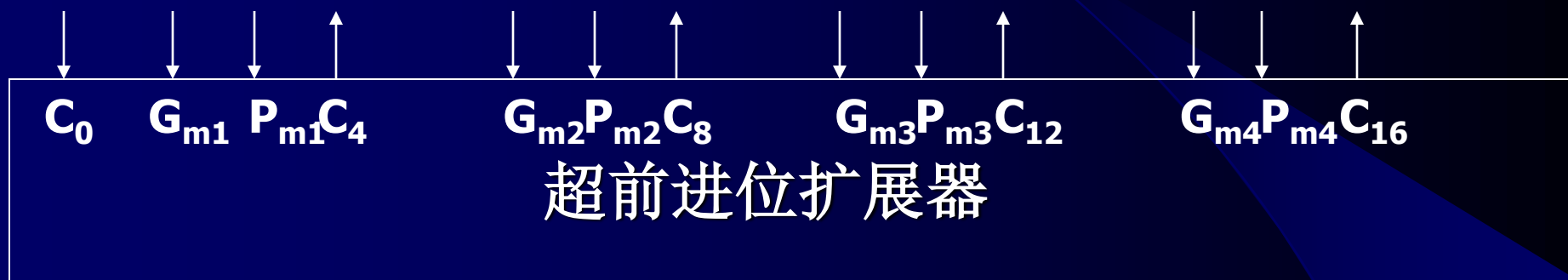
## 3.3.5 运算器

$$C_4 = G_{m1} + P_{m1}C_0$$

$$C_8 = G_{m2} + P_{m2}G_{m1} + P_{m2}P_{m1}C_0$$

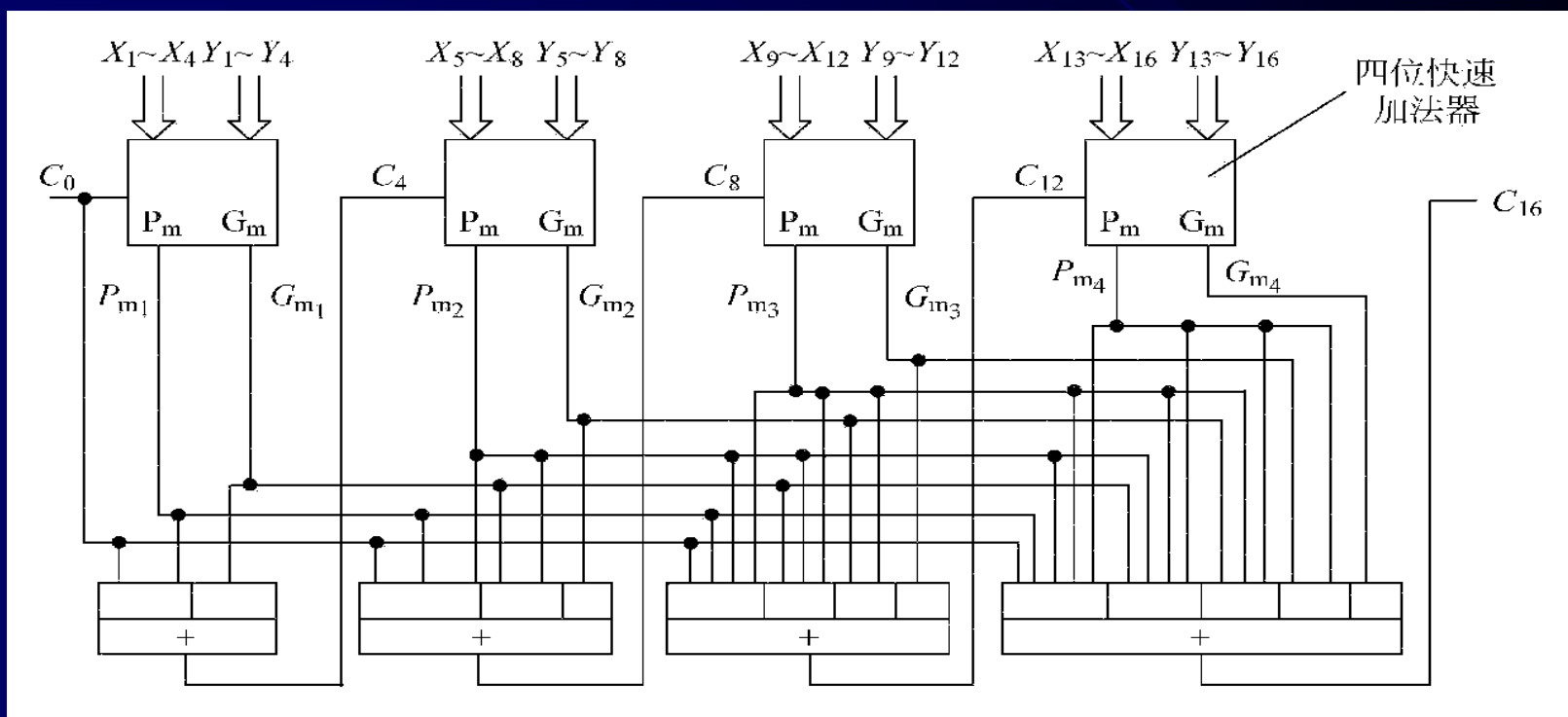
$$C_{12} = G_{m3} + P_{m3}G_{m2} + P_{m3}P_{m2}G_{m1} + P_{m3}P_{m2}P_{m1}C_0$$

$$C_{16} = G_{m4} + P_{m4}G_{m3} + P_{m4}P_{m3}G_{m2} + P_{m4}P_{m3}P_{m2}G_{m1} + P_{m4}P_{m3}P_{m2}P_{m1}C_0$$



## 3.3.5 运算器

- 16位并行进位加法器:
  - 超前进位扩展器



超前进位扩展器, 使得 $C_4, C_8, C_{12}, C_{16}$ 同时产生!

# 运算器

- 16位并行加法器
  - 用类似四位快速加法器中C1、C2、C3、C4形成的原理，去形成片间快速进位C4、C8、C12、C16
- 能否参照4位并行加法器的设计方法设计一个4位并行减法器？
- 能否用PLA、PAL设计一个4位并行加法器，快速进位电路，快速借位电路？

# 组合逻辑小结

- 表达式可以包含全部分分析和设计原理
- 组合逻辑用简单的门电路可以实现，也可以用集成电路实现，原理上是相通的，只是应用不同。
- 不同的组合逻辑，在功能和要求各有不同，设计细节也各不相同
- 现实生活中是集成电路芯片已经存在，需要用现有的集成电路设计一些要求的电路



# 数字逻辑电路总复习

- 四句话
  - 全书一句话
  - 组合逻辑一句话
  - ⇒ ➤ 时序逻辑一句话
  - 可编程逻辑一句话
- 一个忠告

# 时序逻辑一句话

- 时序逻辑：表达式 + 触发器

- 表达式是激励，触发器用于状态转换
- 分析和设计方法有固定的流程
- 现实生活中的设计却是灵活多样，要考虑很多因素

# 触发器的选用

- 不同的应用需求选择不同种类的触发器
  - 电位触发器：接口双方指示是电位时
  - 边沿触发器：统一时钟指挥
- 虽然边沿触发器应用广泛，但边沿触发器不是触发器的最高级

# 触发器中需要掌握的知识

- 触发器结构和原理
  - 基本RS触发器
  - 电位型触发器
    - ✓ D触发器
    - ✓ RS触发器
  - 边沿触发器
    - ✓ D触发器
    - ✓ JK触发器
- 会灵活运用以上触发器

# 同步时序电路的分析和设计

- 掌握基本概念，基本方法

- 时序电路的结构

- 基本概念：功能表，现态 $Q_n$ ，次态 $Q_{n+1}$ ，状态表与状态图，状态方程，激励表

- 同步时序电路的分析方法和设计方法

# 同步时序电路的分析方法

- 基本的分析方法在课件上都有，不再罗嗦
- 分析方法
  - 可以采用固定的方法：缺点是比较死板，不灵活
  - 较好的方法是：
    - (1) 总体浏览
    - (2) 抓住特征
    - (3) 大胆假设，小心求证：不放过一个连接点和连线  
(如有4个异或就有可能是全加或全减计数器，触发器左右都连线有可能是移位寄存器等等.....)

# 同步时序电路的分析和设计

## ● 设计方法

➤ 可以采用固定的方法：缺点是比较死板，不灵活

➤ 较好的方法是

(1) 从概念和要求出发，经过推理来完成初步设计

(2) 加上实用化的知识，完成一个实用化的设计

# 同步时序电路的应用

- 计数器和寄存器

- 计数器

- 各种进制计数器，可逆计数器等
- 计数器的扩展和应用

- 寄存器

- 基本寄存器
- 移位寄存器
- 移位寄存器应用



# 数字逻辑电路总复习

- 四句话

- 全书一句话

- 组合逻辑一句话

- 时序逻辑一句话

- ⇒ ➤ 可编程逻辑一句话

- 一个忠告

# 可编程逻辑一句话

- 实现表达式+触发器的另一种方法
  - 可编程逻辑没有新知识
  - 有的是新方法
- 可编程逻辑电路的原理，具体集成电路的特点

# 可编程逻辑电路

- ROM, PLA, PAL, GAL的要求
  - 掌握基本原理、特点和基本用法
  - 会用可编程逻辑实现组合逻辑电路和同步时序逻辑电路

# 数字逻辑电路课程一句话

- 表达式+触发器包含全部原理。
- 但我们这门课程是实用化技术，只掌握基本原理和方法是不够的
- 需要掌握很多实用化技术

# 数字逻辑电路中的实用化技术

## ● 实际电路设计中的原则：

- 雷锋精神：将方便留给别人，将困难留给自己。
- 我们课程中讲解了很多实用化的逻辑电路，每一个都是很多人，多年辛勤汗水的结晶！每一幅逻辑电路图都是一个实用化的精美的作品。
- 希望大家能够仔细研究和阅读，体会其中包含的知识

# 数字逻辑电路中的实用化技术

## ● 设计中的实用化技术

- 负载问题
- 可靠性问题
- 扩展问题
- 线与问题
- 自启动问题

# 数字逻辑电路中的实用化技术

## ● 设计中的实用化技术

### ➤ 负载问题

- (1) 输入缓冲：减少外部负载
- (2) 内部负载分配
- (3) 输出的扇出最大化（用 $\overline{Q}$ 反馈到内部）

# 数字逻辑电路中的实用化技术

## ● 设计中的实用化技术

### ➤ 可靠性问题

(1) 设计的电路结构规则：如设计电路中有时用“异或”而不用的“与或”

➤ 好处：传输延迟可预测，可读性好，布线方便

(2) 添加冗余电路：消除竞争与冒险



# 数字逻辑电路中的实用化技术

## ● 设计中的实用化技术

- **扩展问题：**设计所有的集成电路都要考虑扩展，没有可扩展性的设计是没有生命力的。

# 数字逻辑电路中的实用化技术

## ● 设计中的实用化技术

### ➤ 线与问题

- (1) 总线的接口电路采用三态结构
- (2) 集电极开路的“线与”

# 数字逻辑电路中的实用化技术

## ● 设计中的实用化技术

- **自启动问题：**所有时序电路都必须解决的问题，否则设计出来的产品是废品。

# 数字逻辑电路总复习

## ● 四句话

- 全书一句话
- 组合逻辑一句话
- 时序逻辑一句话
- 可编程逻辑一句话

⇒ ● 一个忠告

# 数字逻辑电路课程的结束语

- 一句忠告：

- 本课程特点：“说起来容易做起来难”，完整的逻辑设计加上实用化技术才可以设计出合格的产品

- 补充：完整的逻辑设计可以从课本得到，但实用化技术需要长期的积累。

- 祝愿大家好好复习，取得好成绩！
- 我们共同度过的这学期让我感到很愉快！！！！